ICONS 2012

The Seventh International Conference on Systems

VisGra 2012

The First International Workshop on Computer Vision and Computer Graphics

ISBN: 978-1-61208-184-7

February 29 - March 5, 2012

Saint Gilles, Reunion Island

ICONS 2012 Editors

Hermann Kaindl, Vienna University of Technology, Austria

Leszek Koszalka, Wroclaw University of Technology, Poland

Herwig Mannaert, University of Antwerp, Belgium

Marko Jäntti, University of Eastern Finland, Finland

VisGra 2012 Editors

Petre Dini, Concordia University, Canada / China Space Agency Center, China

Vaclav Skala, University of West Bohemia, Plzen and VSB-Technical University, Ostrava, Czech Republic

ICONS 2012

Foreword

The Seventh International Conference on Systems [ICONS 2012], held between February 29th and March 5th, 2012 in Saint Gilles, Reunion Island, continued a series of events covering a broad spectrum of topics. The conference covered fundamentals on designing, implementing, testing, validating and maintaining various kinds of software and hardware systems. Several tracks were proposed to treat the topics from theory to practice, in terms of methodologies, design, implementation, testing, use cases, tools, and lessons learnt.

In the past years, new system concepts have been promoted and partially embedded in new deployments. Anticipative systems, autonomic and autonomous systems, self-adapting systems, or on-demand systems are systems exposing advanced features. These features demand special requirements specification mechanisms, advanced behavioral design patterns, special interaction protocols, and flexible implementation platforms. Additionally, they require new monitoring and management paradigms, as self-protection, self-diagnosing, self-maintenance become core design features.

The design of application-oriented systems is driven by application-specific requirements that have a very large spectrum. Despite the adoption of uniform frameworks and system design methodologies supported by appropriate models and system specification languages, the deployment of application-oriented systems raises critical problems. Specific requirements in terms of scalability, real-time, security, performance, accuracy, distribution, and user interaction drive the design decisions and implementations. This leads to the need for gathering application-specific knowledge and develop particular design and implementation skills that can be reused in developing similar systems.

Validation and verification of safety requirements for complex systems containing hardware, software and human subsystems must be considered from early design phases. There is a need for rigorous analysis on the role of people and process causing hazards within safety-related systems; however, these claims are often made without a rigorous analysis of the human factors involved. Accurate identification and implementation of safety requirements for all elements of a system, including people and procedures become crucial in complex and critical systems, especially in safety-related projects from the civil aviation, defense health, and transport sectors.

Fundamentals on safety-related systems concern both positive (desired properties) and negative (undesired properties) aspects. Safety requirements are expressed at the individual equipment level and at the operational-environment level. However, ambiguity in safety requirements may lead to reliable unsafe systems. Additionally, the distribution of safety requirements between people and machines makes difficult automated proofs of system safety. This is somehow obscured by the difficulty of applying formal techniques (usually used for equipment-related safety requirements) to derivation and satisfaction of human-related safety requirements (usually, human factors techniques are used).

This conference also featured the workshop:

- VisGra 2012, The First International Workshop on Computer Vision and Computer Graphics

We take here the opportunity to warmly thank all the members of the ICONS 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 ICONS 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 ICONS 2012 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICONS 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 systems.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the charm of Saint Gilles, Reunion Island.

ICONS Chairs:

Raimund Ege, Northern Illinois University, USA
Hermann Kaindl, Vienna University of Technology, Austria
Leszek Koszalka, Wroclaw University of Technology, Poland
Daniela Dragomirescu, LAAS-CNRS / University of Toulouse, France

VisGra Chairs:

Petre Dini, Concordia University, Canada / China Space Agency Center, China
Vaclav Skala, University of West Bohemia, Plzen and VSB-Technical University, Ostrava, Czech Republic

ICONS 2012

Committee

ICONS Advisory Committee

Raimund Ege, Northern Illinois University, USA
Hermann Kaindl, Vienna University of Technology, Austria
Leszek Koszalka, Wroclaw University of Technology, Poland
Daniela Dragomirescu, LAAS-CNRS / University of Toulouse, France

ICONS 2012 Technical Program Committee

Marco Aiello, University of Groningen, The Netherlands
Giner Alor Hernández, Instituto Tecnológico de Orizaba - Veracruz, México
César Andrés, Universidad Complutense de Madrid, España
Rafic Bachnak, Texas A&M International University - Laredo, USA
Lubomir Bakule, Institute of Information Theory and Automation of the ASCR, Czech Republic
Jacob Barhen, Oak Ridge National Laboratory, USA
Nicolas Belanger, Eurocopter Group, France
Ateet Bhalla, NRI Institute of Information Science and Technology - Bhopal, India
Jun Bi, Tsinghua University - Beijing, China
Freimut Bodendorf, University of Erlangen-Nuremberg, Germany
Mietek Brdys, University of Birmingham, UK
Mario Cannataro, University "Magna Græcia" of Catanzaro - Germaneto, Italy
Wojciech Cellary, Poznan University of Economics, Poland
Chi-Hua Chen, National Chiao Tung University, Taiwan , R.O.C.
Albert M. K. Cheng, University of Houston, USA
Andrzej Chydzinski, Silesian University of Technology - Gliwice, Poland
Nicolas Damiani, Eurocopter Group, France
Jianguo Ding, University of Luxembourg, Luxembourg
António Dourado, University of Coimbra, Portugal
Daniela Dragomirescu, LAAS-CNRS / University of Toulouse, France
Raimund Ege, Northern Illinois University, USA
Yezyd Enrique Donoso Meisel, Universidad de los Andes - Bogotá, Colombia
Andras Farago, The University of Texas at Dallas, USA
Miguel Franklin de Castro, Federal University of Ceará, Brazil
Laurent George, University of Paris-Est Creteil Val de Marne, France
Eva Gescheidtová, Brno University of Technology, Czech Republic
Luis Gomes, Universidade Nova de Lisboa, Portugal
Dongbing Gu, University of Essex - Colchester, UK
Aseem Gupta, Freescale Semiconductor - Austin, USA
Mohanad Halaweh, UD College of Information Technology, UAE
Yo-Ping Huang, National Taipei University of Technology - Taipei, Taiwan
Wen-Jyi Hwang, National Taiwan Normal University - Taipei, Taiwan
Jiri Jaros, Australian National University, Australia
Jaroslav Kadlec, Brno University of Technology, Czech Republic

VisGra 2012

Committee

IARIA Advisory Chair

Petre Dini, Concordia University, Canada / China Space Agency Center, China

VisGra Workshop Chair

Vaclav Skala, University of West Bohemia, Plzen and VSB-Technical University, Ostrava, Czech Republic

VisGra 2012 Technical Program Committee

Selim Balcisoy, Sabanci University, Turkey
Gladimir Baranoski, University of Waterloo, Canada
Bedrich Benes, Purdue University, USA
Kadi Bouatouch, IRISA, University of Rennes 1, France
Leszek J. Chmielewski, Warsaw University of Life Sciences, Poland
Sabine Coquillart, INRIA, France
Stuart Fergusson, Queen University -Belfast, UK
Francisco R. Feito, University of Jaen, Spain
Marina Gavrilova, University of Calgary, Canada
Ugur Gudukbay, Bilkent University, Turkey
Helwig Hauser, University of Bergen, Norway
Eckhard M.S. Hitzer, University of Fukui, Japan
Rafael Jesus Segura Sanchez, University of Jaen, Spain
James T. Klosowski, AT&T Labs Research, USA
Helio Pedrini, University of Campinas, Brazil
Jose Ignacio Rojas Sola, University of Jaen, Spain
Przemyslaw Rokita, Warsaw University of Technology, Poland
Isaac Rudomin, Tec de Montrerrey, Mexico
Georgios Sakas, Fraunhofer-Institut-IGD, Germany
Heidrun Schumann, University of Rostock, Germany
Wu Shi Ting, University de Campinas, Brazil
Alade O.Tokuta, North Carolina Central University, USA
Vaclav Skala, University of West Bohemia / VSB-Technical University, Czech Republic
Gunter Weiss, TU Dresden, Germany
Charles Wuethrich, University of Weimar, Germany

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

Mobile Code Security in Contemporary Information Systems - Past, Present and Trends

Denis Trček
Faculty of Computer and Information Science
University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana, Slovenia - EU
denis.trcek@fri.uni-lj.si

Marko Bajec
Faculty of Computer and Information Science
University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana, Slovenia - EU
marko.bajec@fri.uni-lj.si

***Abstract*—Despite the fact that mobile code security issues appeared some ten years ago, they remain important also in the era of service oriented architectures and cloud computing. Mobile code security certainly has some specifics, because it presents an opposite paradigm from the traditional one. In the traditional setting a host (i.e., local operating system, local environment) has to be protected against the code, while with the mobile code security this very code (more precisely, the program code and data) has to be protected against malicious host. And significant break-through is still to be seen in this area. Therefore this paper provides an analysis of the main problems related to mobile code security, and gives an outlook by identifying focuses of future research. It also provides a new paradigm called non-deterministic security services to address mobile code security issues. Clearly, even with the latest advancements in communications systems, the importance of secure mobile code remains one of the most challenging issues in information systems, and critical infrastructures.**

Keywords–information systems; critical infrastructures; distributed services; security; mobile code integrity.

I. INTRODUCTION

The latest advancements in information technology (IT) and information systems (IS) in general are leading towards SOA (services oriented architectures) [1], [2] and cloud computing [3] (actually, SOA is a concept that plays an important role in the area of cloud computing, and one concrete implementation of SOA are Web Services [4]).

Mobile code security issues were first tackled some ten years ago within the area of (intelligent) mobile agents. Despite the above new emerging IT and IS trends, these trends are not reducing the importance of mobile code and its security - one should just think of search engines and web crawlers that are needed for their operation.

The core problem of mobile code security is the opposite from the one in traditional security. While in traditional environments we want to protect local operating systems, programs and data from a malicious incoming code, in mobile code security area this paradigm is reversed. In this latter case we have to assure protection of mobile code, its original data and data that the mobile code has gathered while traversing the (global) network against a local, foreign malicious environment.

Many generic threats in mobile code security area have been identified in the past, and their overview will be given in the next section. In the third section, security mechanisms will be given that have been developed to counter these threats. In the fourth section, an analysis of the these counter measures will follow with the identification of open issues. In the fifth section, there will be argumentation that a paradigmatic shift is needed in the area of mobile code security, and the concept of non-deterministic security services will be given. There are conclusions given in the sixth section, which are followed by acknowledgement and references.

Last but not least, security services are about authentication, confidentiality, integrity, non-repudiation, access control, auditing and alarming [5]. The focus of this paper is on integrity of mobile code, which seems to be the hardest issue beside confidentiality.

II. MOBILE CODE SECURITY THREATS

The area of mobile code security threats is now quite reach, but it was comprehensively covered by NIST already at the end of the former century [6]. NIST work can be treated as a kind of a reference work and we summarize these threats as follows:

- Masquerade - A hosting platform may pretend to be the other one than claimed. In case of pretending to be a trusted third party, such platform may get access to confidential data, intervene with agent's communication with other agents, etc. Masquerade can also happen when an agent deploys services of a remote platform that is playing a masquerade.
- Denial of service - When a malicious hosting environment does not respond, it effectively stops the artificial life of an agent and prevents its goals and mission to be fulfilled. The platform may stuck the agent also by continuously assigning new subtasks to the agent. As agents communicate and may perform some distributed tasks, the whole community of agents gets stuck this way. It is also possible that the local platform (or remote

 ISBN: 978-1-61208-184-7

platform) that are needed by agent are attacked by some third party, and thus not able to provide services needed by an agent.
- Eavesdropping - The hosting malicious platform may not only eavesdrop an inter-agent communications, but all agent's data, its code and the execution process of this code. Even in case where parts of agent's data are encrypted, such platform has access to complete local life of the agent and can infer with the content of this encrypted code. Similar applies to agent's communication with other agents.
- Alteration - When being hosted by a malicious host, agent's code, state and data may be easily exposed to this host, and altered. The similar holds true for agent's communications. Let us state already at this point that the security service of (strong) integrity can prevent this threat to some extent, but it will be discussed in the next section how hard this is.
- Unauthorized access - Remote platforms and other agents may get access to agent's code, state and data, for which they are not authorized. In case of a local platform, this problem reduces to the above described threat of eavesdropping.
- Copy and replay - A malicious platform may simply reproduce an agent or its message(s), so an exact (and virtually regular copy) can be obtained. This can have serious security impacts - for example, original agent's messages may be reproduced by its clone. Therefore they can be accepted by other parties as fully valid, and reacted accordingly upon.

These threats have been quite successfully addressed in the latest years and will be described in the next section.

III. Mobile code security mechanisms and services

This section provides security mechanisms and services that were developed during the last years to address the above threats these are typical for mobile code security (beside from [6] below solutions are taken also from [7] and [8]):

- Co-operating agents principle - This principle requires that tasks are split and assigned to agents that run on different platforms (these agents should never encounter the same host), while these agents use authenticated (and confidential) channels to communicate.
- Execution tracing principle - This principle is intended to detect improper modification of agents code, state or execution flow. Cryptographic traces (logs of agents actions performed during its lifetime) serve for this purpose. Each platform produces digitally signed traces and appends them to the agent (digitally signed knapsack).
- Environmental key-generation principle - An agent waits until it receives a certain message (typically containing a decryption key) that is generated by other party when a certain condition is met in the environment. After receipt the agent decrypts its encrypted part of code and executes it.
- Code obfuscation - An agent is transformed in a way that preserves the intended behavior of the agent, while makes analysis of its code harder.
- Partial result encapsulation - With this technique the results of agent's calculations are encrypted by using the private key of a visited platform. Afterwards, the results are sent to domestic platform for verification.
- Sliding encryption - An agent uses a certain public key (e.g. of its domestic environment) to encrypt sensitive data, which can be later decrypted only by an entity that possesses the corresponding private key.
- Trail obscuring - With this technique an agent modifies its own binary image to make pattern matching harder so it cannot be identified as the same agent when traversing from one form to another.
- State appraisal functions - These functions serve to prevent tampering with agents dynamic data and reside in the encrypted (or digitally signed) part of the agent. They are built into agent by the sender (domestic environment) and can be used not only by the agent itself, but also trustfully behaving visited platforms.
- Encrypted functions (also referred to as mobile cryptography) - This approach deploys a principle where agent's code would be encrypted in a way that would enable correct execution at a guest platform, while the platform would not recognize the content (semantics) of this execution.

As to the last bullet - encrypted functions would really enable a significant advancement. The initial work in this area has been done by Sander and Tsudin in 1998 [9], and later extended by Lee, Alves-Foss and Harisson in 2004 [10]. The basic principle goes as follows [6]:

Suppose A knows an algorithm for computing a function f, while B has an input x for this function. A wants to compute $f(x)$ for B without revealing any details of f to B. If f can be encrypted in such way that it results in $E(f)$, then A creates program $program(E(f))$ and sends it to B. B executes $program(E(f))$ on x and returns the output to A who decrypts the result and obtains $f(x)$. This paradigm would provide effective means for solving mobile code challenges, but such cryptographic principle still needs appropriate concrete implementations (functions) to be found. Therefore things in this area (seem to) remain stalled at a theoretical level.

The last research that should bring some new advancement in this area is published in [11]. This solution is focused on confidentiality of code and agent's baggage together with integrity during agent's execution. The analysis of this paper reveals that the solution is about very complex architecture with many security protocols. History shows

 ISBN: 978-1-61208-184-7

that such complex structures are often leaking at some point (the formal verification of this solution is yet to be done). Further, the solution does provide integrity of a runtime code, but at a very high level through encryption of the whole code. Once a hosting platform is supposed to be trusted, it receives decryption key and from this point on, ”peeking & poking” of the code becomes feasible. This is not to say that this approach is useless - on the contrary. Our proposed solution actually further complements such solutions at a finer level and provides additional security, as will be seen in the rest of the paper.

IV. An analysis of existing countermeasures

In short - none of the above approaches radically reduces the problem of mobile code integrity and its protection against malicious host (except, of course, mobile code cryptography).

This is the core problem, which we will refer to as code morphing problem: When an agent is executed within a foreign operating system (or virtual machine) this operating system (or hypervisor in case of a virtual machine) can ”peek and poke” working memory locations (i.e., RAM locations) where agents execution code resides. The only 100% security would be if one could do strong integrity probing of executing agent in a RAM. But this is impossible because of operating systems principles like splitting a code and placing it into various RAM locations, loading it only partially into RAM (paging), etc. Therefore the actual binary image within various hosting environments can have too many varying digital representations when executed to provide a unique digital fingerprint.

Now what can be appropriate concept to counter this situation? The basic fact is that integrity is a security service and these services are implemented by using cryptographic protocols. This already gives the first hint, which (as a positive by product) complements research efforts at the level of security mechanisms (i.e., encrypted functions). Further, why shall one remain focused on a deterministic output at the level of security service? We could go for a solution where runs of a certain protocol with the same input would result in outputs that are dispersed according to some distribution on a certain interval. This distribution can then serve as a basis for calculation of a probability which of the obtained values is actually the correct one. And this is the core idea of non-deterministic security services.

Particular such approaches in this area already exist, starting with zero-knowledge (ZN) techniques on one side [12] and RFID tailored solutions on the other [13]. But they have not been recognized as a new concept so far. Such conceptual approach would be beneficial, and an advancement due to a paradigmatic shift would not happen for the first time in the area of security in computer systems (and cryptography in general). Actually, public key cryptography (PKI) was invented on the basis of a concept and principles

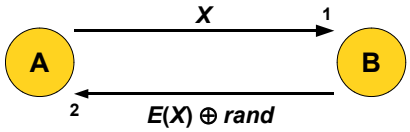


Figure 1. Classical challenge-response modified for non-deterministic security service (E stands for encryption function, X for challenge and $rand$ for a random bits string)

that were clearly developed in advance. PKI started with a strong focus of a freelance cryptographer W. Diffie on the problem of keys distribution. Diffie got in contact with M. Hellman at Stanford in the seventies of the former century. Having a clear concept in their minds they started to look for appropriate concrete candidate functions, and this led to the invention of public key cryptography.

V. Paradigmatic shift - non-deterministic security services

Now what is a non-deterministic security service? This kind of security refers to security services (i.e., cryptographic protocols) that

- do not provide a unique output after one run, so this output is from a certain interval of possible values and thus considered to be correct with a certain probability

OR

- require more (potentially parallel) runs with varying outputs, where all these outputs have to be considered as a whole to calculate the probability of assurance of the intended security service of the protocol.

To demonstrate the principle of the first case, assume a classical challenge-response authentication protocol, where the first message is, as usual, some random challenge. However, the second step is modified by EXOR-ing certain pre-agreed bit positions with a random string (see Fig.1). On receipt, the receiver should check all possible outputs as follows. All modified positions are initially assumed to be 0, and systematically changing them, bit by bit, one obtains outputs of all possible responses. If one of these messages produces the right output, the receiver can be assured about identity of the other party up to a certain probability (which depends on the number of masked bits and concrete encryption mechanism).

The first principle is given in [13], while to demonstrate the principle of the second case, let us consider Zero-knowledge based proof, the Fiat-Shamir protocol [14]:

1) Claimant A selects a secret s and computes $v = s^2 \bmod n$ and registers the result with a trusted center.
2) A sends to verifier B the value $x = r^2 \bmod n$.
3) Verifier B responds with a random $e \in 0, 1$.
4) A replies with $y = rs^e \bmod n$.
5) B verifies $y^2 \equiv xv^e (\mathrm{mod} n)$, and depending on e checks if the obtained value $y^2 = x$ or $y^2 = xv$.

 ISBN: 978-1-61208-184-7

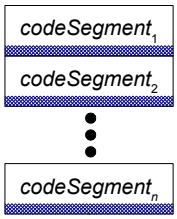


Figure 2. Agent's code structure (dashed areas denote segments' ICC, i.e., a code for its calculation and communication with the reference host)

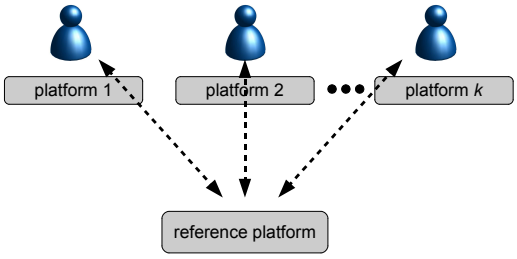


Figure 3. Protocol architecture for non-deterministic integrity checks

In the above scheme, a trusted center selects and publishes RSA-like modulus $n = pq$, where p and q are large primes that are kept secret, $s \in [1, n-1]$, and $y = 0$ is rejected, because it precludes $r = 0$. Without going into details, an attacker can impersonate A by choosing appropriate r, but once this value is selected, e in the second step defines the required result in the third step. As e is randomly chosen with probability 0.5, the attacker can successfully cheat in a single round with this probability. To reduce attacker's success probability, protocol can be run k times (steps 2 to 5), and the probability of successful cheating in this case becomes 2^{-k}.

Now how can non-deterministic security services be deployed for mobile code integrity? One possible protocol architecture is given in Fig. 2 and Fig. 3. The core premise of this approach is that each agent is segmented, and each of the segments has appended integrity check value (ICV). The main reasoning behind this is to actively check agents also during execution, when the code may be paged or dislocated in various segments of RAM.

Next, the agent is exactly replicated and forwarded to n subsequent hosts. Therefore these replicas are deployed over the network by communicating between a trusted, reference host, and n remote (distrusted) hosts. Remote hosts are continuously polled and the results are obtained and compared. As it is very unlikely that all hosts will be able to synchronize a coordinated attack on each instance and all related segments of its code, the reference platform will, in case of malicious host, obtain ICV values that will be dispersed. According to a concrete segmentation of an agent, number of its deployed instances, the nature of the polling protocol and the nature of the ICV, the reference platform is able to infer the probability of a proper behavior of a particular instance of the agent. Based on this data, reference platform can identify healthy agents and instruct them to proceed to other platforms, while attacked agents can be destroyed.

One complementary research already exists, but it is concerned with reliability of agents execution and deploys non-determinism in the agent code to improve this reliabilty [15]. So this work can be used for management of deploying multiple instances as mentioned above. However, t should be emphasized that the approach is not about security - it is about reliability and fault tolerance.

VI. Outlook and Conclusions

Mobile code and its security is playing a vital role also today in the era when internet has become a critical infrastructure and when it is architecturally heading towards cloud computing. However, as we have seen, one of the hardest issues is to ensure mobile code integrity. The paper has presented and analyzed approaches that were introduced during recent years, however, none of them radically reduces the problem.

Therefore this paper argues that we should change our paradigm and try to research mobile code integrity issues by deploying a new concept, called non-deterministic security services. This argumentation is based on the fact that paradigmatic shift was crucial also in the case of invention of public key cryptography (interestingly, paradigmatic shift is also visible in mobile cryptography area, too). With the proposed shift of paradigm the code is segmented and each segment checked for ICV during execution, while many replicas are deployed on various foreign hosts.

However, this solution requires development of new concrete implementations, where so-called lightweight protocols (and mechanisms) will play a central role.

Acknowledgments

Authors thank to Slovene Research Agency ARRS for support of this research with grant P2-0359 (Pervasive computing). We also thank to the reviewers that have provided constructive comments, which has helped us to improve the paper.

References

[1] Nagappan R., Skoczylas R., Sriganesh P.R., *Developing Java Web Services*, John Wiley & Sons, Indianapolis, 2003.

[2] Chase N., *XML Primer Plus*, SAMS Publishing, Indianapolis 2002.

[3] Mell P., Grance T., The NIST Definition of Cloud Computing, NIST Special Pub. 800 - 145 (Draft), Gaithersburg, 2011.

[4] Booth D., Haas H., McCabe F., Newcomer E., Champion M., Ferris C., Orchard D. (Eds.), Web Services Architecture, W3C Working Group Note, February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

 ISBN: 978-1-61208-184-7

[5] ISO, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*, ISO standard 7498-2, Geneva, 1989.

[6] Jansen W., Karygiannis T., *Mobile Agent Security*, NIST Special Publication 800-19, Gaithersburg, 1999.

[7] Alfalayleh M., Brajkovic L., An Overview of Security Issues and Techniques in Mobile Agents, *Proc. of the Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS2004)*, pp. 59-78, Lake Windermere, 2004.

[8] Greenberg M.S., Byington J.C., Holding T., Harper D.G., Mobile Agents and Security, IEEE Communications Magazine, Vol. 36, No.7, pp. 76-85, IEEE, 1998.

[9] Sander T., Tschudin C., Protecting Mobile Agents Against Malicious Hosts. In G. Vigna, editor, Mobile agents and security, Lecture Notes in Computer Science, Vol. 1419, pp. 44–60. Springer-Verlag, 1998.

[10] Lee H., Alves-Foss J., Harrison S., The Use of Encrypted Functions for Mobile Agent Security, *Proc. of the 37.th Hawaii Int. Conference on System Sciences*, pp. 110, Hawaii, 2004.

[11] Shibli M.A., Muftic S., Giambruno A., Lioy A., MagicNET: Security System for Development, Validation and Adoption of Mobile Agents, *Proc. of the NSS 2009*, pp. 389-396, 2010.

[12] Goldreich O., Micali S., Wigderson A., *Proofs that yield nothing but their validity*, Journal of the ACM, Vol. 38, No. 3, pp. 690-728, 1991.

[13] Trcek D., Japinnen P., RFID security, RFID and sensor networks: architectures, protocols, security, and integrations, pp. 147–168, Taylor & Francis, Boca Raton, 2010.

[14] Menezes A.J., van Oorschot P., Vanstone S. A., Handbook of Applied Cryptography, CRC Press, Boca Raton, 1996.

[15] Mohindra A.,and Purakayastha A., Exploiting non-determinism for reliability of mobile agent systems, *Proc. of the International Conference on Dependable Systems and Networks*, pp. 144–153, New York, 2000.

 ISBN: 978-1-61208-184-7

A Trusted Information Agent for Security Information and Event Management

Luigi Coppolino
Epsilon S.r.l.,
Naples, Italy
luigi.coppolino@epsilonline.com

Michael Jäger
Technische Hochschule Mittelhessen
Giessen, Germany
michael.jaeger@mni.thm.de

Nicolai Kuntze and Roland Rieke
Fraunhofer Institute for
Secure Information Technology
Darmstadt, Germany
{nicolai.kuntze,roland.rieke}@sit.fraunhofer.de

***Abstract*—This paper addresses security information management in untrusted environments. A security information and event management system collects and examines security related events and provides a unifying view of the monitored system's security status. The sensors, which provide the event data, are typically placed in a non-protected environment at the boarder of the managed system. They are exposed to various kinds of attacks. Compromised sensors may lead to misjudgement on the system's state with possibly serious consequences. The particular security requirements arising from these problems are discussed for large scale critical infrastructures. The main contribution of this paper is a concept that provides trusted event reporting. Critical event sources are holistically protected such that authenticity of the security related events is guaranteed. This enables better assessment of the managed system's reliability and trustworthiness. As a proof of this concept, the paper presents an exemplary realisation of a trustworthy event source.**

Keywords-reliability aspects of security information and event management systems; trusted event reporting; trusted android application; critical infrastructure protection.

I. INTRODUCTION

Security information and event management (SIEM) systems provide important security services. They collect and analyse data from different sources, such as sensors, firewalls, routers or servers, and provide decision support based on anticipatory impact analysis. This enables adequate response to attacks as well as impact mitigation by adaptive configuration of countermeasures. The project MASSIF [1], a large-scale integrating project co-funded by the European Commission, addresses these challenges with respect to four industrial domains: (i) the management of the Olympic Games information technology (IT) infrastructure [2]; (ii) a mobile phone based money transfer service, facing high-level threats such as money laundering; (iii) managed IT outsource services for large distributed enterprises; and (iv) an IT system supporting a critical infrastructure (dam) [3].

Common to these use cases is the requirement to prove that a measured value has been acquired at a certain time and within a specified "valid" operation environment. Authenticity of such measures can only be assured together with authentication of the used device itself, it's configuration, and the software running at the time of the event.

In geographically dispersed infrastructures, various equipment, including the critical sources of event data, is often placed in non-protected environments. Therefore, attackers are able to access and manipulate this equipment with relative ease[4].

Proposition 1. *When physical access to the sensoring devices cannot be inhibited, an effective security solution must address detection of manipulations.*

Manipulated equipment can be used to hide critical conditions, generate false alerts, and in general cause misjudgement on system's state. Wrong assumptions about a system's state in turn can lead to false decisions with severe impact on the overall system.

Proposition 2. *Whenever a certain control decision is made, the input information that presumably led to it must be authentic.*

As a consequence, the system has to assure that all safety critical actions using sensor data must only use authentic sensor data. The question, which measurements and system control decisions are critical to the overall system behaviour, cannot be answerded independently of the concrete system and application context determined.

Proposition 3. *A risk assessment of the deployed monitoring capabilities is necessary.*

Contribution: By means of a representative example, namely a hydroelectric power plant in a dam, we analyse security threats for critical infrastructures and justify the relevance of the postulated propositions for adequate security requirements. Further, the paper presents both, a concept and a prototypical implementation for trustworthy event reporting. Digital signatures obviously can provide authenticity and integrity of recorded data [5]. However, a signature gives no information on the status of the measurement device at the time of measurement. Our solution, the *trusted information agent* (TIA), is based on trusted computing technology [6] and integrates industry approaches to the attestation of event reporter states. This approach provides a certain degree of trustworthiness and non-repudiation for the collected events, which can be used as a basis for risk assessment according to Proposition 3.

The paper is structured as follows. Section II gives an overview of the related work. In Section III we introduce the

 ISBN: 978-1-61208-184-7

exemplary application scenario. We then elicit a number of specific security requirements from the application scenario and justify the propositions for our concept in Section IV. Based on these requirements, we address a solution for our propositions and describe the concept and a prototypical implementation of a trusted information agent in Section V. Finally, the paper ends with conclusions and an outlook in Section VI.

II. Related Work

The paper addresses the integration of Trusted Computing concepts into SIEM systems for critical infrastructures based on examples from a hydroelectric power plant in a dam.

Security information and event management technology provides log management and compliance reporting as well as real-time monitoring and incident management for security events from networks, systems, and applications. Current SIEM systems' functionalities are discussed in [7]. SIEM systems manage security events but are not concerned with the trustworthiness of the event sources. Security requirements analysis and an authenticity concept for event sources is, however, the main topic of this paper. The specification of the application level security requirements is based on the formal framework developed by Fraunhofer SIT [8]. In this framework, systems are specified in terms of sequences of actions and security properties are constraints on these sequences. Applying the methods of this framework, we derive security requirements for the event sources in the dam scenario.

Dam monitoring applications with *automated data acquisition systems* (ADAS) are discussed in [9], [10]. Usually, an ADAS is organised as a *supervisory control and data acquisition* (SCADA) system with a hierarchical organisation. Details on SCADA systems organisation can be found in [11], [12]. In the majority of cases, SCADA systems have very little protection against the escalating cyber threats.

Compared to traditional IT systems, securing SCADA systems poses unique challenges. In order to understand those challenges and the potential danger, [4] provides a taxonomy of possible cyber attacks including cyber-induced cyber-physical attacks on SCADA systems.

Trusted Computing technology standards provide methods for reliably verifying a system's integrity and identifying anomalous and/or unwanted characteristics [6]. An approach for the generation of secure evidence records was presented in [13]. This approach, which is the basis for our proof-of-concept implementation, makes use of established hardware-based security mechanisms for special data recording devices. Our communication protocols extend the Trusted Network Connect (TNC) [14] protocol suite. We use the open source implementation of IF-MAP presented in [15].

III. Application Scenario

Our analysis of security threats for critical infrastructures is based on examples from a hydroelectric power plant in a dam. The dam scenario is typical for critical infrastructures in many respects. On the one hand, it is a layered system with intra- and cross-layer dependencies, and, on the other hand, there are various other sources of complexity; several distinct functionalities influence controlling and monitoring activities. Moreover, different components, mechanisms, and operative devices are involved, each one with different requirements in terms of produced data and computational loads.

A dam might be devised for a multitude of purposes and its features are strictly related to the aims it is built for, e.g., food water supplying, hydroelectric power generation, irrigation, water sports, wildlife habitat granting, flow diversion, or navigation. Since a dam is a complex infrastructure, a huge number of parameters must be monitored in order to guarantee safety and security. Which parameters are actually monitored, depends on the dam's structure and design (earthfill, embankment or rockfill, gravity, concrete arch, buttress), the purpose (storage, diversion, detention, overflow), and the function (hydroelectric power generation, water supply, irrigation).

Table I
Dam instrumentation sensors

Sensor	Parameter or physical event
Water level sensor (WLS)	Current water level (wl)
Inclinometer/Tiltmeter (TM)	Earth or wall inclination or tilt (tm)
Crackmeter (CM)	Wall/rock crack enlargement (cm)
Jointmeter (JM)	Joint shrinkage (jm)
Piezometer	Seepage or water pressure
Pressure cell	Concrete or embankment pressure
Turbidimeter	Fluid turbidity
Thermometer	Temperature

Table I lists some of the most commonly employed sensors together with a brief explanation of their usage. The heterogeneity of currently used devices is a relevant challenge in the dam process control: they range from old industrial control systems, designed and deployed over the last 20 years and requiring extensive manual intervention by human operators, to more recently developed systems, conceived for automatic operations (SCADA). Indeed, the trend of development is toward increasingly automated dam control systems. While automation leads to more efficient systems and also prevents operating errors; on the downside, it poses a limit to human control in situations, where an operator would possibly foresee and manually prevent incidents.

Modern automated systems support remote management and also provide for centralised control of multiple infrastructures. As an example, the Terni hydroelectric complex, located about 150 Km in the north of Rome, is composed by 16 hydroelectric power plants, three reservoirs (Salto, Turano and Corbara), and one pumping plant, all of them supervised by a single remote command post located at Villa Valle.

As a severe disadvantage, increased automation and remote

 ISBN: 978-1-61208-184-7

Table II
SECURITY RELATED SCENARIOS AND THE RESPECTIVE MONITORING

Monitored Event	Impact	Detection
Changes in the flow levels of the seepage channels	Seepages always affect dams (whatever their structure and design are). Seepage channels are monitored to evaluate the seepage intensity. A sudden change in flow levels could show that the structure is subject to internal erosion or to piping phenomena. This event can be the cause of dam cracks and failures	By inserting into the channel a weir with a known section the depth of water (monitored by using a water level sensor) behind the weir can be converted to a rate of flow.
Gates opening	Intake gates are opened to release water on a regular basis for water supply, hydroelectricity generation, etc. Moreover spillways gates(aka overflow channels) release water (during flood period) so that the water does not overtop and damage or even destroy the dam. Gates opening must be operated under controlled conditions since it may result in: i) Flooding of the underlying areas; ii) Increased rate of flow in the downstream that can ultimately result in a catastrophic flooding of down-river areas.	A tiltmeter (angle position sensor) can be applied to the gate to measure its position angle.
Changes in the turbine/infrastructure vibration levels	Increased vibrations of the infrastructure or the turbines in a hydro-powerplant can anticipate a failure of the structure. Possible reasons for such event include: i) earthquakes (Fukushima, Japan, a dam failure resulted in a village washed away); ii) unwanted solicitations to the turbines (Sayano-Shushenskaya, Siberia, 75 dead due to a failure of the turbines in a hydro-powerplant).	Vibration sensors can be installed over structures or turbines to measure the stress level they are receiving.
Water levels overtake the alert thresholds	Spillway are used to release water when the reservoir water level reaches alert thresholds. If this does not happen the water overtops the dam resulting in possible damage to the crest of the dam (Taum Sauk hydroelectric power station).	This event can be used to detect unexpected discharges. Water level can also be correlated to other parameters to detect anomalous behaviour (e.g., not revealed gate opening).

control raise a new class of security-induced safety issues, i.e., the possibility that cyber attacks against the IT layer of the dam ultimately result in damage to people and environment.

Dam monitoring aims towards identifying anomalous behaviour related to the infrastructure. Table II summarises a list of possible scenarios illustrating the necessity of monitoring specific parameters.

IV. SECURITY REQUIREMENTS ANALYSIS

We use a model-based approach to systematically identify security requirements for the dam application scenario. Specifically, *authenticity* can be seen as the assurance that a particular action has occurred in the past. For a formal specification of the application-level authenticity requirements, we use Definition 1, which is taken from [8].

Definition 1. *$auth(a,b,P)$: Whenever an action b happens, it must be authentic for an Agent P that in any course of events that seem possible to him, a certain action a has happened.*

In [8] a *security modelling framework* (SeMF) for the formal specification of security properties was presented. Requirements are defined by specific constraints regarding sequences of actions than can or can not occur in a system's behaviour. Actions in SeMF represent an abstract view on actions of the real system, which models the *interdependencies* between actions and ignores their functionality. An action is specified in a parameterised format, consisting of the action's name, the acting agent and a variable set of parameters:

$$actionName(actingAgent, parameter1, parameter2, ...)$$

Table III lists the dam scenario actions used for our security requirements analysis.

Table III
DAM ACTIONS

Action	Description
$sense(WLS, wl)$	Measurement of the water level.
$sense(TM, tm)$	Measurement of the tilt.
$sense(CM, cm)$	Measurement of the crack enlargement.
$sense(JM, jm)$	Measurement of the joint shrinkage.
$sense(PP, power)$	Measurement of voltage and current in the power grid. The power plant PP sends commands ppc to the dam control station depending on these measurements.
$sense(SDC, wdc)$	Measurement of the water discharge on the penstock gates PG.
$sense(PG, open)$	Reporting of the state of the penstock gates.
$display(DCS, X)$	Display X at the dam control station, with $X \in \{wl, tm, cm, jm, ppc, wdc, open\}$.
$activate(Admin, cmd)$	Decision of the administrator, which command shall be triggered.
$exec(PG, cmd)$	Command to be executed by penstock gates.

We now analyse some possible misuse cases, which have been reported in the scenario deliverable [16] of the MASSIF project.

Water level sensor compromise: The attacker takes control of the water level sensors and uses them to send spoofed measurements to the dam control station (*DCS*). This hides the real status of the reservoir to the dam administrator (*Admin*). In this way, the dam can be overflown without alarms being raised by the monitoring system.

From this, we get the requirement that the water level measures have to be authentic for the administrator when they are displayed at the dam control station. More formally,

 ISBN: 978-1-61208-184-7

we get the authenticity requirement:

$$auth(sense(WLS, wl), display(DCS, wl), Admin) \quad (1)$$

Tiltmeter compromise: The attacker takes control of the tiltmeter sensors and uses them to send false measurements to the dam control station, thus hiding the real status of the tilt of the dam's walls to the dam administrator. An excessive tilt may lead to the wall's failure. The respective authenticity requirement is:

$$auth(sense(TM, tm), display(DCS, tm), Admin) \quad (2)$$

Crackmeter / jointmeter compromise: The attacker has access to one of the crackmeters or jointmeters deployed across the dam's walls and takes control of it. So the attacker can weaken the joint or increase the size of the crack at the wall's weak point without any alarm being raised at the monitoring station, which leads to the following authenticity requirements:

$$auth(sense(CM, cm), display(DCS, cm), Admin) \quad (3)$$
$$auth(sense(JM, jm), display(DCS, jm), Admin) \quad (4)$$

These examples show that some elementary security requirements can be derived directly from misuse cases. In general, however, information flows between systems and components are highly complex, especially when organisational processes need to be considered. Hence, not all security problems are discoverable easily. In order to achieve the desired security goals, security requirements need to be derived systematically.

An important aspect of a systematic security evaluation is the analysis of potential information flows. A method to elicit authenticity requirements by analysis of functional dependencies is described in [17]. From the use case descriptions, atomic actions are derived and set into relation by defining the functional flow among them. The action-oriented approach considers possible sequences of actions (control flow) and information flow (input/output) between interdependent actions. Actions of interest are specifically the *boundary actions*, which represent the interaction of the system's internals with the outside world. From a functional dependency graph, the boundary actions can be identified. We now give an example of security information flows by a use case of the dam scenario [16].

On demand electric production: The Dam Control Station feeds an hydroelectric turbine, connected to the dam by means of penstocks, for producing electric power on demand. The turbine and hydroelectric power production depends on the water discharge in the penstocks. By analysing the parameters of the command received by the dam control station, we can infer that the safety critical actions are the opening and closing actions of the penstock gates (*PG*).

An identification of functional dependencies reveals that the dam control activity makes use of the (i) current water level, (ii) the state of the gates joined to the hydroelectric power plant, (iii) the gates openness, and, (iv) the discharge through the penstocks. Figure 1 shows the dependency graph of this use case. The decision of the administrator, which command shall be triggered, depends on the displayed measurements. The dashed line indicates that there is no direct functional dependency.

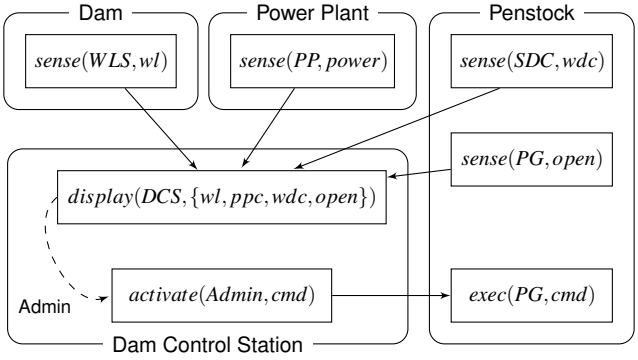


Figure 1. Functional dependencies: *On demand electric production*

An analysis of the dependencies depicted in Figure 1 leads to the following conclusion: The control display values are derived from the measurements of *wl*, *power*, *wdc*, and *open*. From this, we conclude that, in addition to the water level *wl* (1), the measurements of *power*, *wdc*, and *open* have to be authentic. More formally:

$$auth(sense(PP, power), display(DCS, ppc), Admin) \quad (5)$$
$$auth(sense(SDC, wdc), display(DCS, wdc), Admin) \quad (6)$$
$$auth(sense(PG, open), display(DCS, open), Admin) \quad (7)$$

Furthermore, the activation of the penstock command by the administrator has to be authentic for the penstock gate when executing it.

$$auth(activate(Admin, cmd), exec(PG, cmd), PG) \quad (8)$$

So the authenticity requirements for the use case described in Figure 1 are given by: (1) and (5)–(8).

In summary, the analysis of the use case and misuse cases of this critical infrastructure scenario shows that the overall function of the system requires authenticity of measurement values for several sensors, namely (1) – (7). In that sense, the dam scenario is a prime example for the relevance of the requirements postulated in Proposition 1 and 2. It is evident that further types of security requirements are needed in order to cover important liveness properties such as *availability* of necessary information at a certain place and time. In some cases also *confidentiality* of certain information may be required. These requirements are important but not in the scope of the work presented here.

 ISBN: 978-1-61208-184-7

V. Trusted Information Agent

The usefulness of monitoring large systems clearly depends on the observer's level of confidence in the correctness of the available monitoring data. In order to achieve that confidence, network security measures and provisions against technical faults are not enough. As stated above, unrevealed manipulation of monitoring equipment can lead to serious consequences. In order to improve the coverage of this type of requirements in a SIEM framework, we now describe a concept and a prototypical implementation of a trusted information agent (TIA).

A. Trust Anchor and Architecture

As shown in Section IV, protection of the identity of the device for measurement collection is necessary. Furthermore, the lack of control on the physical access to the sensor node induces strong requirements on the protection level.

By a suitable combination of hardware- and software-level protection techniques any manipulations of a sensor have to be revealed. In addition to the node-level protection, network security measures are needed in order to achieve specification-conformant behaviour of the sensor network, e.g., secure communication channels that protect data against tampering. This paper is not intended to discuss network security, neither protection of hardware components. We rather concentrate on the important problem of clandestine manipulations of the sensor software.

A commonly used technique to reveal manipulation of a software component is software measurement: Each component is considered as a byte sequence and thus can be measured by computing a hash value, which is subsequently compared to the component's reference value. The component is authentic, if and only if both values are identical. Obviously, such measurements make no sense if the measuring component or the reference values are manipulated themselves. A common solution is to establish a chain of trust: In a layered architecture, each layer is responsible for computing the checksums of the components in the next upper layer. At the very bottom of this chain a dedicated security hardware chip takes the role of the trust anchor or "root of trust".

Trusted Computing [6] offers such a hardware root of trust providing certain security functionalities, which can be used to reveal malicious manipulations of the sensors in the field. Trusted Computing technology standards provide methods for reliably checking a system's integrity and identifying anomalous and/or unwanted characteristics. A trusted system in this sense is build on top of a Trusted Platform Module (TPM) as specified by the Trusted Computing Group (TCG). A TPM is hardened against physical attacks and equipped with several cryptographic capabilities like strong encryption and digital signatures. TPMs have been proven to be much less susceptible to attacks than corresponding software-only solutions.

The key concept of Trusted Computing is the extension of trust from the TPM to further system components. This concept is commonly used to ensure that a system is and remains in a predictable and trustworthy state and thus produces authentic results. As described above, each layer of the chain checks the integrity of the next upper layer's programs, libraries, etc. On a PC, for example, the TPM has to check the BIOS before giving the control of the boot-process to it. The BIOS then has to verify the operating system kernel, which in turn is responsible for the measurement of the next level. Actually, a reliable and practically useful implementation for PCs and systems of similar or higher complexity is not yet feasible. Sensoring and measuring devices, however, typically have a considerably more primitive architecture than PCs and are well-suited for this kind of integrity check concept. Even for modern sensor-equipped smartphones, able to act as event detectors, but having the same magnitude of computing power that PCs had a few years ago, an implementation of the presented concept is possible. A prototypical implementation is presented in more detail now.

B. Proof of Concept: Base Measure Aquisition

Figure 2 depicts the architecture of the TIA.

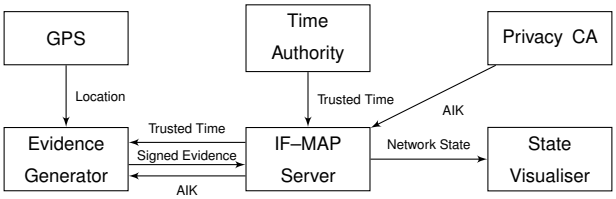


Figure 2. TIA architecture

The main component of the TIA is the *evidence generator* (EG), which collects base measures and provides the measurement functions used to produce derived measures. Furthermore, the EG supports the processing of measures from external sensors, e.g., location data from a GPS module. The EG is expected to operate in unprotected environments with low physical protection and externally accessible interfaces such as wireless networks and USB access for maintenance. A necessary precondition to guarantee authenticity of the measures, is a trustworthy state of the measurement device. To meet this requirement, the EG is equipped with a TPM as trust anchor and implements a chain of trust [18]. As explained above, revelation of software manipulations is based on the comparison between the software checksums and the corresponding reference values. This comparison may be done locally within the node (self-attestation) or by a remote verifier component (remote attestation) [6].

The EG submits the collected measures digitally signed to an IF-MAP [14] server, which acts as an event information broker. During initialisation, the EG obtains two credentials

 ISBN: 978-1-61208-184-7

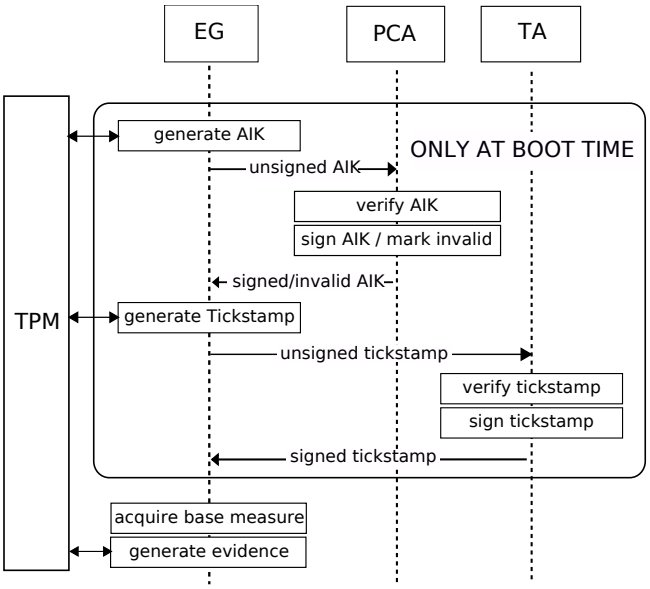


Figure 3. Process model

from trusted third-party services for signature purposes. Figure 3 depicts the boot-time interaction between the EG and those services, and the role of the TPM in this interaction.

An Attestation Identity Key (AIK) is used to sign measurement results in a manner that allows verification by a remote party. The Privacy Certification Authority (PCA) issues a credential for the TPM-generated AIK. The certified AIK is, henceforth, used as an identity for this platform. According to TCG standards, AIKs cannot only be used to attest origin and authenticity of a trust measurement, but also, to authenticate other keys and data generated by the TPM. However, the AIK functionality of a TPM is designed primarily to support remote attestation by signing the checksums of the EG's software components, while signing arbitrary data is, in fact, not directly available as a TPM operation. We have shown elsewhere, how to circumvent this limitation [19]. Hence, we are able to use TPM-signatures for arbitrary data from the EG's sensors.

Any TPM is equipped with an accurate timer. Each event signature includes the current timer value. However, the TPM timer is a relative counter, not associated to an absolute time. A *time authority* (TA) issues a certificate about the correspondence between a TPM timestamp (tickstamp) and the absolute time. The combination of tickstamp and TA-certificate can be used as a trusted timestamp. Alternatively, another trusted time source, such as GPS, could have been used.

Putting it all together, a measurement record includes arbitrary sensor data, a TA-certified time stamp, and a hash value of the EG's software components. The record itself is signed by the TA-certified AIK.

Figure 4 shows a prototype EG, which has been implemented based on the Android smartphone platform. This platform has been selected for various reasons. Modern smartphones are equipped with a variety of sensors such as GPS, gyro sensor, electronic compass, proximity sensor, accelerometer, barometer, and ambient light sensor. Furthermore, photos, video and sound can be regarded and processed as event data. Moreover, Android is well-suited as a software platform for future embedded devices.

The TPM-anchored chain of trust is extended to the linux system and linux application layers by using the Integrity Measurement Architecture (IMA), which is integrated into any stock linux kernel as a kernel module. The Android application layer is based on libraries and the Dalvik Virtual Machine (VM). While the linux kernel layer can check the Android system libraries and the VM, Android applications run on top of the VM and are invisible to the kernel. Thus, we built a modified VM, which extends the chain of trust to the Android application level by computing the applications' checksums. A timestamp-based variant of remote attestation provided by the TPM is used for the verification of the node authenticity. All communication is based on the Trusted Network Connect (TNC) [14] protocol suite, which offers advanced security features, such as dedicated access control mechanisms for TPM-equipped nodes.

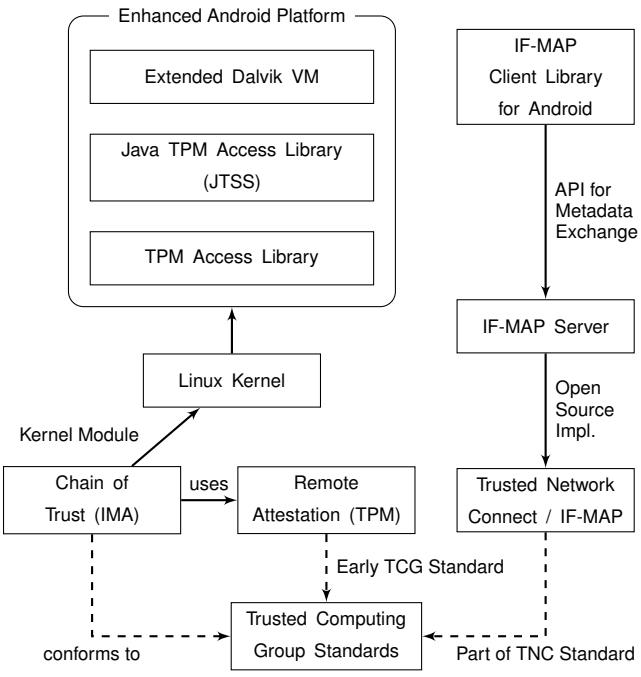


Figure 4. Technical building blocks

VI. Conclusion and Future Work

In geographically dispersed infrastructures the critical sources of event data are often placed in non-protected environments. Attackers can thus easily manipulate these

sensors and thereby hide critical conditions, generate false alerts, and in general cause misjudgement on system's state. By exemplary analysis of a typical application scenario we have demonstrated that this can lead to false decisions with severe impact on the overall system. In order to prevent such threats, we presented a concept for holistically protected critical event sources by assuring a trustworthy state of the measurement devices. This enables better assessment of the managed system's reliability and trustworthiness.

As a proof of this concept, the paper presented an exemplary realisation of a trusted information agent based on trusted computing technology. Planned next steps include a detailed analysis on the impact on scalability and bandwidth of different schemes to generate evidence using this architecture. Especially, the correlation of independent events may allow for improvements but also requires trustworthy schemes to cryptographically link various events to one evidence record. Also, the hardware-based security functionalities can be improved with respect to scalability and performance. Further, suggestions to improve standards for future hardware security modules, are planned.

ACKNOWLEDGEMENT

Luigi Coppolino and Roland Rieke developed the work presented here in the context of the project MASSIF (ID 257475) being co-funded by the European Commission within FP7. Nicolai Kuntze developed the work presented here in the context of the project ESUKOM (ID 01BY1052) which is funded by the German Federal Ministry of Education and Research.

REFERENCES

[1] "Project MASSIF website," 2012. [Online]. Available: http://www.massif-project.eu/

[2] E. Prieto, R. Diaz, L. Romano, R. Rieke, and M. Achemlal, "MASSIF: A promising solution to enhance olympic games IT security," in *International Conference on Global Security, Safety and Sustainability (ICGS3 2011)*, 2011.

[3] L. Coppolino, S. D'Antonio, V. Formicola, and L. Romano, "Integration of a System for Critical Infrastructure Protection with the OSSIM SIEM Platform: A dam case study," in *SAFECOMP*, ser. Lecture Notes in Computer Science, F. Flammini, S. Bologna, and V. Vittorini, Eds., vol. 6894. Springer, 2011, pp. 199–212.

[4] B. Zhu, A. Joseph, and S. Sastry, "Taxonomy of Cyber Attacks on SCADA Systems," in *Proceedings of CPSCom 2011: The 4th IEEE International Conference on Cyber, Physical and Social Computing, Dalian, China*, 2011.

[5] J. Choi, I. Shin, J. Seo, and C. Lee, "An efficient message authentication for non-repudiation of the smart metering service," *Computers, Networks, Systems and Industrial Engineering, ACIS/JNU International Conference on*, vol. 0, pp. 331–333, 2011.

[6] C. Mitchell, *Trusted Computing*. Iet, 2005.

[7] M. Nicolett and K. M. Kavanagh, "Magic Quadrant for Security Information and Event Management," Gartner Reasearch, May 2010.

[8] S. Gürgens, P. Ochsenschläger, and C. Rudolph, "Authenticity and provability - a formal framework," in *Infrastructure Security Conference InfraSec 2002*, ser. LNCS, vol. 2437. Springer, 2002, pp. 227–245.

[9] M. Parekh, K. Stone, and J. Delborne, "Coordinating intelligent and continuous performance monitoring with dam and levee safety management policy," in *Association of State Dam Safety Officials,Proceedings of Dam Safety Conference 2010*, 2010.

[10] B. K. Myers, G. C. Dutson, and T. Sherman, "Utilizing Automated Monitoring for the Franzen Reservoir Dam Safety Program," in *25th USSD Annual Meeting and Conference Proceedings (2005)*.

[11] L. Coppolino, S. D'Antonio, and L. Romano, "Dependability and resilience of computer networks (scada cybersecurity)," in *CRITICAL INFRASTRUCTURE SECURITY: Assessment, Prevention, Detection, Response*. WIT press, in press.

[12] L. Coppolino, S. D'Antonio, L. Romano, and G. Spagnuolo, "An intrusion detection system for critical information infrastructures using wireless sensor network technologies," in *Critical Infrastructure (CRIS), 2010 5th International Conference on*, sept. 2010, pp. 1 –8.

[13] J. Richter, N. Kuntze, and C. Rudolph, "Security Digital Evidence," in *2010 Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering*. IEEE, 2010, pp. 119–130.

[14] T. C. Group, "TCG Trusted Network Connect – TNC IF-MAP Binding for SOAP Version 2.0," www.trustedcomputing.org, 2010.

[15] J. v. H. I. Bente, J. Vieweg, "Towards Trustworthy Networks with Open Source Software," in *Horizons in Computer Science Volume 3*. Nova Science Publishers Inc., T. S. Clary (Eds.), 2011.

[16] M. Llanes, E. Prieto, R. Diaz, , L. Coppolino, A. Sergio, R. Cristaldi, M. Achemlal, S. Gharout, C. Gaber, A. Hutchison, and K. Dennie, "Scenario requirements (public version)," MASSIF Project, Tech. Rep. Deliverable D2.1.1, 2011.

[17] A. Fuchs and R. Rieke, "Identification of Security Requirements in Systems of Systems by Functional Security Analysis," in *Architecting Dependable Systems VII*, ser. LNCS. Springer, 2010, vol. 6420, pp. 74–96.

[18] N. Kuntze and C. Rudolph, "Secure digital chains of evidence," in *Sixth International Workshop on Systematic Approaches to Digital Forensic Engeneering*, 2011.

[19] N. Kuntze, D. Mähler, and A. U. Schmidt, "Employing trusted computing for the forward pricing of pseudonyms in reputation systems," in *Axmedis 2006, Proceedings of the 2nd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, Volume for Workshops, Industrial, and Application Sessions*, 2006.

 ISBN: 978-1-61208-184-7

New Approach to Mitigating Distributed Service Flooding Attacks

Mehmud Abliz* Taieb Znati*†
*Department of Computer Science
†Telecommunication Program
University of Pittsburgh
Pittsburgh, Pennsylvania 15260 USA
{mehmud, znati}@cs.pitt.edu

***Abstract*—Distributed denial of service (DDoS) attacks pose great threat to the Internet and its public services. Various computation-based cryptographic puzzle schemes have been proposed to mitigate DDoS attacks when detection is hard or has low accuracy. Yet, existing puzzle schemes have shortcomings that limit their effectiveness in practice. First, the effectiveness of computation-based puzzles decreases, as the variation in the computational power of clients increases. Second, while mitigating the damage caused by the malicious clients, the puzzle schemes also require the benign clients to perform the same expensive computation that doesn't contribute to any useful work from the clients' perspective. In this study, we introduce *guided tour puzzles*, a novel puzzle scheme that addresses these shortcomings. The guided tour puzzle scheme uses latency — as opposed to computational delay — as a way of forcing sustainable request arrival rate on clients. We evaluate the DoS mitigation effectiveness of the scheme in a realistic simulation environment, and show that guided tour puzzle scheme provides a strong mitigation of request flooding DDoS and puzzle solving DDoS attacks.**

Keywords- denial of service; availability; tour puzzles; proof of work; client puzzles; cryptography.

I. INTRODUCTION

A Denial of Service (DoS) attack is an attempt by malicious parties to prevent legitimate users from accessing a service, usually by depleting the resources of the server which hosts that service. DoS attacks may target resources such as server bandwidth, CPU, memory, storage, or any combination thereof. These attacks are particularly easy to carry out if a significant amount of server resource is required to process a client request that can be generated trivially. Cryptographic puzzles have been proposed to defend against DoS attacks with the aim of balancing the computational load of the server relative to the computational load of the clients [1] [2] [3] [4] [5].

In a cryptographic puzzle scheme, a client is required to solve a moderately hard computational problem, referred to as *puzzle*, and submit the solution as a proof of work before the server spends any significant amount of resource on its request. Solving a puzzle typically requires performing significant number of cryptographic operations, such as hashing, modular multiplication, etc. Consequently, the more a client requests service from the server, the more puzzles it has to compute, further expending its own computational resources. Puzzles are designed so that their construction and verification can be achieved with minimum server computational load in order to avoid DoS attacks on the puzzle scheme itself (attacks aimed at the puzzle scheme itself are thereafter referred to as *puzzle solving attacks.*).

Originally, cryptographic puzzles were proposed to combat spams [6]. They have then been extended to defend against other attacks, including DoS [1] [2] [5] [7] [8] and Sybil attacks [9] [10]. Furthermore, different ways of constructing and distributing puzzles have been explored [5] [11] [12] [13] [14]. Unfortunately, existing puzzle schemes have shortcomings that limit their effectiveness in defending against DoS attacks.

First, the effectiveness of computation-based puzzles decreases, as the variation in the computational power of clients increases. To illustrate this limitation, consider a system composed of a server whose capacity is R requests per second, N_l legitimate clients whose clock frequency is f, and N_m malicious clients whose clock frequency is $a \cdot f$, where a is a *disparity factor* that represents the degree of disparity between the CPU powers of malicious and legitimate clients. Furthermore, assume that legitimate clients can tolerate a maximum puzzle difficulty of D_{max}, expressed in terms of the number of instructions. The maximum protection the server can achieve against a DoS attack is by setting the puzzle difficulty to D_{max}. During an attack, the total load on the server is the sum of the loads generated by the legitimate and malicious clients, which can be expressed as $N_l \frac{f}{D_{max}} + N_m \frac{af}{D_{max}}$ (without loss of generality, we assume that when solving puzzles clients use their full CPU capacity). Therefore, to carry out a DoS attack against the server, an attacker must at least induce a load on the server that exceeds the server's full capacity, i.e. $N_l \frac{f}{D_{max}} + N_m \frac{af}{D_{max}} \geq R$. Using simple deductions, it is clear that the minimum number of malicious clients required to cause denial of service should satisfy the inequality $N_m \geq \frac{RD_{max} - N_l f}{af}$. Consequently, the minimum number of malicious clients required to stage a successful DoS attack against the server becomes smaller as the disparity factor a increases, decreasing the effectiveness of a puzzle-based defense in mitigating the DoS attacks.

Second, existing puzzle schemes may exact heavy compu-

 ISBN: 978-1-61208-184-7

tational penalty on legitimate clients, when the server load becomes heavy and increasing the computational complexity of the puzzle becomes necessary to prevent overloading the server. The negative impact of such a penalty is further compounded by the fact that the puzzle-induced computation does not usually contribute to the execution of any task that is useful to the client, thereby further wasting client resources and limiting the client's ability to carry out other computational activities.

In this paper, we propose a novel, latency-based puzzle scheme, referred to as *guided tour puzzle*, to address the shortcomings of current cryptographic puzzle schemes in dealing with DoS attacks. The guided tour puzzle scheme is the first to use network latency to control the rate of client requests and prevent potential DoS attacks on the server.

The rest of the paper is organized as follows. Section II describes the system model and the threat model used. Section III introduces the guided tour puzzle scheme. In Section IV, we use analysis and measurement to show that guided tour puzzles satisfy our requirements and design goals. The effectiveness of the guided tour puzzles in mitigating DDoS attacks is evaluated in Section V. Future plans for extending the guided tour puzzle scheme and conclusion of the paper are presented in Section VI.

II. System Model

A. System Overview

We consider an Internet-scale distributed system of clients and servers. A *server* is a process that provides a specific service, for example a Web server or an FTP server. A *client* is a process that requests service from a server. The term *client* and *server* are also used to denote the machines that runs the server process and the client process respectively. Clients are further classified as *legitimate clients* that do not contain any malicious logic and *malicious clients* that contain malicious logic. In the denial of service context, a malicious client attempts to prevent legitimate clients from receiving service by flooding the server with spurious requests. An *attacker* is a malicious entity who controls the malicious clients. We refer to a *user* as a person who uses a client machine.

B. Threat Model

The attacker attempts to disrupt service to the legitimate clients by sending apparently legitimate service requests to the server to consume its computational resources. We consider DoS attacks that flood the server with large amount of requests and attacks that attempt to thwart puzzle defense using massive computational resources. It is assumed that network resources are large enough to handle all traffic, and the resource under attack is server computation.

Our threat model assumes a stronger attacker than previous schemes do. First we assume the attacker may possess the best commercially available hardware and bandwidth resources. Meanwhile, the attacker can take maximum advantage of her resources by perfectly coordinating all of her available computation resources. Next, the attacker can eavesdrop on all messages sent between a server and any legitimate client. We assume that the attacker can modify only a limited number of client messages that are sent to the server. This assumption is reasonable since if an attacker can modify all client messages, then it can trivially launch a DoS attack by dropping all messages sent by all clients to the server. Finally, the attacker may launch attacks on the puzzle scheme itself, including puzzle construction, puzzle distribution, or puzzle verification.

III. Guided Tour Puzzle

A. Overview

When a server suspects that it is under attack or its load is above a certain threshold, it asks all clients to solve a puzzle prior to receiving service. In the guided tour puzzle protocol, the puzzle is simply a tour that needs to be completed by the client via taking round-trips to a set of special nodes, called *tour guides*, in a sequential order. We call this tour a *guided tour*, because the client does not know the order of the tour beforehand, and each tour guide has to direct the client towards the next tour guide for the client to complete the tour in the correct order. Each tour guide may appear zero or more times in a tour, and the term *stop* is used to represent a single appearance of a tour guide in a tour.

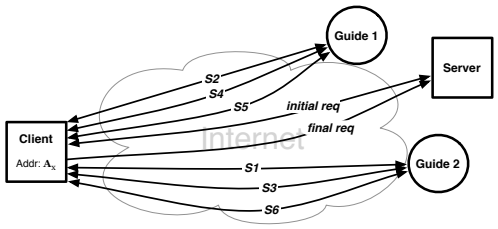


Figure 1. Example of a guided tour; the tour length is 6, and the order of visit is: $G_2 \rightarrow G_1 \rightarrow G_2 \rightarrow G_1 \rightarrow G_1 \rightarrow G_2$.

The tour guide at the first stop of a tour is randomly selected by the server, and will also be the last stop tour guide, i.e., a guided tour is a closed-loop tour. The tour guide at each stop randomly selects the next stop tour guide to visit. Starting from the first stop, the client contacts the tour guide at each stop and receives a reply. Each reply contains a token that proves to the next stop and the last stop that the client has visited this stop. Prior to sending its reply, the tour guide at each stop verifies that the client visited the previous stop tour guide, so that the client cannot contact multiple tour guides in parallel. After completing $L-1$ stops in a L-stop tour, the client submits the set of tokens it collected from all previous stops to the last stop tour guide (which is also the first stop tour guide), which will issue the client a proof that it completed the tour. The client then sends this

 ISBN: 978-1-61208-184-7

proof to the server, along with its service request, and the server grants the client service if the proof is valid. Figure 1 shows an example of a guided tour with two tour guides and 6 stops.

B. Basic Scheme

We set up N tour guides in the system, where $N \geq 2$. The server keeps a secret k_S that only it knows, and a set of keys $k_{S1}, k_{S2}, \ldots, k_{SN}$ are shared between the server and each tour guide. Each tour guide G_i maintains a pairwise shared key $k_{i,j}$ with every other tour guide G_j, where $i \neq j$ and $1 \leq i, j \leq N$. The total number of keys need to be maintained by each tour guide or the server is N, and this key management overhead is acceptable since N is usually a small number between 2 and 20. The tour length L is decided by the server to adjust the puzzle difficulty. Notations are summarized in Table I. The four steps of the guided tour puzzle protocol is described below.

Table I
NOTATION SUMMARY.

N	Number of tour guides in the system
G_j	j-th tour guide ($1 \leq j \leq N$)
k_S	Secret key only known to the server
k_{Sj}	Shared key between the server and G_j
$k_{i,j}$	Shared key between G_i and G_j ($i \neq j$)
L	Length of a guided tour
A_x	Address of client x
i_s	Index of the s-th stop tour guide ($1 \leq i_s \leq N$)
t_s	Timestamp at the s-th stop of the tour
R_s	Client puzzle solving request at s-th stop
B	Size of the $hash$ digest in bits

1) Service request: A client x sends a service request to the server. If the server load is normal, the client's request is serviced as usual; if the server is overloaded, then it proceeds to the next step — initial puzzle generation.

2) Initial puzzle generation: The server replies to the client x with a message that informs the client to complete a guided tour. The reply message contains $\{L, i_1, t_0, h_0, m_0\}$, where i_1 is the uniform-randomly selected index of the first stop tour guide, t_0 is a timestamp, h_0 is a hash value, m_0 is a message authentication code (MAC). The value of h_0, m_0 are computed as follows:

$$h_0 = hash(A_x \mid\mid L \mid\mid i_1 \mid\mid t_0 \mid\mid k_S) \quad (1)$$

$$m_0 = hash(A_x \mid\mid L \mid\mid i_1 \mid\mid t_0 \mid\mid h_0 \mid\mid k_{Si_1}) \quad (2)$$

where, $\mid\mid$ means concatenation, A_x is the address (or any unique value) of the client x, and $hash$ is a cryptographic hash function such as Secure Hash Algorithm - 1 (SHA-1) [15]. Since m_0 is computed using the key k_{Si_1} that is shared between the first stop tour guide G_{i_1} and the server, it enables G_{i_1} to do integrity checking later on.

3) Puzzle solving: After receiving the puzzle information, the client visits the tour guide G_{i_s} at each stop s, where $1 \leq s \leq L$, and receives a reply. Each reply message contains $\{h_s, m_s, i_{s+1}, t_s\}$, where i_{s+1} is the uniform-randomly selected index of the next stop tour guide, t_s is the timestamp at stop s, and h_s, m_s are computed as follows:

$$h_s = hash(h_0 \mid\mid A_x \mid\mid L \mid\mid s \mid\mid i_s \mid\mid i_{s+1} \mid\mid k_{i_s,i_1}) \quad (3)$$

$$m_s = hash(m_{s-1} \mid\mid A_x \mid\mid L \mid\mid s \mid\mid i_s \mid\mid i_{s+1} \mid\mid k_{i_s,i_{s+1}}) \quad (4)$$

At each stop s, the client sends a puzzle solving request message R_s that contains $\{h_0, L, s, t_{s-1}, m_{s-1}, i_1, i_s\}$ to the tour guide G_{i_s}, and the tour guide G_{i_s} replies to the client only if m_{s-1} is valid. In other words, each stop enforces that the client correctly completed the previous stop of the tour.

At the $(L-1)$-th stop, the tour guide $G_{i_{L-1}}$ knows that the next stop is the last stop, and replaces i_{s+1} with i_1 (recall that the first stop i_1 is also the last stop) when computing h_s and m_s. After completing the $(L-1)$-th stop, the client computes h_L as follows:

$$h_L = h_1 \oplus h_2 \oplus \ldots \oplus h_{L-1} \quad (5)$$

where $\oplus$ means exclusive or, and submits $\{h_0, h_L, L, m_{L-1}, i_1, i_2, \ldots, i_L\}$ to the first stop tour guide G_{i_1}. Using these information, G_{i_1} can compute $h_1, h_2, \ldots, h_{L-1}$ using formula (3), and subsequently h_L using formula (5). Note that only G_{i_1} can compute hash values h_1 to h_{L-1}, since only it knows the keys k_{i_1,i_2} to $k_{i_1,i_{L-1}}$ that are used in the hash computations.

If the h_L submitted by the client matches the h_L computed by G_{i_1} itself, then G_{i_1} sends back the client a value h_{sol} that can prove to the server that the client did complete a tour of length L. The hash value h_{sol} is computed as follows:

$$h_{sol} = hash(h_0 \mid\mid A_x \mid\mid L \mid\mid t_L \mid\mid k_{Si_1}) \quad (6)$$

4) Puzzle verification: The client submits to the server $\{h_0, h_{sol}, t_0, t_L, i_1\}$ along with its service request, and the server checks to see if h_0 and h_{sol} that it computes using formulas (1) and (6) matches the h_0 and h_{sol} submitted by the client. If both hash values match, the server allocates resources to process the client's request.

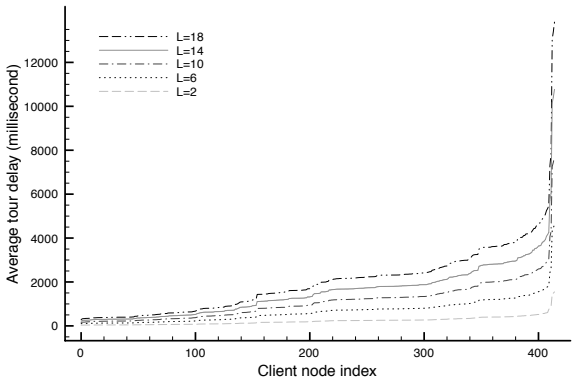
IV. ANALYSIS

In this section, we use analytical reasoning and experimental results to demonstrate that guided tour puzzles are not effected by the disparity in the clients' computational power and minimizes the useless work required for clients.

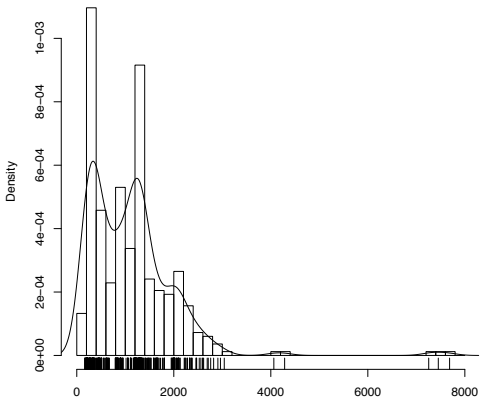
A. Minimize the Effect of Computational Power Disparity

The guided tour puzzle scheme is not effected by the disparity in the computational power of clients. This is because the round trip delays that consist the puzzle solving time of a puzzle are mostly determined by the network(s) between the client and the tour guides, and the clients' CPU, memory, or bandwidth resources have minimal impact on them. As the data that needs to be transferred between client and tour guides is trivial in size, the bandwidth of the end hosts does not effect the round trip delay.

 ISBN: 978-1-61208-184-7



(a) Average tour delays of two-week period, $N = 4$.



(b) PDF of tour delay (unit: millisecond) when $N = 4$

Figure 2. The tour delays of clients when 4 tour guides are used.

Since it is possible that the round trip delays from different clients to a tour guide may have significant variation, it is possible that the sum of round trip delays — referred to as *tour delay* — for different clients differ significantly. Next, we use experimental and analytical analysis to show that this variation is small, compared to the variation in the puzzle solving times of computation-based puzzles.

1) Experimental Analysis: We use a two-week long measurement data collected from little over 400 machines on PlanetLab [16] to show that the variation in tour delay across clients is within a small factor for a large distributed system.

We first randomly chose 20 nodes, out of the 400 nodes, as candidates for tour guides. The remaining nodes are used as client nodes. The number of tour guides N is varied from 4 to 20, and the tour length L is varied from 2 to 18. For each (N, L) setting, guided tours are generated for all client nodes. The tour delay at a given time is computed buy summing the corresponding round trip delays in that time period. To find the average tour delay of a client for a specific (N, L) setting, the two-week long tour delays of that client for that setting is averaged. Next, the average tour delays are sorted by least-to-most to provide a better view of the delay variation across clients. Figures 2(a) shows the average tour delays computed using this method for all client nodes when N = 4. Results for other values of N are skipped due to the space limitation, but they are very similar to the results shown in Figures 2(a). The ratio of the largest and the smallest tour delays is around 5, when 5% outliers are excluded. This disparity is several orders of magnitude smaller when compared to the disparity in available computational power (which can be in thousands [11] [12]). Figure 2(b) shows that majority of tour delays are clustered within a small area of delay and the distribution of tour delays closely reflect a normal distribution. The probability density curve is concave around 1000 milliseconds, as fewer nodes complete the tour in around 1000 millisecond compared to nodes that do in about 500 and 1500 milliseconds.

2) Analytical Analysis: Since the majority of the PlanetLab machines are connected to the Internet through campus networks, the delay data may not sufficiently reflect the diverse access network technologies that are used for connecting end hosts to the Internet. Next, we use latency data from the existing literature to show that even when clients are connected to the Internet using access technologies that provide very different delay properties, the disparity in their end-to-end round trip delays is several times smaller than the disparity in the computational power.

Let us take four very common access network technologies with very different delay characteristics: 3rd generation mobile telecommunications (3G), Asymmetric Digital Subscriber Line (ADSL), cable, and campus Local Area Network (LAN). The average access network delays are $200ms$ for 3G [17], $15 \sim 20ms$ for ADSL and cable [18], [19], and in the order of $1ms$ or negligible for campus LANs (here, we refer to the access network delay as the round-trip delay between the end host and the edge router of the host's service provider; this latency is usually measured by measuring the round-trip delay to the first pingable hop). Based on the measurement analysis of the Internet delay space [20], the delay space among edge networks in the Internet can be effectively classified into three major clusters with average round trip propagation delays of about $45ms$ for the North America cluster, $135ms$ for Europe cluster, and $295ms$ for Asia cluster. Using these edge to edge propagation delay values and the average access network delay values, we can compute an average end to end round trip delays of 245, 335, and 495 ms for 3G hosts, 65, 155, and 315 ms for DSL & cable hosts, and 45, 135, and 295 ms for campus LAN hosts. The biggest disparity occurs between the hosts in the Asia cluster that connect through 3G and the hosts in the US cluster that connects through campus LAN, and the ratio of their round trip delays is $495ms/45ms = 11$. This disparity is about 4 times smaller than the low estimate of computational disparity provided in

 ISBN: 978-1-61208-184-7

[21]. The round trip delays may get higher than $495ms$ due to congestion and high queuing delays in the intermediate routers. However, these congestions and high queuing delays effects all packets, regardless of whether they are from malicious clients or legitimate clients.

B. Minimize Interference and Useless Work

In guided tour puzzle scheme, a client has to perform only one type of operation: sending packets to tour guides. To complete a guided tour puzzle with tour length L, a client only needs to send and receive a total of $2 \times L$ packets with a data payload less than 100 bytes per packet. Since L is usually a small number below 30, this creates negligible CPU and bandwidth overhead even for resource-constrained devices such as cellular phones.

V. Study of DDoS Defense Efficacy

In this study, we focus our evaluation on the ability of guided tour puzzles in preventing the application layer DDoS (also referred to as *distributed service flooding*) attacks. We show that the guided tour puzzle scheme provides an optimal defense against request flooding attacks and a near optimal defense against puzzle solving attacks.

A. Simulation Setup

We use Network Simulator 2 (NS-2) [22] to achieve a practical simulation environment. A topology with $5,000$ nodes is generated using Inet-3.0 [23] to closely simulate large-scale wide area networks, such as the Internet. The bandwidth and the link delay values are calculated based on the Inet-3.0 generated link distance values. The link and queueing delays are set differently for different links, therefore the round trip delays, and consequently the tour delays, of different nodes will be very different.

As clients, tour guides, and server nodes will be located in the edge in real networks, we use degree-one nodes in the topology as the client, server, and tour guide nodes. The topology contains a total of $1,922$ degree-one nodes. We randomly choose a degree-one node as the server node and another 20 degree-one nodes as candidates for tour guides. The remaining $1,901$ degree-one nodes are all used as client nodes, including legitimate and malicious client nodes. The number of malicious client nodes is varied from 0% to 90% with an increment of 10%, and the server load is increased from 0.96 to 8.74 correspondingly.

A simulation model of the guided tour puzzle scheme is developed in NS-2. As the Internet traffic is self-similar and the self-similar traffic can be generated by multiplexing ON/OFF sources that have fixed rates in the ON periods and heavy-tail distributed ON/OFF period lengths [24] [25], each client is implemented as an ON/OFF source with ON/OFF period lengths are taken from a Pareto distribution to simulate the Internet traffic. On average, each legitimate client sends 1 request every 2 seconds, and each malicious client sends 10 times the rate of a legitimate client. The server capacity is set to $1,000$ requests per second, such that the server's full capacity can be reached when setting all clients as legitimate. The server load increases by 96% with each 10% increase of the percentage of malicious clients. Using the average estimated client request rate of 0.5 request per second and the server CPU rate of $1,000$ requests per second, we can compute that the expected utilization of the server is $\frac{0.5 \times 1901}{1000} = 0.9505$ when all clients are legitimate. We achieved a utilization of 0.9656 for this setting in our experiments, which validates the correctness of our simulation setup.

Three evaluation metrics are used: average completion time per legitimate request, legitimate utilization of the server, and legitimate request drop rate. The average completion time is calculated by averaging the time spent between sending of a request and the receiving of its response, including the time spent on solving puzzles, for all completed requests of all the legitimate clients. The legitimate utilization of the server is computed as the fraction of the time the server's CPU is processing the requests of legitimate clients. The legitimate request drop rate is computed by dividing the total number of dropped legitimate requests by the total number of legitimate requests sent.

We experimented with two types of attacks: the flooding attack and the puzzle solving attack. In a flooding attack, a malicious client sends requests at a high rate and ignores the server's request for solving puzzles; whereas in the puzzle solving attack, a malicious client solves puzzles as fast as they can to send requests at the maximum speed possible.

B. Simulation Results

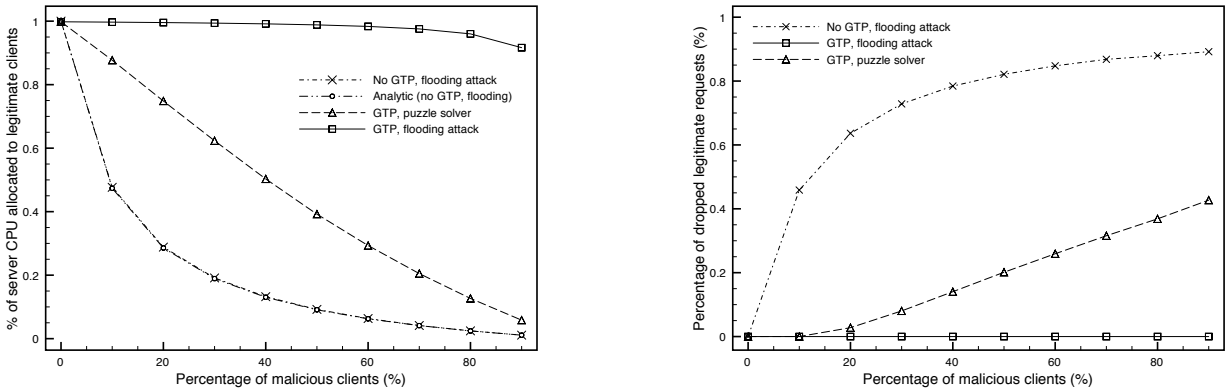
The first set of simulations are conducted with a fixed tour length of 8 and using 4 tour guides. The results are reported in Figure 3.

1) Server CPU utilization: Figure 3(a) illustrates the improvement in the legitimate utilization of the server. As the curve "No GTP (), flooding attack" (GTP stands for Guided Tour Puzzle) indicates, the legitimate clients' share of the server's CPU capacity drops rapidly as the percentage of attackers increases when no guided tour puzzle is used. The legitimate utilization of the server in this case is predominantly decided by the ratio of total number of legitimate requests to the total number of requests. This can be validated by computing the percentage of legitimate requests for different settings using the following formula:

$$\frac{r \times (1-x) \times N_c}{r \times (1-x) \times N_c \; + \; 10 \times r \times x \times N_c} = \frac{1-x}{1+9x} \qquad (7)$$

where, r denotes the request rate of legitimate clients, N_c is the total number of client nodes, x is the percentage of malicious nodes, and $10 \times r$ is the malicious request rate. The curve "Analytic (no GTP, flooding attack)" is then

 ISBN: 978-1-61208-184-7



(a) Legitimate utilization of the server

(b) Legitimate request drop rate

Figure 3. The effectiveness of guided tour puzzle against flooding attacks and puzzle solving attacks, $N = 4, L = 8$.

computed using Formula (7), and it overlaps perfectly with the experiment results from the NS-2 simulation for the case of "No GTP, flooding attack".

The curve "GTP, flooding attack" in the Figure 3(a) shows that using guided tour puzzle eliminates the impact of flooding attackers entirely. In this scenario, the malicious clients do not solve any puzzle, but send requests that include fake puzzle solutions at a high rate in an attempt to consume as much server CPU capacity as possible. The slight decrease in the legitimate clients' utilization of the server CPU as the percentage of attackers increases is due to the increase in the percentage of server's CPU capacity allocated to verifying puzzle solutions. We intentionally used a low estimate of 10^6 hash operation per second as the server's hash computation rate to protrude the cost of puzzle solution verification.

The last curve "GTP, Puzzle solver" in Figure 3(a) is corresponding to the attack targeted at the guided tour puzzle scheme itself. It shows that, when the guided tour puzzle scheme is used, the legitimate utilization of the server is roughly equal to the percentage of legitimate clients in the system. We argue that without being able to identify malicious clients, the best a DoS mitigation scheme can achieve is to treat every client equally and fairly allocate the server's CPU to all clients that are requesting service.

2) Request drops: Figure 3(b) shows the legitimate request drop rate. When no guided tour puzzle is used, the flooding attack caused legitimate clients to drop most of their requests as the curve "No GTP, flooding attack" indicates. When the percentage of attacker is increased to 90%, near 100% of legitimate requests are dropped as a result of the flooding attack. After switching to use guided tour puzzles (curve "GTP, flooding attack"), the percentage of dropped requests becomes zero under the flooding attack, including when the 90% of the clients are malicious. In the puzzle solving attacks, guided tour puzzle scheme reduces the legitimate request drops by more than half in all cases and to zero in some cases. This legitimate request drops can be eliminated entirely by optimally adjusting the tour length, as the simulation results in the next section show.

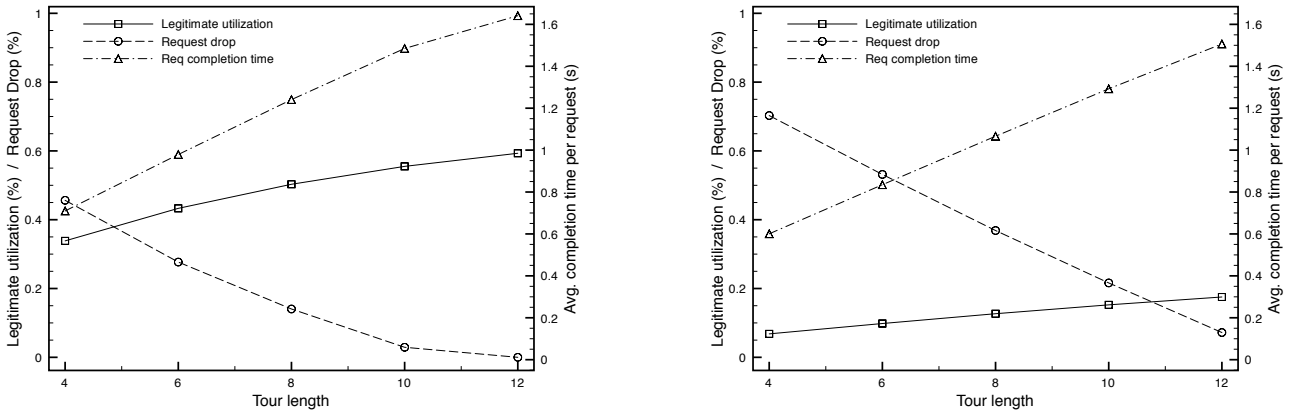
3) Effect of tour length: The tour length in guided tour puzzles is critical for the optimality of the guided tour puzzle defense, especially for the legitimate clients' utilization of server CPU in the case of puzzle solving attacks. The next set of simulation experiments are conducted to measure the effect of tour length on utilization, request completion time, and request drops in the case of puzzle solving attacks. These experiments are conducted using 4 tour guides and 40% and 80% of malicious clients respectively.

The response of various metrics to the change in tour length is illustrated in Figure 4. As the tour length increases, the legitimate utilization of the server (curve "Legitimate utilization") and the request completion time (curve "Req completion time") increase, while the legitimate request drop rate (curve "Request drop") decreases. After increasing the tour length to 12, the legitimate request drop rate becomes zero, and the legitimate utilization of the server becomes optimal in both cases of 40% and 80% malicious clients. Here "optimal" means legitimate clients are granted the amount of server CPU capacity that is equal to the percentage of legitimate clients in the system. Further increasing the tour length does not improve the utilization and request drop metrics and decreases the total utilization of the server CPU, as well as increases the request completion time. The increase in the request completion time is evident since larger tour length means more round trips between clients and tour guides. These observations show that choosing the right tour length is important in achieving optimal DDoS mitigation results and providing better trade-off between mutually restricting performance metrics.

VI. Conclusion and Future Work

In this paper, we showed that existing cryptographic puzzle schemes become less effective as the variation in the computational power of clients increases, and that they

 ISBN: 978-1-61208-184-7



(a) 40% clients are malicious, $N = 4$

(b) 80% clients are malicious, $N = 4$

Figure 4. The effect of the tour length on the effectiveness of the guided tour puzzle defense.

require benign clients to perform the same expensive computation that doesn't contribute to any useful work. To this end, we introduced the guided tour puzzle scheme, and showed that it addresses the shortcomings of the existing puzzle schemes and achieves better protection against DDoS attacks. Meanwhile, using extensive simulation studies, we showed that guided tour puzzle is effective in mitigating distributed service flooding attacks and that it is a practical solution to be adopted.

As future work, we would like to further improve the guided tour puzzle scheme in terms of the following. First, we would like to eliminate the need for the server's involvement in the puzzle generation process. Second, further investigation is needed to find out optimal ways to position tour guides in the network. Last but not least, we would like to devise an optimal strategy for adjusting the tour length.

REFERENCES

[1] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *NDSS '99*, San Diego, CA, 1999, pp. 151–165.

[2] W. Feng, E. Kaiser, and A. Luu, "The design and implementation of network puzzles," in *IEEE INFOCOM '05*, 2005.

[3] T. Aura, P. Nikander, and J. Leiwo, "DoS-resistant authentication with client puzzles," in *8th International Workshop on Security Protocols*, vol. 2133, 2000, pp. 170–181.

[4] D. Dean and A. Stubblefield, "Using client puzzles to protect TLS," in *10th USENIX Security Symposium*, 2001, pp. 1–8.

[5] X. Wang and M. K. Reiter, "Defending against denial-of-service attacks with puzzle auctions," in *IEEE Symposium on Security and Privacy*, Oakland, 2003, pp. 78–92.

[6] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *CRYPTO '92*, 1992, pp. 139–147.

[7] X. Wang and M. K. Reiter, "Mitigating bandwidth-exhaustion attacks using congestion puzzles," in *CCS '04*, 2004.

[8] B. Waters, A. Juels, J. A. Halderman, and E. W. Felten, "New client puzzle outsourcing techniques for dos resistance," in *11th ACM CCS*, 2004, pp. 246–256.

[9] N. Borisov, "Computational puzzles as sybil defenses," in *the 6th IEEE International Conference on Peer-to-Peer Computing*, 2006, pp. 171–176.

[10] H. Rowaihy, W. Enck, P. Mcdaniel, and T. L. Porta, "Limiting sybil attacks in structured p2p networks," in *the IEEE INFOCOM '07*, 2007, pp. 2596–2600.

[11] M. Abadi, M. Burrows, M. Manasse, and T. Wobber, "Moderately hard, memory-bound functions," in *NDSS '03*, 2003.

[12] C. Dwork, A. Goldberg, and M. Naor, "On memory-bound functions for fighting spam," in *CRYPTO '03*, 2003.

[13] M. Ma, "Mitigating denial of service attacks with password puzzles," in *International Conference on Information Technology*, vol. 2, Las Vegas, 2005, pp. 621–626.

[14] B. Groza and D. Petrica, "On chained cryptographic puzzles," in *3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence*, Timisoara, Romania, 2006.

[15] *Secure Hash Standard*, National Institute of Standards and Technology (NIST) Std., 1995.

[16] "About planet lab," Planet Lab. [Online]. Available: http://www.planet-lab.org/about

[17] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl, "Anatomizing application performance differences on smartphones," in *MobiSys '10*, 2010, pp. 165–178.

[18] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu, "Characterizing residential broadband networks," in *IMC '07*, 2007, pp. 43–56.

[19] M. Yu, M. Thottan, and L. Li, "Latency equalization as a new network service primitive," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, p. 1, May 2011.

[20] B. Zhang, T. S. E. Ng, A. Nandi, R. H. Riedi, P. Druschel, and G. Wang, "Measurement-based analysis, modeling, and synthesis of the internet delay space," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 229–242, 2010.

[21] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y. Hu, "Portcullis: Protecting connection setup from denial-of-capability attacks," in *SIGCOMM '07*, 2007, pp. 289–300.

[22] VINT, "The network simulator - ns-2," 2009.

[23] J. Winick and S. Jamin, "Inet-3.0: Internet topology generator," Univ. of Michigan, Tech. Rep. CSE-TR-456-02, 2002.

[24] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[25] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, 1995.

 ISBN: 978-1-61208-184-7

Architecture of a Security and Surveillance System

The benefits of an open and generic approach

Florian Segor, Axel Bürkle, Sven Müller, Rainer Schönbein, Matthias Kollmann
IAS – Interoperabilität und Assistenzsysteme
Fraunhofer IOSB
Karlsruhe, Germany
{florian.segor, axel.buerkle, sven.mueller, rainer.schoenbein, matthias.kollmann}@iosb.fraunhofer.de

***Abstract*—During the recent years, it has been increasingly shown that open and generic system platforms exhibit considerable advantages over closed systems in different areas of security technology. With systems that are flexible and adaptable to emerging demands, it can avoid expensive dedicated solutions becoming useless when requirements change. In this paper, a universal infrastructure is presented. It is shown exemplarily how easy and cost-effective new demands can be met by integrating new software and hardware components. The described architecture is designed to achieve maximum reusability.**

Keywords - generic; system architecture; security system; control station

I. INTRODUCTION

In order to be adaptable to a wide range of different requirements and applications, the complex surveillance system AMFIS [1] (**A**ufklärung mit **M**iniatur**f**luggeräten **i**m **S**ensorverbund) presented in this paper was developed as a mobile and generic system, which delivers an extensive situation picture in complex surroundings - even with the lack of stationary security technology. In order to achieve maximum flexibility, the system is implemented open and mostly generalized so that different stationary and mobile sensors and sensor platforms can be integrated with minimal effort, establishing interoperability with existing and future assets. The system is modular and can be scaled arbitrarily or be tailored by choosing the modules suitable to the specific requirements.

After a short introduction into the AMFIS framework an overview of the application scenarios is presented, followed by a detailed description of the AMFIS architecture in section IV. Section V introduces the integration of a new sensor. The paper closes with conclusions and future work.

II. THE AMFIS FRAMEWORK

The AMIFS system consists of a universal ground control station and a customizable set of sensors and sensor carriers (see Fig. 1). In addition, there are interfaces to external exploitation stations and control centers.

The ground control station is an adaptable prototype system for managing data acquisition with various sensors, mobile ad hoc networks and mobile sensor platforms. The main tasks of the ground control station are to work as an ergonomic user interface and as a data integration hub between multiple sensors and a super-ordinated control center. The sensors can be stationary or mounted on moving platforms such as micro UAVs [2], unmanned ground vehicles (UGVs) or underwater vehicles. The system includes means to control different kinds of mobile platforms and to direct them to potentially interesting locations especially in areas with no prior sensor equipment. The actual AMFIS system is highly mobile and operational at any location with relative ease. The sensor carriers of this multi-sensor system can be combined in a number of different configurations to meet a variety of specific requirements. The functions of the ground control station include: task

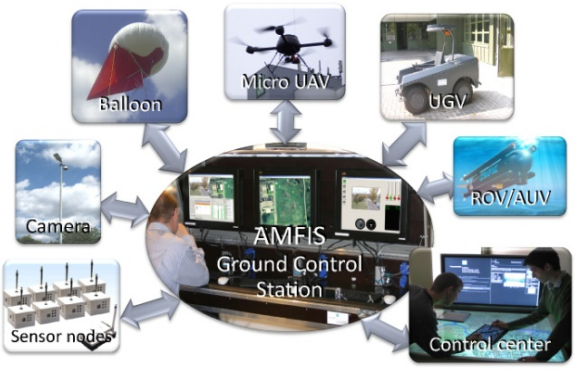


Fig. 1 Modular Ground Control Station AMFIS.

management, mission planning, control of sensors and mobile platforms, situation awareness, fusion and exploitation of sensor data, reporting, generation of alarms and archiving.

III. APPLICATION SCENARIOS

The sense of security of our society has significantly changed during the past years. Besides the risks arising from natural disasters there are dangers in connection with criminal or terroristic activities, traffic accidents or accidents in industrial environments. Especially in the civil domain in the event of big incidents, there is a need for a better data basis to support the rescue forces in decision making. The search for buried people after building collapses or the

 ISBN: 978-1-61208-184-7

clarification of fires at big factories or chemical plants are possible scenarios addressed by the AMFIS system.

Many of these events have very similar characteristics. They cannot be foreseen in their temporal and local occurrence so that situational in situ security or supervision systems are not present. The data basis, on which decisions can be made, is rather slim and therefore the present situation is very unclear to the rescue forces at the beginning of a mission. Precisely in these situations, it is extremely important to understand the context as fast as possible in order to initiate the suitable measures specifically and efficiently.

Applications of the AMFIS system include support of fire-fighting work with a conflagration, clarification of the debris and the surroundings after building collapses and search for buried or injured people. Additionally, the system can be used to support the documentation and perpetuation of evidence during the cleaning out of the scene at regular intervals.

```
<message key="string" type="string">
  <parentmessage key="string" />
  <timestamp>YYYY-MM-DD hh:mm:ss</timestamp>
  <originator type="SENSOR|SENSORNODE|USER|SYSTEM">
    <sensornodeid>bigint</sensornodeid>
    <sensorid>bigint</sensorid>
    <userid>bigint</userid>
  </originator>
  <subject
type="SENSORNETWORK|SENSORNODE|SENSOR|SYSTEM">
    <sensornetwork>bigint</sensornetwork>
    <sensornodeid>bigint</sensornodeid>
    <sensorid>bigint</sensorid>
  </subject>
  <value>value</value>
</message>
```

Fig. 2 XML Example of an AMFIS Message.

IV. AMFIS ARCHITECTURE

In order to create an open and generic system, one of the most important objectives was to design a sustainable architecture. One of the central demands for the system is the flexibility regarding new hardware, software or sensor components. The tasks, for which the AMFIS system is developed are varied and the technological development during the last years in related fields, e.g., mini and micro UAVs are enormous. Therefore it is obvious that the AMFIS architecture must be able to master new demands resulting from future assets.

To achieve this flexibility, the architecture was designed to not only be adaptable to components unknown today but also to be achieved with low expenditure.

The AMFIS ground control station's software architecture is basically 3-tiered following a pattern similar to the MVC (Model-view-controller) paradigm best known from web application development.

The central application is the so-called AMFIS Connector (see Fig. 3), which is a message broker responsible for relaying metadata streams within the network.

To be able to manage a vast amount of sensors and sensor carriers, the physical sensors and sensor carriers are logically mapped on the so-called sensorweb, a tree structure which contains virtual representatives of the actual units. The root node (sensorweb) is connected with a row of sensor networks, for example a set of PTZ cameras (Pan-Tilt-Zoom cameras). Each of these sensor networks consists of one or several sensor nodes, which correspond in each case to a physical sensor (for example a single PTZ camera). The sensor nodes themselves may again contain different sensors, for example a camera contains a compass.

The sensorweb is stored permanently in a database, from which an XML document is generated at runtime. This is also done by the AMFIS Connector, which can be seen as the central information service of the ground control station.

The communication protocol within AMFIS is based on XML strings (see Fig. 2), which are sent to TCP-Sockets. To simplify the use of this protocol and to provide different possibilities for software development, different implementations for different runtime environments (e.g. .NET, Java) are available, which enclose the XML management and allow an object oriented view of the messages to the user.

The implementation of the communication protocol is multicast-oriented; every incoming message is passed on by the Connector to all connected client applications. Each application decides itself if and how these messages are processed.

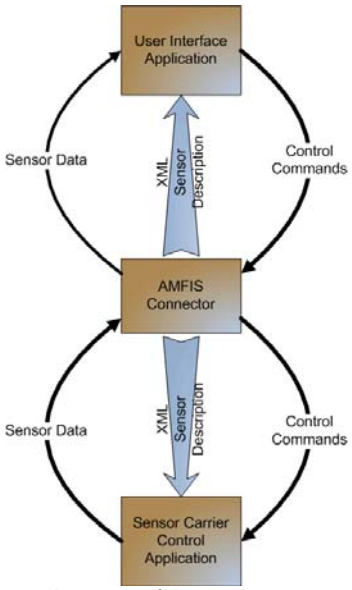


Fig. 3 Architecture of the AMFIS Framework.

If a client application connects to the AMFIS Connector, it first receives the XML structure of the sensorweb followed by a steady stream of XML messages. Each of these messages contains metadata (e.g., sensor status data or control information), which originates from one of the sensors in the sensorweb or is meant to change the status of one or more sensor nodes. After successful establishment of

 ISBN: 978-1-61208-184-7

a connection, the Connector supplies the client application with a constant stream of live sensor data.

A client application in this context is any application that either includes one of the numerous AmfisCom implementations (.NET, Qt, Java) or implements the AMFIS message protocol directly.

The communication library (AmfisCom) builds the object tree from the XML data, providing the application developer with a type-safe and object-oriented view of the network of sensors and sensor carriers.

All applications within the AMFIS system, from graphic user interfaces to background system services, are designed as client applications:

- The various GUI applications of the user interface, most importantly the analyst's interface, the situation overview, the Photo Flight [3] or the pilot's interface. Those applications offer a visual representation of received metadata to the user, for example by displaying the current geographical locations of the various sensor carriers in the map and transmit commands to the sensor carriers, e.g., a user-generated waypoint for a UAV.
- A number of services running in the background, notably the video server, offering time shifting and archiving for both video and metadata (more on video management within AMFIS see below) and the rule engine respectively the multi-agent system, both supporting the user by automating certain processes.
- Drivers for various sensor carriers, e.g., a dedicated control software for UAVs, which translates high-level flight commands like waypoints into the proprietary RS232-based control protocol of the respective drone and in turn generates metadata XML status messages containing the current position, heading, remaining flight time etc.
- Interfaces to third-party applications or networks, e.g., command and control centers.

While all metadata is sent as XML messages irrespective of their type (whether they are sensor measuring values, steering information or user generated announcements) the large amount of generated video data must be processed and stored differently.

To do so, the connector is tightly coupled with a server application called the video server. It is responsible for storing and distributing video streams, serving the dual purpose of providing time shifting capabilities to the network as well as reducing the load on the usually wireless links between sensor carriers and the ground control station. Since time shifting or archiving is not always required, this functionality was not integrated in the Connector itself in order to keep it as light-weight as possible. To store the generated mission data permanently, the video server is connected to a database, from which the data streams can be restored for playback.

In order to be able to transmit reconnaissance results to external systems, the stored video data or the live video streams can be accessed externally. The main disadvantage of this method is the lack of metadata generated in the AMFIS system along with the video stream. For reconnaissance tasks, additional data such as location, time, sensor carrier or sensor type are often vital. Hence, the video server offers the possibility to convert the video data into standardized video formats, which contain these metadata. This is done by the AMFIS Transcoder Process, which encodes the metadata and the video data into a STANAG 4609 compliant data stream. That way, not only the imagery but also corresponding additional informations are available to systems, which can handle this standardized data format. The video streams generated by the AMFIS Transcoder can be stored as video clips in a CSD (coalition shared database) [4] or be transferred in real-time to an exploitation system such as ABUL [5].

To be able to receive messages or data information requests from an external system on top of the possibility to publish information, a communication module was implemented, which is called the XMPP Client. The XMPP Client translates reconnaissance requests (e.g. a Region-Of-Interest) placed by external systems into the AMFIS message format.

V. SENSOR INTEGRATION

With the software architecture described in section IV, it is possible to integrate new sensors and sensor carriers as well as completely new technologies without changes in the system's basic structure.

The integration of a new technology of this structure is described exemplarily in the following sections. It explains the use of a smartphone as a new sensor in the AMFIS system.

The mobile device is integrated as a client and allows the user to access data of the AMFIS system. Additionally, it provides tools to acquire and generate sensor data and to feed the accumulated data into the overall system. Therefore, the person carrying the mobile device becomes a mobile sensor in AMFIS.

The functional structure of the new sensor is basically divided into three modules. Besides the sensor itself, it

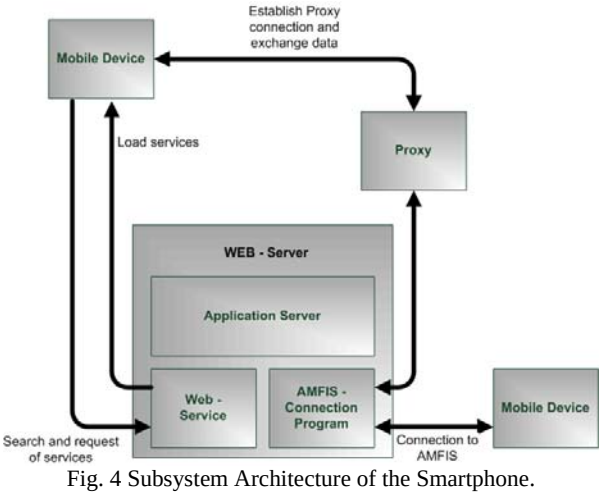


Fig. 4 Subsystem Architecture of the Smartphone.

 ISBN: 978-1-61208-184-7

affects the web server, which is responsible for supplying services and the application server, which structures the communication between the sensor and the system.

Providing the services by using a web server allows an easy addition of new functions to the sensor. By the separation between processing of data and the communication infrastructure it becomes possible to insert additional and more computationally intensive functions, which can be transferred to another computer system if necessary.

Splitting the architecture into component-based modules enables the development of the subsystems without affecting the already existing services. In addition, platform independence is achieved by using the functionalities of a web service. Thus, different types of smartphones with different operating systems can be used (e.g., Android, iOS, Windows phone, etc.) without having to change the data processing functions. Only the client software on the mobile device has to be adapted.

A. Smartphone

To gain a maximum degree of independence from suppliers and other external influence and to be able at the same time to benefit from the possibilities of an up-to-date smartphone, the Samsung Nexus S was used as a prototype mobile device for AMFIS. It runs the Android™ 2.3 (“Gingerbread”) operating system and has a 1 GHz processor.

In the first stage of development the functionality of an off-line GIS (Geographical Information System) [6] was integrated. This offers the possibility of a spatial orientation to the user regardless of a data connection to the internet or the AMFIS control station. In addition, the data communication is designed so that information from AMFIS can be shown on the map application when a connection is available. This concerns, for example, the geographical positions of other sensors or sensor carriers and additional information or instructions such as the request to move to a certain location for reconnaissance. Furthermore, it is possible to feed back own sensor data into the AMFIS system. At present, this includes own position data and the generation of images with the corresponding metadata.

To establish a connection between the smartphone and AMFIS, the available communication possibilities of the mobile device are used. In many scenarios it can be assumed that the user remains in immediate vicinity of the ground station AMFIS. In these cases the connection is established by using Wi-Fi. However, a shortfall of connectivity cannot be ruled out. Hence the data transfer via GSM/UMTS is realized as an alternative communication method.

Nevertheless, the complete loss of the connection has to be taken into account, too. Therefore, the sensor can also function as an independent stand-alone sensor. In this case the accumulated data is buffered and transferred automatically with a re-established connection.

B. Web server

Since the new sensor and its architecture are bound to already existing infrastructure and the independency from any operating system should be preserved, a service-oriented approach was chosen.

By implementing this solution a later integration of other platforms such as the Apple iPhone or a Windows phone device is possible since only the application has requires adaption and/or compilation for the new platform. The functions and processes of the web service or the connection management system for AMFIS do not require any changes.

The web server provides the connection to the internet and offers the runtime environment for the application server. It can be used to pre-filter packages or carry out the authentication for the mobile device. The basic infrastructure is thereby provided by an Apache Http server.

Among other features two web services were implemented to provide the basic functions.

One service is used to transmit the user's data (e.g., photos) to AMFIS. The other one is an update service, which can be used by the smartphone to establish a connection.

The first service for transferring user's data (e.g., photos and text news) is designed as a one way connection to be used exclusively by the smartphone to send data to the application server only. The application server sends an acknowledgment for the incoming data or messages. A transmission of data or messages accumulated by the application server is not possible with this web service. This service is only invoked if required, e.g., if an image or a text message needs to be transferred.

The update service for the representation of the positions of sensors and sensor carriers is used to dispatch data from the application server to the smartphone (see Section C). The smartphone uses this possibility to transfer not only its own position, but also any additional data accumulated so far. Another difference to the first web service consists of the fact that the smartphone calls this service periodically.

The AMFIS Connector software is therefore providing the suitable time slots according to the number of devices to be served. The smartphone is using the allocated time slot to change its own communication configuration accordingly. This is done to prevent a capacity overload, which might result in a data jam or even a breakdown of the communication infrastructure.

Within the update service, not only messages dedicated to be read by the mobile device are fetched but also other information is transferred; for example the position data of the mobile device. The position information of the mobile device is communicated with every existing proxy connection but at the latest with an established connection to the update service.

C. Application Server

The application server provides the web services as well as the connection management to AMFIS. It contains the runtime environment for these functions and is realized by an Apache Tomcat server. The application server offers different web services, which can be accessed by the smartphone. As soon as the smartphone has requested a service, a proxy is generated between the application server and the smartphone. This proxy runs only until the acknowledgement of the service and is terminated thereafter.

 ISBN: 978-1-61208-184-7

The AMFIS Connector management software runs on the application server. The software contains an encoder, which re-encodes the data received by the smartphone in suitable AMFIS messages and transmits these data to AMFIS. This process works also conversely using a suitable decoder to provide information to the mobile sensor. The management process is therefore able to decode AMFIS messages and to route the information to the receiving smartphone. As it is to be expected that the mobile device will not be able to sustain a connection under all circumstances, an agent object for each registered smartphone is created. Within this agent a stack of "messages to be transferred" exists, which contains all messages encoded in the smartphone application-readable format. Without any further transformation, necessary data can be transmitted faster in the event that a connection can be established. The messages remain in the list until reception of the data has been acknowledged. Even though, this increases the duration of the data exchange as well as the amount of the data to be exchanged, this is necessary to prevent a possible loss of messages, which might be critical. If the connection is disturbed or terminated during the transfer process, the data is not deleted but further provided and transferred with the next possible connection. In addition to these services, the agent object provides a certain supervision function. The agent is equipped with a timer, which is reset on each connection between the agent and the smartphone. If no connection to the smartphone can be established before timeout, the system can be informed about the connection state to the mobile device, which can be used for example, to notify the AMFIS system and the user about the lost link.

VI. CONCLUSIONS AND FUTURE WORK

The current state of the development is the first attempt to integrate a smartphone into the heterogeneous sensor network of the security framework AMFIS. The functions realized so far are only the most essential and most basic processes within this subsystem to make it usable.

Beside generating and providing reconnaissance results or reports in the form of images or text messages, the next logical step is to create and distribute video data. The essential problem to be solved is the higher amount of data if one or several sensors start to transmit video data simultaneously.

However, not only electro-optical sensors are conceivable. The array of sensors utilized in smartphones is constantly growing (compass, acceleration sensors etc.). Also, standardized interfaces, such as Bluetooth, can be used to link additional and more specialized sensors to the system.

Conceivable external sensors could be portable gas detectors or systems for detecting radiological and perhaps in the near future, also biological dangers. Moreover, the evaluation and processing of the data could already occur on-site. However, the ascertained results can be transmitted immediately to the ground station including the accompanying geo-information.

Furthermore, functions as the supervision of vital signs or an indoor localization of task forces, using a combination of runtime calculation of the signal, the values of the acceleration sensors and the build-in compass is conceivable.

In addition to these options, the use of the mobile device as an extra access point to distribute data without having to rely on an external connection is a benefit. A sort of multi-hopping network could be built up, which routes the information even under difficult communication conditions so that accumulated data can reach the ground station and other sensors carriers in the overall system.

ACKNOWLEDGMENT

The authors would like to thank Torsten Großkurth and Judy Lee-Wing for their contribution.

REFERENCES

[1] S. Leuchter, T. Partmann, L. Berger, E.J. Blum, and R. Schönbein, "Karlsruhe generic agile ground station," in: Beyerer J. (ed.) Future Security, 2nd Security Research Conference, 12th-14th September 2007, Karlsruhe, Germany, Fraunhofer Defense and Security Aliance, pp. 159-162. Universitätsverlag, Karlsruhe (2007).

[2] A. Bürkle, "Collaborating miniature drones for surveillance and reconnaissance", in Proceedings of SPIE 7480, 74800H, Berlin (2009); doi:10.1117/12.830408.

[3] F. Segor, A. Bürkle, M. Kollmann, and R. Schönbein, "Instantaneous Autonomous Aerial Reconnaissance for Civil Applications - A UAV based approach to support security and rescue forces," in: The 6th International Conference on Systems ICONS 2011, pp.72-76, St. Maarten, The Netherlands Antilles (2011).

[4] B. Essendorfer and W. Müller, "Interoperable sharing of data with the Coalition Shared Data (CSD) server," in: North Atlantic Treaty Organization (NATO)/Research and Technology Organziation (RTO): C31 in Crisis, Emergency and Consequence Management, page 12 - 24, Bucharest (2009).

[5] N. Heinze, M. Esswein, W. Krüger and G. Saur, "Image exploitation algorithms for reconnaissance and surveillance with UAV", in Proceedings of SPIE 7668, 76680U, Orlando (2010); doi:10.1117/12.852555.

[6] M-H. Tsou and J. Smith "Free and Open Source Software for GIS education," A White Paper, (2011), http://www.iapad.org/publications/ppgis/tsou_free-GIS-for-educators-whitepaper.pdf <retrieved: 11, 2011>.

 ISBN: 978-1-61208-184-7

Reliability Aspects of Uniformly Parameterised Cooperations

Peter Ochsenschläger and Roland Rieke
Fraunhofer Institute for Secure Information Technology, SIT
Darmstadt, Germany
Email: peter-ochsenschlaeger@t-online.de, roland.rieke@sit.fraunhofer.de

***Abstract*—In this paper, we examine reliability aspects of systems, which are characterised by the composition of a set of identical components. These components interact in a uniform manner, described by the schedules of the partners. Such kind of interaction is typical for scalable complex systems with cloud or grid structure. We call these systems "uniformly parameterised cooperations". We consider reliability of such systems in a possibilistic sense. This is formalised by always-eventually properties, a special class of liveness properties using a modified satisfaction relation, which expresses possibilities. As a main result, a finite state verification framework for uniformly parameterised reliability properties is given. The keys to this framework are structuring cooperations into phases and defining closed behaviours of systems. In order to verify reliability properties of such uniformly parameterised cooperations, we use finite state semi-algorithms that are independent of the concrete parameter setting.**

Keywords-reliability aspects of scalable complex systems; liveness properties; uniformly parameterised reliability properties; finite state verification; possibilistic reliability.

I. INTRODUCTION

The transition from systems composed of many isolated, small-scale elements to large-scale, distributed and massively interconnected systems is a key challenge of modern information and communications technologies. These systems need to be dependable, which means they need to remain secure, robust and efficient [1]. Examples for highly scalable systems comprise (i) grid computing architectures; and (ii) cloud computing platforms. In grid computing, large scale allocation issues relying on centralised controls present challenges that threaten to overwhelm existing centralised management approaches [1]. Cloud computing introduced the concept, to make software available as a service. This concept can only be successful, if certain obstacles such as reliability issues are solved [2]. In order to be able to model functional requirements of dependable systems best satisfying both fault-tolerance and security attributes, three distinct classes of (system specification) properties need to be considered, namely *safety*, *liveness*, and *information flow* [3]. Concrete reliability problems related to liveness properties range from replica selection to consistency of cloud storage (which allows multiple clients to access stored data concurrently in a consistent fashion) [4]. Most existing replica selection schemes rely on either central coordination (which has reliability, security, and scalability limitations) or distributed heuristics (which may lead to instability) [4]. Another important issue is, that clients of cloud services do not operate continuously, so clients should not depend on other clients for liveness of their operations [5].

In this paper, we consider systems that interact in a way that is typical for scalable complex systems. These systems, which we call *uniformly parameterised cooperations*, are characterised by (i) the composition of a set of identical components (copies of a two-sided cooperation); and (ii) the fact that these components interact in a uniform manner (described by the schedules of the partners). As an example of such uniformly parameterised systems of cooperations, e-commerce protocols can be considered. In these protocols, the two cooperation partners have to perform a certain kind of financial transactions. Such a protocol should work for several partners in the same manner, and the mechanism (schedule) to determine how one partner may be involved in several cooperations is the same for each partner. So, the cooperation is parameterised by the partners and the parameterisation should be uniform with respect to the partners.

Reliability is an important concept related to dependability, which ensures *continuity of correct service* [6]. In this paper, we consider reliability in a possibilistic sense, which means that correct services can be provided according to a certain pattern of behaviour again and again. These possibilities of providing correct services are expressed by a special class of liveness properties using a modified satisfaction relation. We call these properties *always-eventually properties*.

As a main result of the work presented, a finite state verification framework for *uniformly parameterised reliability properties* is given. The keys to this framework are structuring cooperations into phases and defining closed behaviours of systems. In this framework, *completion of phases strategies* and corresponding *success conditions* can be formalised [7], which produce finite state semi-algorithms that are independent of the concrete parameter setting. These algorithms are used to verify reliability properties of uniformly parameterised cooperations under certain regularity restrictions.

The paper is structured as follows. Section II gives an overview of the related work. In Section III, uniform parameterisations of two-sided cooperations in terms of formal

 ISBN: 978-1-61208-184-7

language theory is formalised. Section IV introduces the concept of uniformly parameterised reliability properties. The concept of structuring cooperations into phases given in Section V enables completion of phases strategies, which are described in Section VI. Consistent with this, corresponding success conditions can be formalised [2], which produce finite state semi-algorithms to verify reliability properties of uniformly parameterised cooperations. Finally, the paper ends with conclusions and an outlook in Section VII.

II. Related Work

System properties: A formal definition of safety and liveness properties is proposed in [8]. In [9], we defined a satisfaction relation, called *approximate satisfaction*, which expresses a possibilistic view on liveness and is equivalent to the satisfaction relation in [8] for safety properties. In this paper, we extended this concept (cf. Section IV) and defined *uniformly parameterised reliability properties*, which fit to the parameterised structure of the systems, which we consider here. Besides these safety and liveness properties so called "hyperproperties" [10] are of interest because they give formalisations for non-interference and non-inference.

Verification approaches for parameterised systems: An extension to the *Mur*φ verifier to verify systems with replicated identical components through a new data type called RepetitiveID is presented in [11]. A typical application area of this tool are cache coherence protocols. The aim of [12] is an abstraction method through symmetry, which works also when using variables holding references to other processes. This is not possible in Murφ. In [13], a methodology for constructing abstractions and refining them by analysing counter-examples is presented. The method combines abstraction, model-checking and deductive verification. However, this approach does not consider liveness properties. In [14], a technique for automatic verification of parameterised systems based on process algebra *CCS* [15] and the logic modal *mu-calculus* [16] is presented. This technique views processes as property transformers and is based on computing the limit of a sequence of mu-calculus formula generated by these transformers. The above-mentioned approaches demonstrate, that finite state methods combined with deductive methods can be applied to analyse parameterised systems. The approaches differ in varying amounts of user intervention and their range of application. A survey of approaches to combine model checking and theorem proving methods is given in [17].

Iterated shuffle products: In [18], it is shown that our definition of uniformly parameterised cooperations is strongly related to iterated shuffle products [19], if the cooperations are "structured into phases". The main concept for such a condition are shuffle automata [20] (multicounter automata [21]) whose computations, if they are deterministic, unambiguously describe how a cooperation partner is involved in several phases.

In [22], we have shown in particular that for self-similar parameterised systems $\mathscr{L}_{IK}$ the parameterised problem of verifying a *uniformly parameterised safety property* can be reduced to finite many fixed finite state problems.

Complementary to this, in the present paper, we define a uniformly parameterised reliability property based on this concept. The main result is a finite state verification framework for such *uniformly parameterised reliability properties*.

III. Parameterised Cooperations

The behaviour L of a discrete system can be formally described by the set of its possible sequences of actions. Therefore $L \subset \Sigma^*$ holds where Σ is the set of all actions of the system, and Σ^* (free monoid over Σ) is the set of all finite sequences of elements of Σ (words), including the empty sequence denoted by ε. $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$. Subsets of Σ^* are called formal languages [23]. Words can be composed: if u and v are words, then uv is also a word. This operation is called the *concatenation*; especially $\varepsilon u = u\varepsilon = u$. Concatenation of formal languages $U, V \subset \Sigma^*$ are defined by $UV := \{uv \in \Sigma^* | u \in U \text{ and } v \in V\}$. A word u is called a *prefix* of a word v if there is a word x such that $v = ux$. The set of all prefixes of a word u is denoted by $\text{pre}(u)$; $\varepsilon \in \text{pre}(u)$ holds for every word u. The set of possible continuations of a word $u \in L$ is formalised by the *left quotient* $u^{-1}(L) := \{x \in \Sigma^* | ux \in L\}$.

Infinite words over Σ are called ω-words [24]. The set of all infinite words over Σ is denoted Σ^ω. An ω-language L over Σ is a subset of Σ^ω. For $u \in \Sigma^*$ and $v \in \Sigma^\omega$ the *left concatenation* $uv \in \Sigma^\omega$ is defined. It is also defined for $U \subset \Sigma^*$ and $V \subset \Sigma^\omega$ by $UV := \{uv \in \Sigma^\omega | u \in U \text{ and } v \in V\}$.

For an ω-word w the prefix set is given by the formal language $\text{pre}(w)$, which contains every finite prefix of w. The prefix set of an ω-language $L \subset \Sigma^\omega$ is accordingly given by $\text{pre}(L) = \{u \in \Sigma^* |$ it exist $v \in \Sigma^\omega$ with $uv \in L\}$. For $M \subset \Sigma^*$ the *ω-power* $M^\omega \subset \Sigma^\omega$ is the set of all "infinite concatenations" of arbitrary elements of M. More formal definitions of theses ω-notions are given in the appendix.

Formal languages, which describe system behaviour, have the characteristic that $\text{pre}(u) \subset L$ holds for every word $u \in L$. Such languages are called *prefix closed*. System behaviour is thus described by prefix closed formal languages.

Different formal models of the same system are partially ordered with respect to different levels of abstraction. Formally, abstractions are described by so called alphabetic language homomorphisms. These are mappings $h^* : \Sigma^* \longrightarrow \Sigma'^*$ with $h^*(xy) = h^*(x)h^*(y)$, $h^*(\varepsilon) = \varepsilon$ and $h^*(\Sigma) \subset \Sigma' \cup \{\varepsilon\}$. So, they are uniquely defined by corresponding mappings $h : \Sigma \longrightarrow \Sigma' \cup \{\varepsilon\}$. In the following, we denote both the mapping h and the homomorphism h^* by h. Inverse homomorphism are denoted by h^{-1}. Let L be a language over the alphabet Σ'. Then $h^{-1}(L)$ is the set of words $w \in \Sigma^*$ such that $h(w) \in L$. In this paper, we consider a lot of alphabetic language homomorphisms. So, for simplicity, we

 ISBN: 978-1-61208-184-7

tacitly assume that a mapping between free monoids is an alphabetic language homomorphism if nothing contrary is stated.

To describe a two-sided cooperation, let $\Sigma = \Phi \cup \Gamma$ where Φ is the set of actions of cooperation partner F and Γ is the set of actions of cooperation partner G and $\Phi \cap \Gamma = \emptyset$. Now a prefix closed language $L \subset (\Phi \cup \Gamma)^*$ formally defines a two-sided cooperation.

Example 1. *Let* $\Phi = \{\mathrm{f_s}, \mathrm{f_r}\}$ *and* $\Gamma = \{\mathrm{g_r}, \mathrm{g_i}, \mathrm{g_s}\}$ *and hence* $\Sigma = \{\mathrm{f_s}, \mathrm{f_r}, \mathrm{g_r}, \mathrm{g_i}, \mathrm{g_s}\}$. *An example for a cooperation* $L \subset \Sigma^*$ *is now given by the automaton in Figure 1. It describes a simple handshake between F (client) and G (server), where a client may perform the actions* $\mathrm{f_s}$ *(send a request),* $\mathrm{f_r}$ *(receive a result) and a server may perform the corresponding actions* $\mathrm{g_r}$ *(receive a request),* $\mathrm{g_i}$ *(internal action to compute the result) and* $\mathrm{g_s}$ *(send the result).*

In the following, we will denote initial states by a short incoming arrow and final states by double circles. In this automaton, all states are final states, since L is prefix closed.

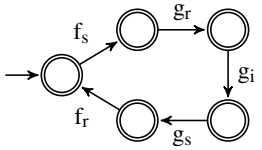


Figure 1. Automaton for 1-1-cooperation L

For parameter sets I, K and $(i,k) \in I \times K$ let Σ_{ik} denote pairwise disjoint copies of Σ. The elements of Σ_{ik} are denoted by a_{ik} and $\Sigma_{IK} := \bigcup_{(i,k)\in I\times K} \Sigma_{ik}$. The index ik describes the bijection $a \leftrightarrow a_{ik}$ for $a \in \Sigma$ and $a_{ik} \in \Sigma_{ik}$. Now $\mathscr{L}_{IK} \subset \Sigma_{IK}^*$ (prefix-closed) describes a *parameterised system*. To avoid pathological cases, we generally assume parameter and index sets to be non empty.

For a cooperation between one partner of type F with two partners of type G in Example 1 let

$$\Phi_{\{1\}\{1,2\}} = \{\mathrm{f_{s11}}, \mathrm{f_{r11}}, \mathrm{f_{s12}}, \mathrm{f_{r12}}\},$$
$$\Gamma_{\{1\}\{1,2\}} = \{\mathrm{g_{r11}}, \mathrm{g_{i11}}, \mathrm{g_{s11}}, \mathrm{g_{r12}}, \mathrm{g_{i12}}, \mathrm{g_{s12}}\} \text{ and}$$
$$\Sigma_{\{1\}\{1,2\}} = \Phi_{\{1\}\{1,2\}} \cup \Gamma_{\{1\}\{1,2\}}.$$

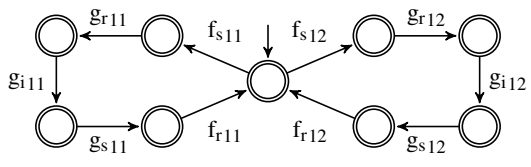


Figure 2. Automaton for 1-2-cooperation $\mathscr{L}_{\{1\}\{1,2\}}$

A 1-2-cooperation, where each pair of partners cooperates restricted by L and each partner has to finish the handshake it just is involved in before entering a new one, is now given (by reachability analysis) by the automaton in Figure 2 for $\mathscr{L}_{\{1\}\{1,2\}}$. It shows that one after another client 1 runs a handshake either with server 1 or with server 2. Figure 3 in contrast depicts an automaton for a 2-1-cooperation $\mathscr{L}_{\{1,2\}\{1\}}$ with the same overall number of partners involved but two of type F and one partner of type G. Figure 3 is more complex than Figure 2 because client 1 and client 2 may start a handshake independently of each other, but server 1 handles these handshakes one after another. A 5-3-cooperation with the same simple behaviour of partners already requires 194.677 states and 1.031.835 state transitions (computed by the SH verification tool [25]).

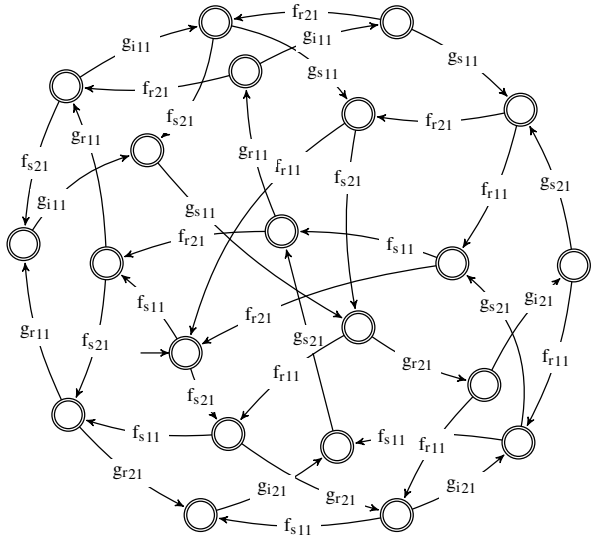


Figure 3. Automaton for the 2-1-cooperation $\mathscr{L}_{\{1,2\}\{1\}}$

For $(i,k) \in I \times K$, let $\pi_{ik}^{IK} : \Sigma_{IK}^* \to \Sigma^*$ with

$$\pi_{ik}^{IK}(a_{rs}) = \begin{cases} a \mid & a_{rs} \in \Sigma_{ik} \\ \varepsilon \mid & a_{rs} \in \Sigma_{IK} \setminus \Sigma_{ik} \end{cases}.$$

For *uniformly parameterised systems* $\mathscr{L}_{IK}$ we generally want to have

$$\mathscr{L}_{IK} \subset \bigcap_{(i,k)\in I\times K} ((\pi_{ik}^{IK})^{-1}(L))$$

because from an abstraction point of view, where only the actions of a specific Σ_{ik} are considered, the complex system $\mathscr{L}_{IK}$ is restricted by L.

In addition to this inclusion, $\mathscr{L}_{IK}$ is defined by *local schedules* that determine how each "version of a partner" can participate in different cooperations. More precisely, let $SF \subset \Phi^*$, $SG \subset \Gamma^*$ be prefix closed. For $(i,k) \in I \times$

 ISBN: 978-1-61208-184-7

K, let $\varphi_i^{IK} : \Sigma_{IK}^* \to \Phi^*$ and $\gamma_k^{IK} : \Sigma_{IK}^* \to \Gamma^*$ with

$$\varphi_i^{IK}(a_{rs}) = \begin{cases} a \mid & a_{rs} \in \Phi_{\{i\}K} \\ \varepsilon \mid & a_{rs} \in \Sigma_{IK} \setminus \Phi_{\{i\}K} \end{cases} \text{ and}$$

$$\gamma_k^{IK}(a_{rs}) = \begin{cases} a \mid & a_{rs} \in \Gamma_{I\{k\}} \\ \varepsilon \mid & a_{rs} \in \Sigma_{IK} \setminus \Gamma_{I\{k\}} \end{cases},$$

where Φ_{IK} and Γ_{IK} are defined correspondingly to Σ_{IK}.

Definition 1 (uniformly parameterised cooperation)**.** *Let I, K be finite parameter sets, then*

$$\mathscr{L}_{IK} := \bigcap_{(i,k)\in I\times K} (\pi_{ik}^{IK})^{-1}(L) \cap \bigcap_{i\in I} (\varphi_i^{IK})^{-1}(SF) \cap \bigcap_{k\in K} (\gamma_k^{IK})^{-1}(SG)$$

denotes a uniformly parameterised cooperation*.*

By this definition,

$$\mathscr{L}_{\{1\}\{1\}} = (\pi_{11}^{\{1\}\{1\}})^{-1}(L) \cap (\varphi_1^{\{1\}\{1\}})^{-1}(SF) \cap (\gamma_1^{\{1\}\{1\}})^{-1}(SG).$$

Because we want $\mathscr{L}_{\{1\}\{1\}}$ being isomorphic to L by the isomorphism $\pi_{11}^{\{1\}\{1\}} : \Sigma_{\{1\}\{1\}}^* \to \Sigma^*$, we additionally need

$$(\pi_{11}^{\{1\}\{1\}})^{-1}(L) \subset (\varphi_1^{\{1\}\{1\}})^{-1}(SF) \text{ and}$$
$$(\pi_{11}^{\{1\}\{1\}})^{-1}(L) \subset (\gamma_1^{\{1\}\{1\}})^{-1}(SG).$$

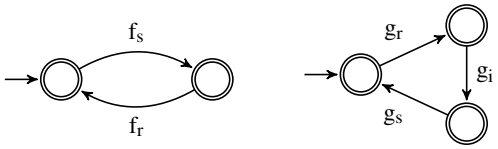
This is equivalent to $\pi_\Phi(L) \subset SF$ and $\pi_\Gamma(L) \subset SG$, where $\pi_\Phi : \Sigma^* \to \Phi^*$ and $\pi_\Gamma : \Sigma^* \to \Gamma^*$ are defined by

$$\pi_\Phi(a) = \begin{cases} a \mid & a \in \Phi \\ \varepsilon \mid & a \in \Gamma \end{cases} \text{ and } \pi_\Gamma(a) = \begin{cases} a \mid & a \in \Gamma \\ \varepsilon \mid & a \in \Phi \end{cases}.$$

So, we complete Def. 1 by the additional conditions

$$\pi_\Phi(L) \subset SF \text{ and } \pi_\Gamma(L) \subset SG.$$

Schedules SF and SG that fit to the cooperations given in Example 1 are depicted in Figs. 4(a) and 4(b). Here, we have $\pi_\Phi(L) = SF$ and $\pi_\Gamma(L) = SG$.



(a) Schedule SF (b) Schedule SG

Figure 4. Automata $\mathbb{SF}$ and $\mathbb{SG}$ for the schedules SF and SG

The system $\mathscr{L}_{IK}$ of cooperations is a typical example of a *complex system*. It consists of several identical components (copies of the two-sided cooperation L), which interact in a uniform manner (described by the schedules SF and SG and by the homomorphisms φ_i^{IK} and γ_k^{IK}).

Remark 1. *It is easy to see that $\mathscr{L}_{IK}$ is isomorphic to $\mathscr{L}_{I'K'}$ if I is isomorphic to I' and K is isomorphic to K'. More precisely, let $\iota_{I'}^{I} : I \to I'$ and $\iota_{K'}^{K} : K \to K'$ be bijections and let $\iota_{I'K'}^{IK} : \Sigma_{IK}^* \to \Sigma_{I'K'}^*$ be defined by*

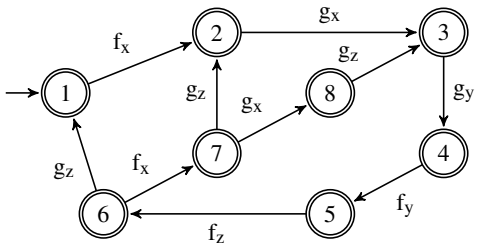
$$\iota_{I'K'}^{IK}(a_{ik}) := a_{\iota_{I'}^{I}(i)\iota_{K'}^{K}(k)} \text{ for } a_{ik} \in \Sigma_{IK}.$$

Hence, $\iota_{I'K'}^{IK}$ is a isomorphism and $\iota_{I'K'}^{IK}(\mathscr{L}_{IK}) = \mathscr{L}_{I'K'}$. The set of all these isomorphisms $\iota_{I'K'}^{IK}$ defined by corresponding bijections $\iota_{I'}^{I}$ and $\iota_{K'}^{K}$ is denoted by $\mathscr{I}_{I'K'}^{IK}$.

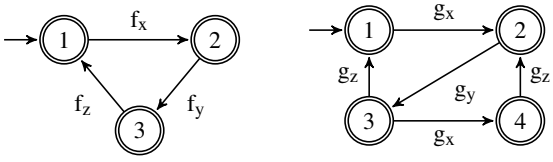
To illustrate the concepts of this paper, we consider the following example.

Example 2. *We consider a system of servers, each of them managing a resource, and clients, which want to use these resources. We assume that as a means to enforce a given* privacy policy *a server has to manage its resource in such a way that no client may access this resource while it is in use by another client (*privacy requirement*). This may be required to ensure* anonymity *in such a way that clients and their actions on a resource cannot be linked by an observer.*

We formalise this system at an abstract level, where a client may perform the actions f_x (send a request), f_y (receive a permission) and f_z (send a free-message), and a server may perform the corresponding actions g_x (receive a request), g_y (send a permission) and g_z (receive a free-message). The possible sequences of actions of a client resp. of a server are given by the automaton $\mathbb{SF}$ resp. $\mathbb{SG}$. The automaton $\mathbb{L}$ describes the 1-1-cooperation of one client and one server (see Figure 5). These automata define the client-server system $\mathscr{L}_{IK}$.



(a) 1-1-cooperation L



(b) Schedule SF (c) Schedule SG

Figure 5. Automata $\mathbb{L}$, $\mathbb{SF}$ and $\mathbb{SG}$ for Example 2

IV. A CLASS OF LIVENESS PROPERTIES

Usually, behaviour properties of systems are divided into two classes: *safety* and *liveness* properties [8]. Intuitively

a safety property stipulates that "something bad does not happen" and a liveness property stipulates that "something good eventually happens". In [8], both classes, as well as system behaviour, are formalised in terms of ω-languages, because especially for liveness properties infinite sequences of actions have to be considered.

Definition 2 (linear satisfaction)**.** *According to [8], a property E of a system is a subset of Σ^ω. If $S \subset \Sigma^\omega$ represents the behaviour of a system, then S* linearly satisfies *E iff $S \subset E$.*

In [8], it is furthermore shown that each property E is the intersection of a safety and a liveness property.

Safety properties $E_s \subset \Sigma^\omega$ are of the form $E_s = \Sigma^\omega \setminus F\Sigma^\omega$ with $F \subset \Sigma^*$, where F is the set of "bad things".

Liveness properties $E_l \subset \Sigma^\omega$ are characterised by $\mathrm{pre}(E_l) = \Sigma^*$. A typical example of a liveness property is

$$E_l = (\Sigma^* M)^\omega \text{ with } \emptyset \neq M \subset \Sigma^+. \tag{1}$$

This E_l formalises that "always eventually a finite action sequence $m \in M$ happens".

We describe system behaviour by prefix closed languages $B \subset \Sigma^*$. So, in order to apply the framework of [8], we have to transform B into an ω-language. This can be done by the *limit* $\lim(B)$ [24]. For prefix closed languages $B \subset \Sigma^*$, their limit is defined by

$$\lim(B) := \{w \in \Sigma^\omega | \mathrm{pre}(w) \subset B\}.$$

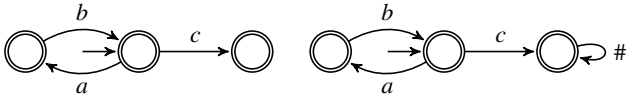
If B contains maximal words u (deadlocks), then these u are not captured by $\lim(B)$. Formally the set $\max(B)$ of all maximal words of B is defined by

$$\max(B) := \{u \in B | \text{ if } v \in B \text{ with } u \in \mathrm{pre}(v), \text{ then } v = u\}.$$

Now, using a dummy action #, B can be unambiguously described by

$$\hat{B} := B \cup \max(B)\#^* \subset \hat{\Sigma}^*,$$

where $\# \notin \Sigma$ and $\hat{\Sigma} := \Sigma \cup \{\#\}$. By this definition, in $\hat{B}$ the maximal words of B are continued by arbitrary many #'s. So, $\hat{B}$ does not contain maximal words.



(a) Automaton for B (b) Automaton for $\hat{B}$

Figure 6. Automata for B and $\hat{B}$

Let for example B be given by the automaton in Figure 6(a), then $\hat{B}$ is given by the automaton in Figure 6(b).

By this construction, we now can assume that system behaviour is formalised by prefix closed languages $\hat{B} \subset \Sigma^*\#^* \subset \hat{\Sigma}^*$ without maximal words, and the corresponding infinite system behaviour $S \subset \Sigma^\omega$ is given by $S := \lim(\hat{B})$.

For such an S and safety properties $E = \hat{\Sigma}^\omega \setminus F\hat{\Sigma}^\omega$ with $F \subset \hat{\Sigma}^*$ it holds

$$S \subset E \text{ iff } S \cap F\hat{\Sigma}^\omega = \emptyset \text{ iff } \mathrm{pre}(S) \cap F = \emptyset \text{ iff } \hat{B} \cap F = \emptyset.$$

If $F \subset \Sigma^*$, then $\hat{B} \cap F = \emptyset$ iff $B \cap F = \emptyset$. Therefore,

$$S \subset E \text{ iff } B \cap F = \emptyset \text{ for } F \subset \Sigma^*. \tag{2}$$

So, by (2) our approach in [22] is equivalent to the ω-notation of safety properties described by $F \subset \Sigma^*$.

Linear satisfaction (cf. Def. 2) is too strong for systems in our focus with respect to liveness properties, because $S = \lim(\hat{B})$ can contain "unfair" infinite behaviours, which are not elements of E.

Let for example $I \supset \{1,2\}$ and $K \supset \{1\}$, then $\lim(\widehat{\mathscr{L}_{IK}}) \cap \Sigma^\omega_{\{1\}\{1\}} \neq \emptyset$, which means that infinite action sequences exist, where only the partners with index 1 cooperate. So, if a property specification involves actions of a partner with index 2, as for instance $E = \Sigma^*_{IK}\Sigma_{\{2\}\{1\}}\Sigma^\omega_{IK}$, then this property is not linearly satisfied because $\lim(\widehat{\mathscr{L}_{IK}}) \not\subset E$.

Instead of neglecting such unfair infinite behaviours, we use a weaker satisfaction relation, called *approximate satisfaction*, which implicitly expresses some kind of fairness.

Definition 3 (approximate satisfaction)**.** *A system $S \subset \hat{\Sigma}^\omega$* approximately satisfies *a property $E \subset \hat{\Sigma}^\omega$ iff each finite behaviour (finite prefix of an element of S) can be continued to an infinite behaviour, which belongs to E. More formally, $pre(S) \subset pre(S \cap E)$.*

In [9], it is shown, that for safety properties linear satisfaction and approximate satisfaction are equivalent.

With respect to approximate satisfaction, liveness properties stipulate that "something good" eventually is possible.

Many practical liveness properties are of the form (1). Let us consider a prefix closed language $B \subset \Sigma^*$ and a formal language $\emptyset \neq M \subset \Sigma^+$. By definition 3 $\lim(\hat{B})$ approximately satisfies $(\hat{\Sigma}^* M)^\omega$ iff each $u \in B$ is prefix of some $v \in B$ with

$$v^{-1}(B) \cap M \neq \emptyset. \tag{3}$$

If B and M are regular sets, then (3) can be checked by usual automata algorithms [23] without referring to $\lim(\hat{B}) \cap (\hat{\Sigma}^* M)^\omega$.

Let us now consider the prefix closed language $L \subset \Sigma^*$ of example 2 and the "phase" $P \subset \Sigma^+$ given by the automaton $\mathbb{P}$ in Figure 7.

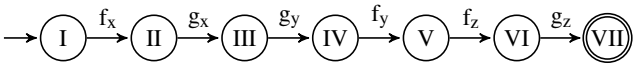


Figure 7. Automaton $\mathbb{P}$

$\lim(\hat{L})$ approximately satisfies the liveness property $(\hat{\Sigma}^* P)^\omega \subset \hat{\Sigma}^*$, because the automaton $\mathbb{L}$ in Figure 5(a) is strongly connected and $P \subset L$. (4)

(4) states that in the 1-1-cooperation $\lim(\hat{L})$ always eventually a "complete run through the phase P" is possible. This is a typical reliability property.

Properties of the form $(\hat{\Sigma}^* M)^\omega$ with $\emptyset \neq M \subset \Sigma^+$ we call *always-eventually properties*.

Let now $\emptyset \neq \mathring{M} \subset \Sigma^+_{\mathring{I}\mathring{K}}$ with fixed finite index sets $\mathring{I}$ and $\mathring{K}$. Then

$$(\hat{\Sigma}^*_{IK} \mathring{M})^\omega$$

is an always-eventually property for each finite index sets $I \supset \mathring{I}$ and $K \supset \mathring{K}$. Using bijections on $\mathring{I}$ and $\mathring{K}$ this can easily be generalised to each finite index sets I and K with $|I| \geq |\mathring{I}|$ and $|K| \geq |\mathring{K}|$, where $|I|$ denotes the cardinality of the set I. More precisely, let $\mathscr{I}^{\mathring{I}\mathring{K}}_{I'K'}$ be the set of all isomorphisms $\iota^{\mathring{I}\mathring{K}}_{I'K'} : \Sigma^*_{\mathring{I}\mathring{K}} \to \Sigma^*_{I'K'}$ generated by bijections $\iota^{\mathring{I}}_{I'} : \mathring{I} \to I'$ and $\iota^{\mathring{K}}_{K'} : \mathring{K} \to K'$ in such a way that

$$\iota^{\mathring{I}\mathring{K}}_{I'K'}(a_{ik}) := a_{\iota^{\mathring{I}}_{I'}(i)\iota^{\mathring{K}}_{K'}(k)}$$

for $a_{ik} \in \Sigma_{\mathring{I}\mathring{K}}$. Then

$$(\hat{\Sigma}^*_{IK} \iota^{\mathring{I}\mathring{K}}_{I'K'}(\mathring{M}))^\omega$$

is an always-eventually property for each $I \supset I'$, $K \supset K'$ and $\iota^{\mathring{I}\mathring{K}}_{I'K'} \in \mathscr{I}^{\mathring{I}\mathring{K}}_{I'K'}$. For finite index sets $\mathring{I}$, I, $\mathring{K}$ and K let

$$\mathscr{I}[(\mathring{I},\mathring{K}),(I,K)] := \bigcup_{I' \subset I, K' \subset K} \mathscr{I}^{\mathring{I}\mathring{K}}_{I'K'}.$$

Note that $\mathscr{I}[(\mathring{I},\mathring{K}),(I,K)] = \emptyset$ if $|\mathring{I}| > |I|$ or $|\mathring{K}| > |K|$.

Definition 4 (uniformly parameterised reliability property)**.** *Let $\mathring{I}$, I, $\mathring{K}$ and K be finite index sets with $|\mathring{I}| \leq |I|$ and $|\mathring{K}| \leq |K|$. If $\emptyset \neq \mathring{M} \subset \Sigma^+_{\mathring{I}\mathring{K}}$, then the family*

$$\mathscr{A}^{\mathring{M}}_{IK} := [(\hat{\Sigma}^*_{IK} \iota^{\mathring{I}\mathring{K}}_{I'K'}(\mathring{M}))^\omega]_{\iota^{\mathring{I}\mathring{K}}_{I'K'} \in \mathscr{I}[(\mathring{I},\mathring{K}),(I,K)]}.$$

is a strong uniformly parameterised always-eventually property *(uniformly parameterised reliability property).*

We say that $lim(\widehat{\mathscr{L}_{IK}})$ approximately satisfies *such a family $\mathscr{A}^{\mathring{M}}_{IK}$ iff $lim(\widehat{\mathscr{L}_{IK}})$ approximately satisfies each of the properties $(\hat{\Sigma}^*_{IK} \iota^{\mathring{I}\mathring{K}}_{I'K'}(\mathring{M}))^\omega$ for $\iota^{\mathring{I}\mathring{K}}_{I'K'} \in \mathscr{I}[(\mathring{I},\mathring{K}),(I,K)]$.*

Remark 2. *We use the adjective* strong*, because in [7] uniform parameterisations of general properties are defined, which, in case of always-eventually properties, are weaker than definition 4.*

Let us return to example 2 and let

$$\mathring{P} := (\pi^{\{1\}\{1\}}_{11})^{-1} P \subset \Sigma^+_{\{1\}\{1\}} \text{ and}$$
$$\mathring{E} := (\widehat{\Sigma_{\{1\}\{1\}}}^* \mathring{P})^\omega \subset \widehat{\Sigma_{\{1\}\{1\}}}^\omega. \quad (5)$$

Because $\pi^{\{1\}\{1\}}_{11} : \Sigma^*_{\{1\}\{1\}} \to \Sigma^*$ is an isomorphism, by (4) $\lim(\widehat{\mathscr{L}_{\{1\}\{1\}}})$ approximately satisfies $\mathring{E}$.

Now by definition 4 $\lim(\widehat{\mathscr{L}_{IK}})$ approximately satisfies $\mathscr{A}^{\mathring{P}}_{IK}$ iff in $\lim(\widehat{\mathscr{L}_{IK}})$ for each pair of clients and servers always eventually a complete run through a phase P is possible.

V. Cooperations based on phases

The schedule *SG* of example 2 shows that a server may cooperate with two clients partly in an interleaving manner. To formally capture such behaviour, cooperations are structured into phases [18]. This formalism is based on iterated shuffle products and leads to sufficient conditions for liveness properties (cf. Section VI).

Shuffling two words means arbitrarily inserting one word into the other word, like shuffling two decks of cards. In [21], this is formalised as follows:

A word $w \in \Sigma^*$ is called a *shuffle* of words $w_1, \ldots, w_m \in \Sigma^*$ if the positions of w can be coloured using m colors so that the positions with color $i \in \{1, \ldots, m\}$, when read from left to right, form the word w_i. *Shuffle of a set* $P \subset \Sigma^*$, is $\{w :$ w is a shuffle of some $w_1, \ldots, w_m \in P$, for some $m \in \mathbb{N}\}$.

However, we now provide an alternative formalisation, which is more adequate to the considerations in this paper.

Definition 5 (iterated shuffle product)**.** *Let $t \in \mathbb{N}$, and for each t let Σ_t be a copy of Σ. Let all Σ_t be pairwise disjoint. The index t describes the bijection $a \leftrightarrow a_t$ for $a \in \Sigma$ and $a_t \in \Sigma_t$ (which is equivalent to a colouring with color t in the formalism of [21]). Let*

$$\Sigma_{\mathbb{N}} := \bigcup_{t \in \mathbb{N}} \Sigma_t, \text{ and for each } t \in \mathbb{N}$$

let the homomorphisms $\tau^{\mathbb{N}}_t$ and $\Theta^{\mathbb{N}}$ be defined by

$$\tau^{\mathbb{N}}_t : \Sigma^*_{\mathbb{N}} \to \Sigma^* \text{ with } \tau^{\mathbb{N}}_t(a_s) = \begin{cases} a & | \quad a_s \in \Sigma_t \\ \varepsilon & | \quad a_s \in \Sigma_{\mathbb{N}} \setminus \Sigma_t \end{cases} \text{ and}$$

$$\Theta^{\mathbb{N}} : \Sigma^*_{\mathbb{N}} \to \Sigma^* \text{ with } \Theta^{\mathbb{N}}(a_t) := a \text{ for } a_t \in \Sigma_t \text{ and } t \in \mathbb{N}.$$

The iterated shuffle product $P^{\sqcup\!\sqcup}$ *of P is now defined by*

$$P^{\sqcup\!\sqcup} := \Theta^{\mathbb{N}}[\bigcap_{t \in \mathbb{N}} (\tau^{\mathbb{N}}_t)^{-1}(P \cup \{\varepsilon\})] \text{ for } P \subset \Sigma^*.$$

It is easy to see that this is equivalent to the definition from [21] above. Let for example $P = \{ab\}$. Now, according to [21], the word $w = aabb$ is a shuffle of two words w_1, $w_2 \in P$ because two colors, namely 1 and 2, can be used to colour the word $aabb$ so that $w_1 = w_2 = ab \in P$. According to definition 5, $aabb \in P^{\sqcup\!\sqcup}$ because $aabb = \Theta^{\mathbb{N}}(a_1a_2b_2b_1)$ and $\tau^{\mathbb{N}}_1(a_1a_2b_2b_1) = \tau^{\mathbb{N}}_2(a_1a_2b_2b_1) = ab \in P$ and $\tau^{\mathbb{N}}_t(a_1a_2b_2b_1) = \varepsilon$ for $t \in \mathbb{N} \setminus \{1,2\}$.

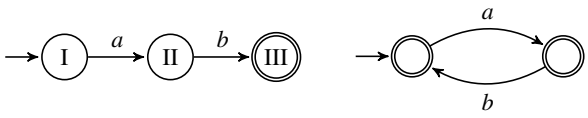
Following the ideas in [18], we structure cooperations into phases.

Definition 6 (based on a phase)**.** *A prefix closed language $B \subset \Sigma^*$ is based on a phase $P \subset \Sigma^*$, iff $B = \mathrm{pre}(P^{\sqcup\!\sqcup} \cap B)$.*

If B is based on P, then $B \subset \mathrm{pre}(P^{\sqcup\!\sqcup}) = (\mathrm{pre}(P))^{\sqcup\!\sqcup}$ and $B = \mathrm{pre}(P)^{\sqcup\!\sqcup} \cap B$.

Let for example $P = \{ab\}$ be given by the Automaton $\mathbb{P}$ in Figure 8(a) and B be given by the automaton $\mathbb{B}$ in Figure 8(b). Then $P^{\sqcup\!\sqcup} \cap B = \{ab\}^*$. This implies that B is based on P.

 ISBN: 978-1-61208-184-7



(a) Automaton $\mathbb{P}$ for P= {ab} (b) Automaton $\mathbb{B}$ for B

Figure 8. Automata $\mathbb{P}$ and $\mathbb{B}$

Generally, each B is based on infinitely many phases. If B is based on P, then B is based on P' for each $P' \supset P$. Each $B \subset \Sigma^*$ is based on Σ because $\Sigma^{\sqcup\!\sqcup} = \Sigma^*$. Figure 9 shows how we use phases to structure cooperations. The appropriate phases for our purposes as well as *closed behaviours* (words, in which all phases are completed) will be discussed in Section VI.

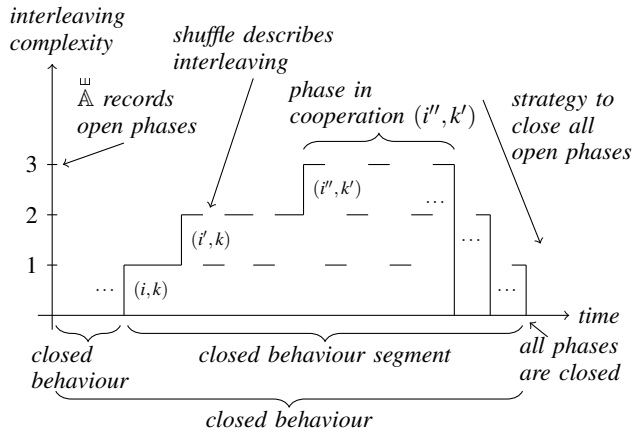


Figure 9. Phases and closed behaviours

We will now provide an automaton representation $\tilde{\mathbb{A}}$ for $P^{\sqcup\!\sqcup}$, which will illustrate "how a language B is based on a phase P". Let $P \subset \Sigma^*$ and $\mathbb{A} = (\Sigma, Q, \Delta, q_0, F)$ with $\Delta \subset Q \times \Sigma \times Q$, $q_0 \in Q$ and $F \subset Q$ be an (not necessarily finite) automaton that accepts P. To exclude pathological cases we assume $\varepsilon \notin P \neq \emptyset$. A consequence of this is in particular that $q_0 \notin F$. Let $\mathbb{N}_0^Q$ denote the set of all functions from Q in $\mathbb{N}_0$. For the construction of $\tilde{\mathbb{A}}$ the set $\mathbb{N}_0^Q$ plays a central role. In $\mathbb{N}_0^Q$ we distinguish the following functions:

$$0 \in \mathbb{N}_0^Q \text{ with } 0(x) = 0 \text{ for each } x \in Q,$$

and for $q \in Q$ the function

$$1_q \in \mathbb{N}_0^Q \text{ with } 1_q(x) = \begin{cases} 1 \mid & x = q \\ 0 \mid & x \in Q \setminus \{q\} \end{cases}.$$

As usual for numerical functions, a partial order as well as addition and partial subtraction are defined.

For $f, g \in \mathbb{N}_0^Q$ let $f \geqslant g$ iff $f(x) \geqslant g(x)$ for each $x \in Q$, $f + g \in \mathbb{N}_0^Q$ with $(f+g)(x) := f(x) + g(x)$ for each $x \in Q$, and for $f \geqslant g$, $f - g \in \mathbb{N}_0^Q$ with $(f-g)(x) := f(x) - g(x)$ for each $x \in Q$.

The key idea of $\tilde{\mathbb{A}}$ is, to record in the functions of $\mathbb{N}_0^Q$ how many open phases are in each state $q \in Q$ respectively. Its state transition relation $\tilde{\Delta}$ is composed of four subsets whose elements describe (a) the entry into a new phase, (b) the transition within an open phase, (c) the completion of an open phase, (d) the entry into a new phase with simultaneous completion of this phase. With these definitions we now define the *shuffle automaton* $\tilde{\mathbb{A}}$.

Definition 7 (shuffle automaton)**.**
The shuffle automaton $\tilde{\mathbb{A}} = (\Sigma, \mathbb{N}_0^Q, \tilde{\Delta}, 0, \{0\})$ *w.r.t.* $\mathbb{A}$ *is an automaton with infinite state set* $\mathbb{N}_0^Q$*, the initial state* 0*, which is the only final state and*

$$\begin{aligned}\tilde{\Delta} := &\{(f,a,f+1_p) \in \mathbb{N}_0^Q \times \Sigma \times \mathbb{N}_0^Q \mid \\ &\quad (q_0,a,p) \in \Delta \text{ and it exists } (p,x,y) \in \Delta\} \cup \\ &\{(f,a,f+1_p-1_q) \in \mathbb{N}_0^Q \times \Sigma \times \mathbb{N}_0^Q \mid \\ &\quad f \geqslant 1_q, (q,a,p) \in \Delta \text{ and it exists } (p,x,y) \in \Delta\} \cup \\ &\{(f,a,f-1_q) \in \mathbb{N}_0^Q \times \Sigma \times \mathbb{N}_0^Q \mid \\ &\quad f \geqslant 1_q, (q,a,p) \in \Delta \text{ and } p \in F\} \cup \\ &\{(f,a,f) \in \mathbb{N}_0^Q \times \Sigma \times \mathbb{N}_0^Q \mid (q_0,a,p) \in \Delta \text{ and } p \in F\}.\end{aligned}$$

Accepting of a word $w \in \Sigma^*$ *is defined as usual [23].*

Generally $\tilde{\mathbb{A}}$ is a non-deterministic automaton with an infinite state set. In the literature, such automata are called multicounter automata [21] and it is known that they accept the iterated shuffle products [26]. For our purposes, deterministic computations of these automata are very important. To analyse these aspects more deeply we use our own notation and proof of the main theorems. In [18], it is shown that $\tilde{\mathbb{A}}$ accepts $P^{\sqcup\!\sqcup}$.

Let for example $P = \{ab\}$ (cf. Figure 8(a)). Then the states $f : Q \to \mathbb{N}_0$ of the automaton $\tilde{\mathbb{P}}$ are described by the sets $\{(q,n) \in Q \times \mathbb{N}_0 | f(q) = n \neq 0\}$.

$$\emptyset \xrightarrow{a} \{(\mathrm{II},1)\} \xrightarrow{a} \{(\mathrm{II},2)\} \xrightarrow{b} \{(\mathrm{II},1)\} \xrightarrow{b} \emptyset$$

is the only computation of $aabb \in P^{\sqcup\!\sqcup}$ in $\tilde{\mathbb{P}}$; it is an accepting computation.

Example 3. *Let L be defined by the automaton $\mathbb{L}$ in Figure 5(a) and $P \subset \Sigma^+$ be defined by the automaton $\mathbb{P}$ in Figure 7, then $L \cap P^{\sqcup\!\sqcup}$ is accepted by the product automaton of $\mathbb{L}$ and $\tilde{\mathbb{P}}$ that is given in Figure 10.*

This automaton is strongly connected and isomorphic to $\mathbb{L}$ (without considering final states), which proves that L is based on phase P. The states $(7, \{(\mathrm{VI},1),(\mathrm{II},1)\})$ and $(8, \{(\mathrm{VI},1),(\mathrm{III},1)\})$ show that L is "in this states involved in two phases".

Note that this product automaton, as well as the product automaton in Figure 11(b) and 12(b), is finite and deterministic.

 ISBN: 978-1-61208-184-7

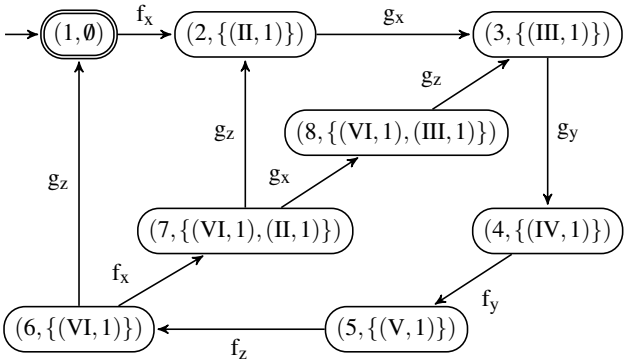


Figure 10. Product automaton of $\mathbb{L}$ and $\overset{\sqcup\!\sqcup}{\mathbb{P}}$

VI. Sufficient Conditions for a Class of Liveness Properties

The following definition is the key to sufficient conditions for strong uniformly parameterised always-eventually properties.

Definition 8 (set of closed behaviours)**.** *Let $B, M \subset \Sigma^*$. M is a* set of closed behaviours *of B, iff $x^{-1}(B) = B$ for each $x \in B \cap M$.*

In Figure 10, the initial state $(1,\emptyset)$ is the only final state of that strongly connected product automaton, so $P^{\sqcup\!\sqcup}$ is a set of closed behaviours of L.

Now, we get a sufficient condition for uniformly parameterised always-eventually properties.

Theorem 1. *Let I, K, $\mathring{I}$ and $\mathring{K}$ be finite index sets with $|\mathring{I}| \leq |I|$ and $|\mathring{K}| \leq |K|$. Let $\mathscr{L}_{IK}$ be a uniformly parameterised system of cooperations and let $\mathscr{C}_{IK} \subset \Sigma_{IK}^*$ be a set of closed behaviours of $\mathscr{L}_{IK}$, such that $\mathscr{L}_{IK} = pre(\mathscr{L}_{IK} \cap \mathscr{C}_{IK})$.*

If $lim(\widehat{\mathscr{L}_{\mathring{I}\mathring{K}}})$ approximately satisfies $(\widehat{\Sigma_{\mathring{I}\mathring{K}}^{*}}\, \mathring{M})^{\omega}$, with $\mathring{M} \subset \Sigma_{\mathring{I}\mathring{K}}^{+}$, then

$$lim(\widehat{\mathscr{L}_{IK}}) \text{ approximately satisfies } \mathscr{A}_{IK}^{\mathring{M}}.$$

For the proof of Theorem 1 see the appendix. The following theorem gives a set of closed behaviours of $\mathscr{L}_{IK}$.

Theorem 2. *Let $P^{\sqcup\!\sqcup}$ be a set of closed behaviours of L and let $\pi_\Phi(P^{\sqcup\!\sqcup})$ resp. $\pi_\Gamma(P^{\sqcup\!\sqcup})$ be a set of closed behaviours of SF resp. SG, then*

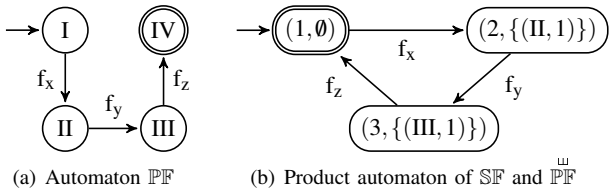
$$\mathscr{C}_{IK} := \bigcap_{(i,k)\in I\times K} (\pi_{ik}^{IK})^{-1}(P^{\sqcup\!\sqcup})$$

is a set of closed behaviours of $\mathscr{L}_{IK}$.

Theorem 2 is proven in [7]. We now show that $\pi_\Phi(P^{\sqcup\!\sqcup})$ is a set of closed behaviours of SF, which is given in Figure 5(b). The automaton $\mathbb{PF}$ in Figure 11(a) is the minimal automaton of $\pi_\Phi(P) \subset \Phi^+$.

By Theorem 3, which is given in the appendix

$$SF \cap \pi_\Phi(P^{\sqcup\!\sqcup}) = SF \cap (\pi_\Phi(P))^{\sqcup\!\sqcup}.$$

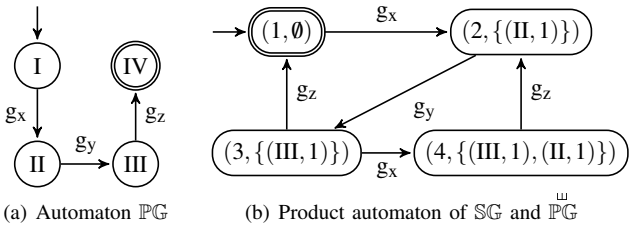


(a) Automaton $\mathbb{PF}$ (b) Product automaton of $\mathbb{SF}$ and $\overset{\sqcup\!\sqcup}{\mathbb{PF}}$

Figure 11. Automaton $\mathbb{PF}$ and product automaton of $\mathbb{SF}$ and $\overset{\sqcup\!\sqcup}{\mathbb{PF}}$

So, $SF \cap \pi_\Phi(P^{\sqcup\!\sqcup})$ is accepted by the product automaton of $\mathbb{SF}$ and $\overset{\sqcup\!\sqcup}{\mathbb{PF}}$ that is depicted in Figure 11(b). By the same argument as for the product automaton of $\mathbb{L}$ and $\overset{\sqcup\!\sqcup}{\mathbb{P}}$ SF is based on $\pi_\Phi(P)$, and $\pi_\Phi(P^{\sqcup\!\sqcup})$ is a set of closed behaviours of SF.

Likewise, the automaton $\mathbb{PG}$ in Figure 12(a) is the minimal automaton of $\pi_\Gamma(P) \subset \Gamma^+$, $SG \cap \pi_\Gamma(P^{\sqcup\!\sqcup})$ is accepted by the product automaton of $\mathbb{SG}$ and $\overset{\sqcup\!\sqcup}{\mathbb{PG}}$ in Figure 12(b), SG is based on $\pi_\Gamma(P)$, and $\pi_\Gamma(P^{\sqcup\!\sqcup})$ is a set of closed behaviours of SG.



(a) Automaton $\mathbb{PG}$ (b) Product automaton of $\mathbb{SG}$ and $\overset{\sqcup\!\sqcup}{\mathbb{PG}}$

Figure 12. Automaton $\mathbb{PG}$ and product automaton of $\mathbb{SG}$ and $\overset{\sqcup\!\sqcup}{\mathbb{PG}}$

So, by Figure 10, 11(b) and 12(b) all assumptions of Theorem 2 are fulfilled. (6)

Now to apply Theorem 1 together with Theorem 2 it remains to find conditions such that each $u \in \mathscr{L}_{IK}$ is prefix of some $v \in \mathscr{L}_{IK} \cap \mathscr{C}_{IK}$. This set of closed behaviours $\mathscr{C}_{IK}$ consists of all words $w \in \Sigma_{IK}^*$, in which all phases are completed.

Considering example 2, we have shown that each phase is initiated by an F-action (Figure 7), each F-partner is involved in at most one phase (Figure 11(b)), and, each G-partner is involved in at most two phases (Figure 12(b)).

Now to construct for each $u \in \mathscr{L}_{IK}$ a $v \in \mathscr{L}_{IK} \cap \mathscr{C}_{IK}$ with $u \in \text{pre}(v)$ one may imagine that the following strategy could work.

1) For each G-partner involved in two phases, complete one of this phases.
2) For each G-partner involved in one phases, complete this phase.
3) Complete the phases, where only an F-partner is involved in.

If L is based on P, SF based on $\pi_\Phi(P)$ and SG based on $\pi_\Gamma(P)$, then by Theorem 3 the assumptions of Theorem 2 imply $L = \text{pre}(L \cap P^{\sqcup\!\sqcup})$, $SF = \text{pre}(SF \cap \pi_\Phi(P^{\sqcup\!\sqcup}))$ and $SG = \text{pre}(SG \cap \pi_\Gamma(P^{\sqcup\!\sqcup}))$.

This is in [7] the starting point of a more general form of such a "completion (of phases) strategy", where also "success conditions" for that strategy are given. It is shown, that under certain regularity restrictions these conditions can be verified by semi-algorithms based on finite state methods. These restrictions are:

The product automata as in Figure 10, 11(b) and 12(b) must be finite and deterministic. (7)

We only get semi-algorithms but no algorithms, because the product automata are constructed step by step and this procedure does not terminate if the corresponding product automaton is not finite.

Using (7), (3) and the Theorems 1 and 2, the approximate satisfaction of uniformly parameterised always-eventually properties can be *verified by semi-algorithms based on finite state methods*. This verification method only depends on L, SF, SG, P and $\mathring{M}$ and *doesn't refer to the general index sets I and K*.

In [7], it is shown that the success conditions are fulfilled in example 2. So, by (4), Theorem 1, Theorem 2 and (6) in example 2 $\lim(\widehat{\mathscr{L}_{IK}})$ approximately satisfies $\mathscr{A}_{IK}^{\mathring{P}}$ for each finite index sets I and K, where $\mathring{P}$ is defined in (5).

VII. Conclusions and Future Work

The main result of this paper is a finite state verification framework for *uniformly parameterised reliability properties*. The uniformly parameterisation of reliability properties exactly fits to the scalability and reliability issues of complex systems and systems of systems, which are characterised by the composition of a set of identical components, interacting in a uniform manner described by the schedules of the partners.

In this framework, the concept of structuring cooperations into phases enables completion of phases strategies. Consistent with this, corresponding success conditions can be formalised [7], which produce finite state semi-algorithms (independent of the concrete parameter setting) to verify reliability properties of uniformly parameterised cooperations. The next step should be to integrate these semi-algorithms in our SH verification tool [25].

Furthermore, we plan a generalisation of the presented approach to systems whose global behaviour is composed of behavioural patterns. The aim is, to eventually derive a set of construction principles for reliable parameterised systems.

Another future work perspective is the application of the approach presented in this paper to the Security Modeling Framework (SeMF) [27]. In SeMF, beside system behaviour, also local views of agents and agents knowledge about system behaviour are considered.

Acknowledgement

Roland Rieke developed the work presented here in the context of the project MASSIF (ID 257475) being co-funded by the European Commission within FP7.

Appendix

A. Basic Notations

The set of all infinite words over Σ is defined by

$$\Sigma^\omega = \{(a_i)_{i\in\mathbb{N}} | a_i \in \Sigma \text{ for each } i \in \mathbb{N}\},$$

where $\mathbb{N}$ denotes the set of natural numbers. On Σ^ω a *left concatenation* with words from Σ^* is defined. Let $u = b_1 \dots b_k \in \Sigma^*$ with $k \geq 0$ and $b_j \in \Sigma$ for $1 \leq j \leq k$ and $w = (a_i)_{i\in\mathbb{N}} \in \Sigma^\omega$ with $a_i \in \Sigma$ for all $i \in \mathbb{N}$, then $uw = (x_j)_{j\in\mathbb{N}} \in \Sigma^\omega$ with $x_j = b_j$ for $1 \leq j \leq k$ and $x_j = a_{j-k}$ for $k < j$. For $w \in \Sigma^\omega$ the prefix set $\text{pre}(w) \subset \Sigma^*$ is defined by $\text{pre}(w) = \{u \in \Sigma^* |$ it exists $v \in \Sigma^\omega$ with $uv = w\}$. For $L \subset \Sigma^*$ the *ω-language* $L^\omega \subset \Sigma^\omega$ is defined by $L^\omega = \{(a_i)_{i\in\mathbb{N}} \in \Sigma^\omega |$ it exists a strict monotonically increasing function $f : \mathbb{N} \to \mathbb{N}$ with $a_1 \dots a_{f(1)} \in L$ and $a_{f(i)+1} \dots a_{f(i+1)} \in L$ for each $i \in \mathbb{N}\}$. $f : \mathbb{N} \to \mathbb{N}$ is called *strict monotonically increasing* if $f(i) < f(i+1)$ for each $i \in \mathbb{N}$.

B. Proof of Theorem 1

To prove Theorem 1 the following lemma is needed.

Lemma 1.

$$\mathscr{L}_{IK} \supset \mathscr{L}_{I'K'} \text{ for } I' \times K' \subset I \times K.$$

For the proof of Lemma 1 see [18] (proof of Theorem 1).

Proof: Proof of Theorem 1.

If $\lim(\widehat{\mathscr{L}_{\mathring{I}\mathring{K}}})$ approximately satisfies $(\widehat{\Sigma_{\mathring{I}\mathring{K}}}^* \mathring{M})^\omega$, then by (3) (with $u = \varepsilon$) there exists $v \in \mathscr{L}_{\mathring{I}\mathring{K}}$ with $v^{-1}(\mathscr{L}_{\mathring{I}\mathring{K}}) \cap \mathring{M} \neq \emptyset$. As $\iota_{I'K'}^{\mathring{I}\mathring{K}}$ is an isomorphism

$$\iota_{I'K'}^{\mathring{I}\mathring{K}}(\mathscr{L}_{\mathring{I}\mathring{K}}) = \mathscr{L}_{I'K'} \text{ and}$$
$$(\iota_{I'K'}^{\mathring{I}\mathring{K}}(v))^{-1}(\mathscr{L}_{I'K'}) \cap \iota_{I'K'}^{\mathring{I}\mathring{K}}(\mathring{M}) \neq \emptyset. \quad (8)$$

As $\mathscr{C}_{IK}$ is a set of closed behaviours of $\mathscr{L}_{IK}$ and each $u \in \mathscr{L}_{IK}$ is prefix of some $v \in \mathscr{L}_{IK} \cap \mathscr{C}_{IK}$, there exists $x \in u^{-1}(\mathscr{L}_{IK})$ with $(ux)^{-1}(\mathscr{L}_{IK}) = \mathscr{L}_{IK}$.

By Lemma 1 $\mathscr{L}_{IK} \supset \mathscr{L}_{I'K'}$ for each $I' \subset I$ and $K' \subset K$, so $(ux)^{-1}(\mathscr{L}_{IK}) \supset \mathscr{L}_{I'K'}$.

Now (8) implies

$$(\iota_{I'K'}^{\mathring{I}\mathring{K}}(v))^{-1}((ux)^{-1}(\mathscr{L}_{IK})) \cap \iota_{I'K'}^{\mathring{I}\mathring{K}}(\mathring{M}) \neq \emptyset. \quad (9)$$

As $(\iota_{I'K'}^{\mathring{I}\mathring{K}}(v))^{-1}((ux)^{-1}(\mathscr{L}_{IK})) = (ux\iota_{I'K'}^{\mathring{I}\mathring{K}}(v))^{-1}(\mathscr{L}_{IK})$, (9) and (3) complete the proof of Theorem 1. ■

C. Homomorphism Theorem for $P^{\sqcup\!\sqcup}$

Theorem 3 (homomorphism theorem for $P^{\sqcup\!\sqcup}$)**.**
Let $\mu : \Sigma^* \to \Sigma'^*$ be an alphabetic homomorphism, then holds

$$\mu(P^{\sqcup\!\sqcup}) = (\mu(P))^{\sqcup\!\sqcup}.$$

For the proof of Theorem 3 see [7] (proof of Theorem 6).

 ISBN: 978-1-61208-184-7

REFERENCES

[1] S. Bullock and D. Cliff, "Complexity and emergent behaviour in ICT systems," Hewlett-Packard Labs, Tech. Rep. HP-2004-187, 2004.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009.

[3] Z. Benenson, F. Freiling, T. Holz, D. Kesdogan, and L. Penso, "Safety, liveness, and information flow: Dependability revisited," in *Proceedings of the 4th ARCS International Workshop on Information Security Applications*, pp. 56–65.

[4] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "Donar: decentralized server selection for cloud services," in *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 231–242.

[5] A. Shraer, C. Cachin, A. Cidon, I. Keidar, Y. Michalevsky, and D. Shaket, "Venus: verification for untrusted cloud storage," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, ser. CCSW '10. New York, NY, USA: ACM, 2010, pp. 19–30.

[6] A. Avizienis, J.-C. Laprie, B. Randell, and C. E. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Trans. Dependable Sec. Comput.*, vol. 1, no. 1, pp. 11–33, 2004.

[7] P. Ochsenschläger and R. Rieke, "Behaviour Properties of Uniformly Parameterised Cooperations," Fraunhofer SIT, Tech. Rep. SIT-TR-2010/2, 2010.

[8] B. Alpern and F. B. Schneider, "Defining Liveness," *Information Processing Letters*, vol. 21, no. 4, pp. 181–185, October 1985.

[9] U. Nitsche and P. Ochsenschläger, "Approximately Satisfied Properties of Systems and Simple Language Homomorphisms," *Information Processing Letters*, vol. 60, pp. 201–206, 1996.

[10] M. R. Clarkson and F. B. Schneider, "Hyperproperties," *Computer Security Foundations Symposium, IEEE*, vol. 0, pp. 51–65, 2008.

[11] C. N. Ip and D. L. Dill, "Verifying Systems with Replicated Components in Murφ," *Formal Methods in System Design*, vol. 14, no. 3, pp. 273–310, 1999.

[12] F. Derepas and P. Gastin, "Model Checking Systems of Replicated Processes with SPIN," in *Proceedings of the 8th International SPIN Workshop on Model Checking Software (SPIN'01)*, ser. Lecture Notes in Computer Science, M. B. Dwyer, Ed., vol. 2057. Toronto, Canada: Springer, May 2001, pp. 235–251.

[13] Y. Lakhnech, S. Bensalem, S. Berezin, and S. Owre, "Incremental Verification by Abstraction." in *TACAS*, ser. Lecture Notes in Computer Science, T. Margaria and W. Yi, Eds., vol. 2031. Springer, 2001, pp. 98–112.

[14] S. Basu and C. R. Ramakrishnan, "Compositional analysis for verification of parameterized systems," *Theor. Comput. Sci.*, vol. 354, no. 2, pp. 211–229, 2006.

[15] R. Milner, *Communication and Concurrency*, ser. International Series in Computer Science. NY: Prentice Hall, 1989.

[16] J. C. Bradfield and C. Stirling, "Modal Logics and Mu-Calculi: An Introduction," in *Handbook of Process Algebra*, J. A. Bergstra, A. Ponse, and S. A. Smolka, Eds. Elsevier Science, 2001, ch. 1.4.

[17] T. E. Uribe, "Combinations of Model Checking and Theorem Proving," in *FroCoS '00: Proceedings of the Third International Workshop on Frontiers of Combining Systems*. London, UK: Springer-Verlag, 2000, pp. 151–170.

[18] P. Ochsenschläger and R. Rieke, "Uniform Parameterisation of Phase Based Cooperations," Fraunhofer SIT, Tech. Rep. SIT-TR-2010/1, 2010.

[19] M. Jantzen, "Extending Regular Expressions with Iterated Shuffle," *Theor. Comput. Sci.*, vol. 38, pp. 223–247, 1985.

[20] J. Jedrzejowicz and A. Szepietowski, "Shuffle languages are in P," *Theor. Comput. Sci.*, vol. 250, no. 1-2, pp. 31–53, 2001.

[21] H. Björklund and M. Bojanczyk, "Shuffle Expressions and Words with Nested Data," in *Mathematical Foundations of Computer Science 2007*, 2007, pp. 750–761.

[22] P. Ochsenschläger and R. Rieke, "Security Properties of Self-similar Uniformly Parameterised Systems of Cooperations," in *Proceedings of the 19th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP)*. IEEE Computer Society, February 2011.

[23] J. Sakarovitch, *Elements of Automata Theory*. Cambridge University Press, 2009.

[24] D. Perrin and J.-E. Pin, *Infinite Words*. Elsevier, 2004, vol. Pure and Applied Mathematics Vol 141.

[25] P. Ochsenschläger, J. Repp, and R. Rieke, "The SH-Verification Tool," in *Proc. 13th International FLorida Artificial Intelligence Research Society Conference (FLAIRS-2000)*. Orlando, FL, USA: AAAI Press, May 2000, pp. 18–22.

[26] J. Jedrzejowicz, "Structural Properties of Shuffle Automata," *Grammars*, vol. 2, no. 1, pp. 35–51, 1999.

[27] A. Fuchs, S. Gürgens, and C. Rudolph, "Towards a Generic Process for Security Pattern Integration," in *Trust, Privacy and Security in Digital Business, 6th International Conference, TrustBus 2009, Linz, Austria, September 3–4, 2009, Proceedings*. Springer, 2009.

 ISBN: 978-1-61208-184-7

System Reverse Engineering to Requirements and Tests

Qi Zhang
Fakultät für Informatik
Universität der Bundeswehr München
Neubiberg, Germany
qi.zhang@unibw.de

Andreas Karcher
Fakultät für Informatik
Universität der Bundeswehr München
Neubiberg, Germany
andreas.karcher@unibw.de

Abstract—The long operational phase of products in the aviation industry demands constant maintenance and adaptation of its systems in order to fulfill to the demands of customers and the market and prevent obsolescence. For this purpose, a display system – including all documentation and tools - will be transferred from development to a maintenance department to be supported there over a long duration. To be able to modify the transferred system, maintenance first needs to achieve an understanding of the system. Due to the long development phase, requirements of embedded systems in this domain are most often historically evolved, and only in rare cases formally documented. Typical weaknesses of informal requirements, such as incompleteness or inconsistency, pervade the subsequent levels of the system's life cycle ranging from design to testing. Insufficiently documented system behavior can be increased according to the methods of reverse engineering by analysis of the requirements and design. This article intends to analyze the state of documentation of an existing aviation system in order to create a solution for completing and improving legacy requirements and tests.

Keywords-Avionic; Embedded Multifunction Displays; Maintenance; Requirements; Reverse Engineering.

I. INTRODUCTION

The development of embedded avionic systems is performed by several suppliers, who derive detailed requirements and test cases from high level specifications of an original equipment manufacturer (OEM) in order to develop the commissioned components. All requirements are expected to be well-defined, comprehensible and testable [1] in order to allow integration and interconnection by the OEM according to „systems thinking" [2]. The system comprehension is impaired by conditions such as lack of traceability, inconsistency or incompleteness, especially within and between text-based requirements – company-internal and -external. The deficient documentation leads to difficulties when modifying requirements and test cases during the maintenance phase. Reverse engineering can assist in some ways to the proper maintenance of these legacy systems [8]. The article at hand reflects reverse engineering in the context of software maintenance and further development in the aviation industry. Starting point is a description of the general challenge maintaining legacy avionic display systems. After analyzing the current state of the art for requirement and reverse engineering, the industrial application is presented. Finally, a domain specific concept for improving requirements and tests by reverse engineering is provided.

II. RELATED RESEARCH

Research into requirement engineering has, in the past, been driven by a trend away from documentation centralization towards model centralization [3]. To improve consistency and completeness of requirements, languages and models have been developed [4] [5] which are compatible with extensible markup language (XML). These are governed by standards specifically tuned for loss-less exchange of requirement data between OEM and suppliers [6]. The focus of these methods lies exclusively on the improvement of requirement quality. Whereas the application protocol AP 233 of the ISO standard STEP 10303 [7] offers models for formalization of requirement data, from which areas like testing can also benefit. The underlying idea of this standard is the development of a tool-independent format for long-term archiving and loss-less exchange of data within a complex system. The standard is non-specialized due to its wide range of applicability and requires being adapted to the system in question before its application.

The technologies and methods mentioned above refer to the development phase. In practice, there is usually a finished product given for which the quality is to be improved. Research in the area of reverse engineering is specialized on the analysis of existing products [8]. Reverse engineering is the process of analyzing an existing system in order to identify its components and their interaction, illustrating it in a different or more abstracted form [9]. Although reverse engineering would be applicable to the entire life cycle – starting with existing source code, re-creation of the design and mapping of inherited requirements – the main focus in this area up until a few years ago was on „implementation and design artifacts" [10]. For this reason, Knodel, Koschke, and Mende [10] require the expansion of reverse engineering procedures onto all levels, such as testing and requirements. Similar results are drawn by

 ISBN: 978-1-61208-184-7

Canfora and Di Penta [11] in their article „New Frontiers of Reverse Engineering". There, a road map for reverse engineering is presented, where it is made clear that reverse engineering can go further than recovering design artifacts: requirements are also an important output that can be produced by reverse engineering.

In the following, a concept based on the state of current technologies for improvement of information acquisition between existing requirements and test cases is presented, using the example of the maintenance of symbols on a helicopter's multifunctional display (MFD).

III. INDUSTRIAL APPLICATION

MFDs are used to display flight data and warnings in the helicopter's cockpit. The embedded software has got to be maintained, and the displayed symbols have got to be changed according to customer expectations where necessary. In the context of this application the MFD was developed by a supplier for air traffic control systems, who derived own specifications and test cases from given format specifications (FOS) of the OEM in order to develop the hardware, software and symbol design.

Since the informal FOS are hard to manage, e.g., data regarding size and position is incomplete, the suppliers were free to determine the method and tool with which to create the displayed symbols. The supplier derived formal software requirement specifications (SRS) to describe the logic of MFD input signals. Each combination of inputs signals offers an output signal triggering the desired symbol to be displayed on the MFD. The related test cases are used to verify input signals (as described by the SRS) and the resulting output symbols (as described by the FOS). The overall requirements and test cases can be correlated with each other but not with the FOS. The symbols' design was realized using a human-machine interface (HMI) modeling and display graphics tool, which stores the size and position of the symbols in human non-readable meta files (Fig. 1). This tool was also used to create pictures for visualization of the symbols in the FOS.

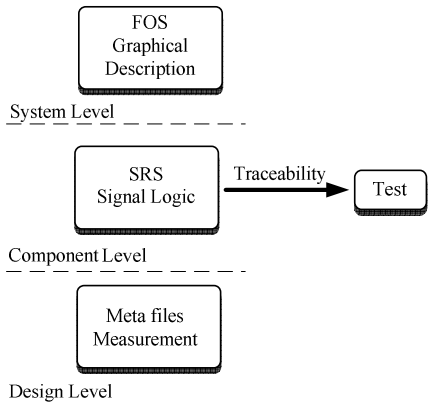


Figure 1. Current traceability of symbol documentation

For this reason, any symbol modification requires to be first implemented in the HMI graphics tool before it can be adapted to the requirements or presented to the customer. The following Figure 2 displays the process for a symbol modification.

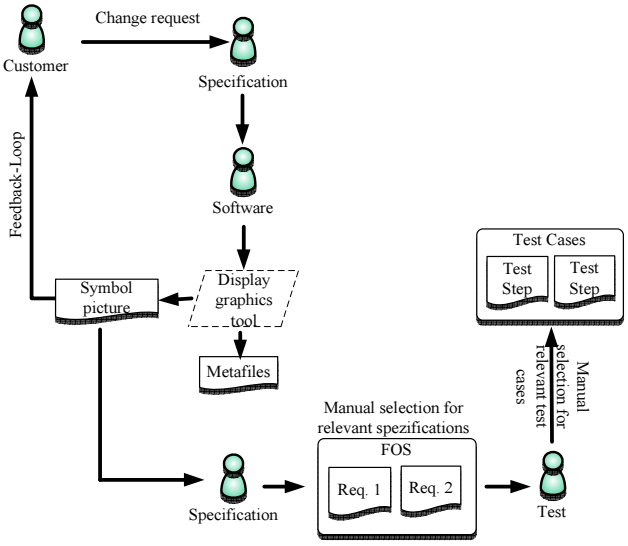


Figure 2. Current process for graphical changes in the MFD

Figure 2 shows that no data is provided from development regarding the criteria on which test cases were developed. Symbols are only tested as output in connection with the signal logic, not as a separate unit. This means that test cases such as „is the symbol ambiguous?" or „can critical conditions like 'white text on white background' occur?" do not exist. For this reason, symbol tests are incomplete.

This missing data is to be rectified during maintenance in order to improve the quality of inherited requirements and test cases of the MFD.

IV. PROPOSED IMPLEMENTATION CONCEPT

With the symbol requirements from development phase being incomplete, a collection of all symbol data in a symbol library is proposed. The idea of maintaining the symbols in a library/database is derived from geographical information systems and mine mapping [12]. The symbol library is to store, in an appropriate structure, data such as a unique attribution, conditional and positional information and its traceability to all relevant requirements and test cases. This builds the foundation for the collection and interconnection of all data from different documents. For processing symbol data like visualization data exchange is done via XML schema. XML offers, besides its platform independence [5], advantages such as translation of the symbol data from and into a database or a scalable vector graphics (SVG)-based graphical user interface (GUI). This is meant to facilitate the access to relevant documents in the case of a modification. Additionally, the collection and representation of data are supportive to the creation and modification of test cases for symbol verification.

 ISBN: 978-1-61208-184-7

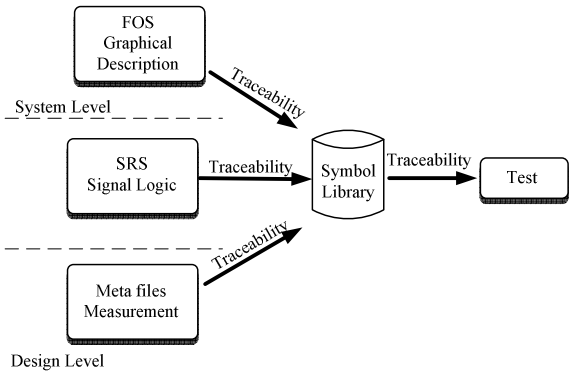


Figure 3. Interconnection of symbol data by a library

With the possibility to display symbols via SVG even before their implementation, the necessity of copying symbol pictures from the display graphics tool to the FOS is eliminated. Furthermore, modifications of symbols can be presented to the customer even in early development stages, without implementation on the design tool. The symbol description in SVG is created by an extensible style-sheet language transformation (XSLT) of the XML data. The XML schema provides the verification of the symbols as an independent artifact. Queries such as „do duplicate symbols exist", or „is the condition 'white text on white background' possible" or „is the new symbol schema-conform" are easily implemented in the database. These queries can serve as criteria for the creation of new test cases. The following section lays out the creation and simulated application of the solution prototype in the industrial environment.

V. EVALUATION IN THE INDUSTRIAL CONTEXT

For better evaluation of the current symbol requirements, an XML schema was created based on current documentation. The structuring and classification of the symbols is performed by assignment of defined conditions, as described in the FOS and in meta files. Figure 4 shows the implementation of the XML schema by example of the symbol „FIRE", which is presented by white writing on a red rectangle. Universal attributes, such as colors and fonts, are defined globally so that they need to be changed only in one place in the case of a modification. The content of the FOS and of the meta files are not consistently structured, so that an automatic query of the data is difficult to realize. For the implementation of the symbol „FIRE", the data was entered manually. Figures 5 and 6 show the XSLT of the prototype based on the SVG and the visualization of the symbol via SVG. It turned out that the current data processing of the development documents from which the XML schema has been created does not represent an adequate structure for a database transition or an XSLT to SVG (Fig. 5). For example, a rectangle is defined in the design tool by the coordinates of two corner points (compare „Rectangle" in Fig. 4), while the same element is defined in SVG by the coordinates of the top left corner, width and height (compare „rect" Fig. 5).

```
<Symbol>
    <Warning>
        <Title>
            <Name>FIRE</Name>
            <Pen_Color>white</Pen_Color>
            <Pen_ColorID>0</Pen_ColorID>
            <Set_Font>1</Set_Font>
            <Set_Linestyle>1</Set_Linestyle>
            <Coord>
                <x_Pos>2</x_Pos>
                <y_Pos>8</y_Pos>
            </Coord>
        </Title>
        <Background>
            <Fill_Color>red</Fill_Color>
            <Fill_ColorID>0</Fill_ColorID>
            <Type>
                <Rectangle>
                    <x1_Pos>0</x1_Pos>
                    <y1_Pos>0</y1_Pos>
                    <x2_Pos>150</x2_Pos>
                    <y2_Pos>50</y2_Pos>
                </Rectangle>
            </Type>
        </Background>
    </Warning>
</Symbol>
```

Figure 4. current data structure by example of „FIRE"

```
<!-- variables for background -->
<xsl:variable name="back_color" select="Symbol/Warning/Background/Fill_Color"/>
<xsl:variable name="back_x1_pos" select="Symbol/Warning/Background/Type/Rectangle/x1_Pos"/>
<xsl:variable name="back_y1_pos" select="Symbol/Warning/Background/Type/Rectangle/y1_Pos"/>
<xsl:variable name="back_x2_pos" select="Symbol/Warning/Background/Type/Rectangle/x2_Pos"/>
<xsl:variable name="back_y2_pos" select="Symbol/Warning/Background/Type/Rectangle/y2_Pos"/>

<!-- variables for text -->
<xsl:variable name="text_color" select="Symbol/Warning/Title/Pen_Color"/>
<xsl:variable name="text_x_pos" select="Symbol/Warning/Title/Coord/x_Pos"/>
<xsl:variable name="text_y_pos" select="Symbol/Warning/Title/Coord/y_Pos"/>
<xsl:variable name="text_size" select="Symbol/Warning/Title/Set_Linestyle"/>
<xsl:variable name="text_font" select="Symbol/Warning/Title/Set_Font"/>

<!-- svg output -->
<svg width="{$back_x2_pos}px" height="{$back_y2_pos}px" xmlns="http://www.w3.org/2000/svg">
<rect x="{$back_x1_pos}" y="{$back_y1_pos}" height="{$back_y2_pos - $back_y1_pos}"
width="{$back_x2_pos - $back_x1_pos}" style="fill:{$back_color}"/>
<text x="{$text_x_pos}" y="{$text_y_pos}" style="font-size:{$text_size}px; font-family:
{$text_font}; fill:{$text_color}">
<xsl:value-of select="Symbol/SysStat_Alarm_Symbols/Generic_Symbol/Warning/Title/Name"/>
</text>
</svg>
```

FIRE

Figure 5. XSLT based on SVG by example of „FIRE"

In long-living products, dependence on tools always comes with the risk of obsolescence. For this reason, the schema is currently adjusted to the standardized SVG symbol description in order to provide wider and more flexible tool compatibility for data processing. As the schema is still under development, the process sequence of a symbol modification was simulated with an idealized implementation concept. Starting point of the simulation is a graphical symbol modification which is included into the symbol library by a specification engineer. The symbol can now be displayed in a GUI simulation and used in a presentation for internal or external customers by using SVG format. This prevents potential misunderstandings or misinterpretations, which increases the efficiency of the modification process. The symbol pictures generated with SVG can also be implemented into the FOS. All documents affected by the modification (requirement and testing) can be displayed automatically due to their linking within the library; the error-prone manual selection can be omitted.

 ISBN: 978-1-61208-184-7

SQL queries are used to eliminate redundancy or other interference with existing symbols. These queries serve to validate the modification and create new test cases.

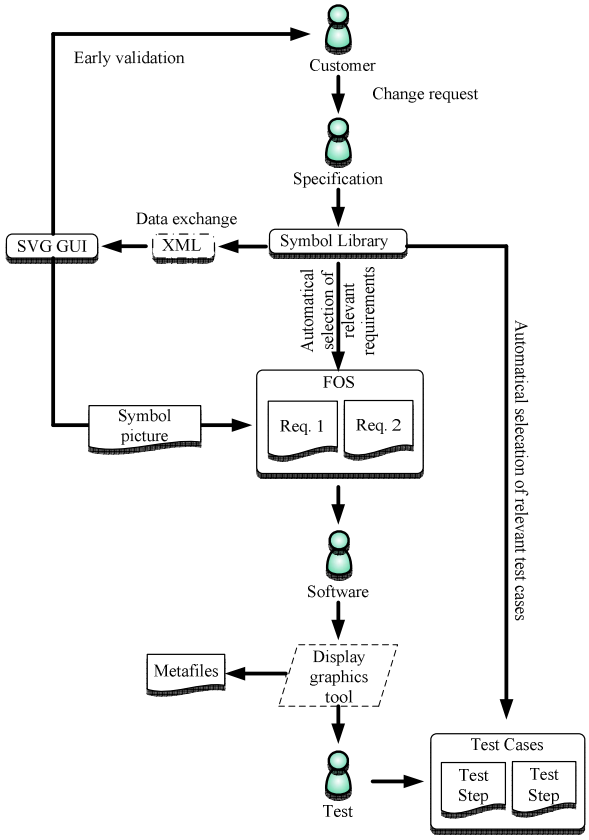


Figure 6. Simulated modification process based on the implementation concept

This implementation concept offers benefits for the specification and test stakeholders by partial automation of several process sections. The simulation of the implementation concept (Fig. 6) makes the actual process of modification more efficient by eliminating the feedback loop (Fig. 2) and automatically selecting all relevant documents. This way, the process of modification is more resistant to errors that may occur in manual research. The whole process is compliant to the „V-Model XT" standard. The resulting benefits for testing offers potential to develop and justify new symbol test cases derived from the database queries. Some further advantages are higher traceability of requirements and test cases, transparency and higher quality of symbol requirements, and improved verification and modification of symbols due to the formal definitions.

VI. CONCLUSION AND PROSPECTS

The implementation concept offers advantages for both requirements and tests of MFDs. Next the adaptation of XML schema is planned providing compliance to SVG. After this, a selection of MFD symbols is used to create a prototype for the implementation of real modification processes, in order to measure the results. The prototypes will be used after evaluation in the upcoming transfer of Supplier Software. Since the concept only provides a completion of symbol information of current specifications a subsequent goal will be the formalization of existing FOS and test cases to facilitate the exchange of their data with the library. This way the modification process from requirement up to testing can be (partially) automated. Developing a configuration management tool for the library is also required. This implementation concept intends, by example of the MFD, to expand current methods of reverse engineering to the levels of requirements and tests.

REFERENCES

[1] K. Bender, Embedded Systems - qualitätsorientierte Entwicklung: Qualitätssicherung bei Embedded Software: Springer Berlin Heidelberg; Auflage: 1, 2005.

[2] W. F. Daenzer and F. Huber, Systems Engineering: Methodik und Praxis, 11th ed.: Zürich Verl. Industrielle Organisation, 2002.

[3] T. Weilkiens, Systems Engineering mit SysML / UML. Modellierung, Analyse, Design: Dpunkt Verlag; Auflage: 1., 2006.

[4] M. Broy, V. Esperstedt, F. Houdek, K. Pohl, and H. Wußmann, "Leitfaden für modellbasiertes Requirements-Engineering und -Management softwareintensiver Eingebetteter Systeme — REMsES —," ed. Essen: REMsES-Konsortium, 2009.

[5] T. Wien, E. Carlson, T. Stålhane, and F. Reichenbach, "Reducing development costs in industrial safety projects with CESAR," Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference, pp. 1-4, 13-16 Sept. 2010.

[6] M. Adedjouma, H. Dubois, and F. Terrier, "Requirements Exchange: From Specification Documents to Models," Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference, pp. 350 - 354, 27-29 April 2011.

[7] D. Kretz, J. Militzer, T. Neumann, and T. Teich, "Developing an integrated solution for generative process planning based on ISO standard 10303 " Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference, pp. 61 - 65, 27-29 May 2011.

[8] H.-J. Choi and S. A. Fahmi, "Software Reverse Engineering to Requirements," Convergence Information Technology, 2007. International Conference, pp. 2199 - 2204, 21-23 Nov. 2007.

[9] E. J. Chikofsky and J. H. Cross, II "Reverse engineering and design recovery: a taxonomy " Software, IEEE vol. 7, pp. 13 - 17, Jan. 1990.

[10] J. Knodel, R. Koschke, and T. Mende, "Reuse in Reverse Engineering," Fraunhofer IESE Kaiserslautern, 2006.

[11] G. Canfora and M. Di Penta, "New Frontiers of Reverse Engineering " Future of Software Engineering, 2007. FOSE '07 pp. 326 - 341, 23-25 May 2007.

[12] S. Li and H. Liu, "COM-based symbol library extension for mine mapping," Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International vol. 6, pp. 4150 - 4152, 20-24 Sept. 2004

 ISBN: 978-1-61208-184-7

Context ontology for Event-Driven Information Systems

Ana Šaša Bastinos, Marjan Krisper
Information Systems Laboratory
Faculty of Computer and Information Science, University of Ljubljana
Ljubljana, Slovenia
{ana.sasa, marjan.krisper}@fri.uni-lj.si

***Abstract*—Event-driven architecture and complex event processing have become important topics in achieving business reactivity and proactivity. However, the use of these approaches in real-world solutions remains limited. One of the reasons for this is insufficient support for semantics in capturing and defining of complex events. In this paper, we address this problem. We present a framework for ontology-based support for complex events, which allows for semantically enriched event definitions and automatic recognition. In order to automatically recognize complex events, an appropriate event definition basis has to be defined. The paper represents a continuation of our previous work by enhancing this basis and applying the aspectual model from the field of linguistics to our base event ontology. We believe that this framework will provide a generic approach that will allow for complex event recognition for events of various complexities and from different domains.**

Keywords-event; complex event; ontology; aspectual model.

I. INTRODUCTION

With increasing demands for agility and reactivity of business processes, scientific circles and leading information technology companies have paid a lot of attention to EDA (event-driven architecture) and CEP (complex event processing). EDA and CEP enable event-driven systems – systems in which actions result from business events [1]. Despite the recognized need for support of events to improve automation, signal processing, data acquisition, manufacturing, computer aided simulations, and business activity monitoring [1][2], the use of EDA and CEP approaches in real-world solutions remains limited. One of the reasons for this is insufficient support for semantics in capturing and defining of complex events (CE) [3]. Existing EDA and CEP approaches do not take into consideration different expressivity requirements that are needed in definition of a large number of diverse CE. Because of this, detection of CE that require semantically expressive descriptions is not automated and experts are required to monitor business operation in order to determine if a complex event has occurred. Such an approach can decrease reactiveness and proactiveness of an organization [3]. We addressed some of these issues in our previous work presented in [3], where we discussed ontology-based framework for complex events. In this paper, we propose an improved ontology basis for this framework. We improve the semantic event definition and recognition by development of a complex event ontology based on Tremblay's adjustment of Pustejovsky's aspectual model [4]. The purpose of applying this cognitive approach to the complex event ontology is to enable richer semantic complex event definitions that better reflect real world events thus providing an improved mechanism for complex event detection.

The paper is structured as follows. In the next section, we introduce the main concepts of event-driven architectures. In section III., we describe our ontology-based framework for complex events. In section IV, we present Pustejovsky's aspectual model and Tremblay's adjustment of this model. In section V, we present our application of this model to our ontology-based framework for complex events. In section VI, we provide final conclusions and discuss our further work.

II. EVENT-DRIVEN ARCHITECTURE

An entity is considered event-driven when it acts in response to an event. EDA is a paradigm that describes an approach to information systems development with a focus on developing an architecture that has the ability to detect events and react intelligently to them [5]. EDA represents a complement to the service-oriented architecture (SOA), which has become one of the most recognized paradigms in information systems development in the recent years [6]. By enhancing the paradigm of SOA, enterprises can improve their ability for business transformation by implementing event-driven architectures that automatically detect and react to significant business events [5]. An important part of every EDA that enables and predetermines to what level a system is able to detect and respond to complex events is complex event processing (CEP). CEP is computing that performs operations on complex events, including reading, creating, transforming, or abstracting them [7]. CEP systems can be classified as advanced decision support systems [3]. In comparison with other types of decision support systems that are not event driven, CEP systems focus on increasing system reactivity and proactivity based on information carried by member events. Events can be simple or complex. A simple event is as an event that is not an abstraction or composition of other events. A complex event is an event that is an abstraction of other events called its members [7].

III. ONTOLOGY-BASED FRAMEWORK FOR COMPLEX EVENTS

In this section, we discuss results of our previous work [3], which are the basis for the research presented of this

 ISBN: 978-1-61208-184-7

paper. In [3], we observed that semantic descriptions providing effective and expressive models for understanding of CE structures and their processing could be very helpful for definition and automation detection of CEs. We presented a framework that enables highly expressive event models and is based on ontologies and the Web Ontology language (OWL) [8]. We see ontologies and ontology representation languages as an opportunity to effectively deal with CEs in EDAs. Ontologies intrinsically provide a means for highly expressive semantic descriptions. In our framework, they are used to semantically describe CEs through conceptualizations of member events they are composed of. We use OWL (Web Ontology Language) as the ontology representation language, because it provides a very appropriate foundation for CE definitions and has a wide support for reasoning. We developed a service that makes part of an EDA and enables translation of event data to OWL, detection of CEs and their triggering. It can act as an event source and as an event sink, which makes our approach complementary to existing approaches to support CEs that require higher expressivity and semantic descriptions. Our framework has been used in a case study project for electrical distribution domain, where it has been shown to be very useful and has improved the overall system flexibility and reactivity.

Our event-driven architectural framework defines a Complex Event Service (CES) that makes part of the overall event-driven service oriented architecture. Fig. 1 illustrates a high-level structure of the CES.

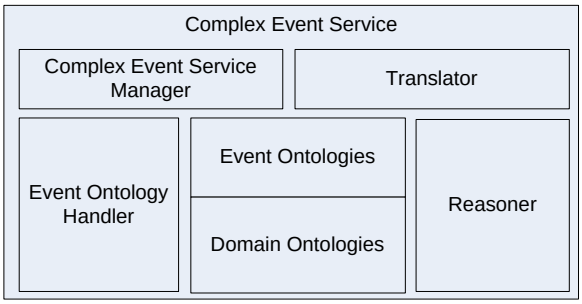


Figure 1. Components of the Complex Event Service

The CES is responsible for detection of CEs based on their complex event definitions and member event occurrences. It acts as an event sink for member events of the CEs it is responsible for, and it acts as an event source for the detected CEs. It catches events and triggers CEs when it detects them based on their definition and member events occurrences. The CES consists of five components (Fig. 1): Complex Event Service Manager, Translator, Ontology, Event ontology handler, and Reasoner. Ontology comprises event and domain ontologies. It is a passive entity that is handled by the Event Ontology Handler and used by the Reasoner to infer new information based on the existing information in the ontology. It comprises information about the domain, about events and event types. An event ontology supports one CE type or several related CE types. We say that an event ontology is subscribed to all events that are members of the CEs that the ontology defines. CES Manager is the component that links the CES with its environment by receiving member events and triggering detected CEs. It also orchestrates the other active components of the CES into a complete process. In the remainder of this section we describe the event ontologies and domain ontologies. For more details about other components of the framework please refer to [3].

Domain ontologies comprise information about the domain where the system is used, for example the electrical distribution domain. For creation and maintenance activities of the domain ontologies, different generic ontology development methodologies that are available can be used. The basic structure of an event ontology is defined by the base event ontology. Fig. 2 illustrates the base event ontology from our previous work.

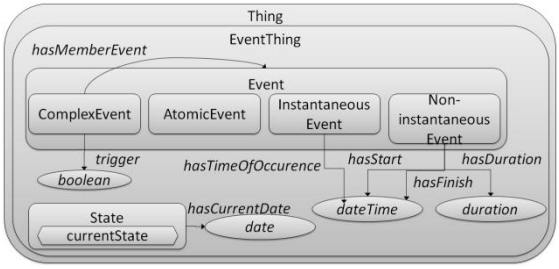


Figure 2. Base event ontology v1.0

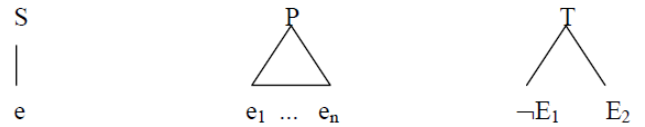
In this paper, we propose an improvement of the base event ontology by applying to it the aspectual model from the field of linguistics. In the next section, we thus introduce this model and in Section V, we present the resulting base event ontology.

IV. EVENT TYPES IN THE LITERATURE ON ASPECT

In 1959, Vendler defined four types of event classes: state, accomplishment, activity, and achievement [10]. Later, Pustejovsky developed an aspectual model in which he only distinguished three main event types [11]:

- States (S), which are single events evaluated relative to no other event (e.g., be sick, love, know),
- Activities or processes (P), which are sequences of events identifying the same semantic expression (e.g., run, push, drag), and
- Transitions (T), which are events identifying a semantic expression evaluated relative to its opposition (e.g., give, build, open, destroy).

If we define ET as the event type domain, and E as the event domain, then we can represent the Pustejovsky's structural representations of states, processes and transitions as illustrated in the following figure:

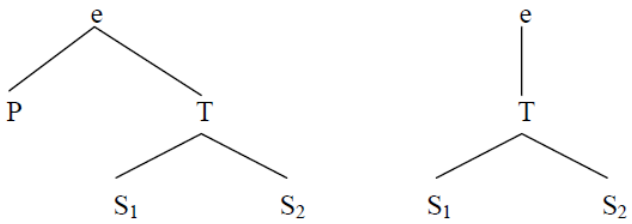


(a) State (S) (b) Process (P) (c) Transition (T)

Figure 3. Structural representations of states, processes and transitions

where e, e1,…en ∈E, and E1,E2∈ET.

Pustejovsky also defined accomplishments and achievements as composite event types and defined their structure. Our work is based on Tremblay's adjustment of Pustejovsky's model:



(a) Accomplishment (b) Achievement

Figure 4. Structural representations of accomplishments and achievements

In his model, an accomplishment is an event composed of a transition from an initial state into the target state, and a process that causes the transition. An achievement is an event composed of a transition from an initial state into the target state (also called achievement without preliminaries).

V. BASE EVENT ONTOLOGY

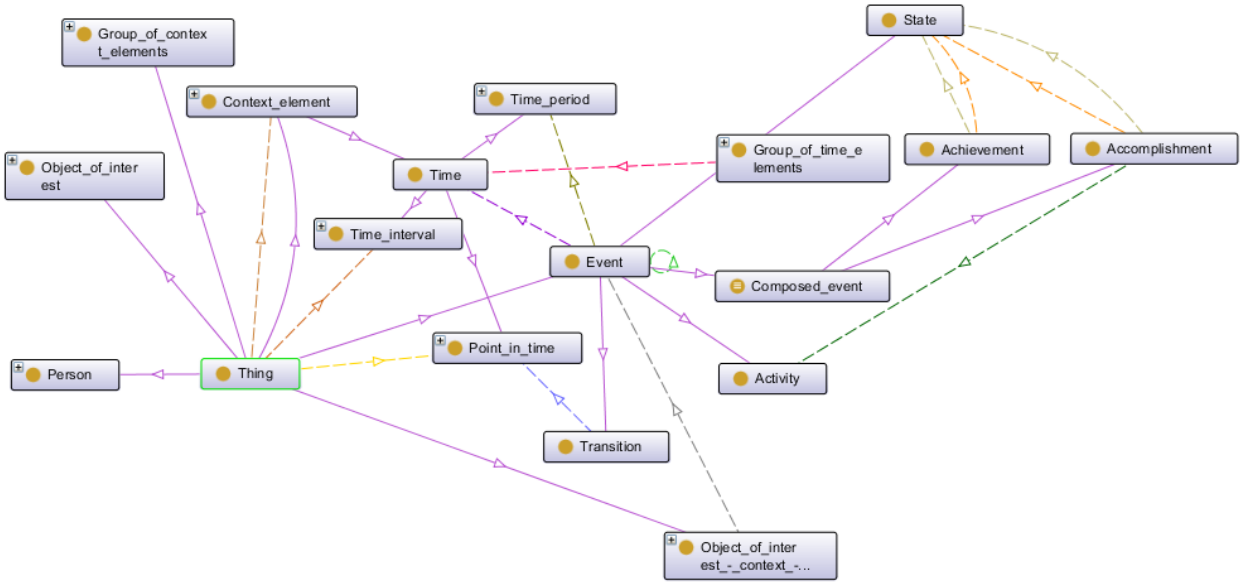
In order to establish an appropriate base event ontology that will satisfy the goal of establishing an expressive base for semantic definitions of complex events, we applied the Tremblay's adjustment of Pustejovsky's aspectual model to the base event ontology. In order to define events we used the concept of a context. In human minds, real-world situations are entertained as contexts [10]. A context is the situation within which something exists or happens, and that can help explain it [9].

The base event ontology is illustrated in Fig. 5. We distinguish between the four categories of events and define object properties between the corresponding classes in order to refleck the structural relations between different event types (Fig. 4). We also define a context element class to represent a general element of an ontology which can represent different concepts important to describe a situation, for example an event context. Object of interest is a class that is used as a basis for decision making, for example when occurrence of a complex event depends on presence or certain state of a specific object. Fig. 4 presents the most important classes and properties of the base event ontology.

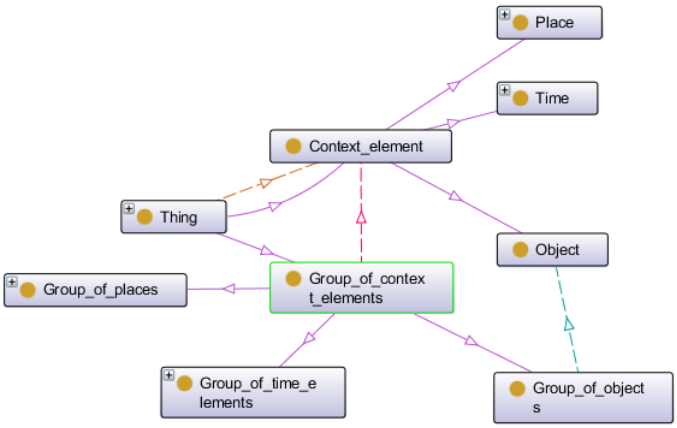
Every event ontology imports at least one domain ontology, the base event ontology and can import one or more other event ontologies. Event ontologies define subclasses of the Event class and its subclasses. Necessary and sufficient membership conditions have to be defined for every complex event, which is represented by the ComposedEvent class. Domain and event ontology concepts can be used to define these conditions. Thus, the necessary and sufficient membership conditions represent the join between the domain and the event concepts. In the event ontology classes and properties are defined that relate an event type to its context.

VI. CONCLUSIONS AND FURTHER WORK

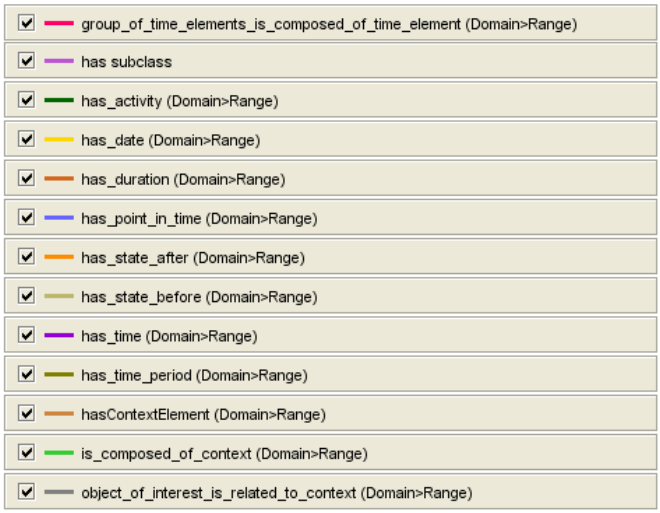
The paper presents an event-driven framework that is based on ontologies. We focused on the base event ontology, which is an important component of the framework. It determines the overall structure for every event ontology and the way the events will be defined. In this paper, we have developed the base event ontology by applying the aspectual model from the field of linguistics, which has studied the event types for several decades.



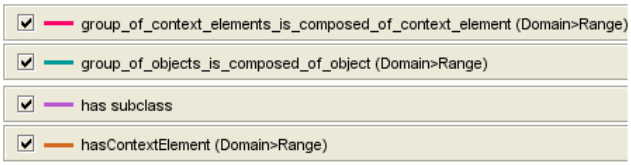
(a) Base event ontology v2.0 visual representation



(b) Base event ontology v2.0 visual representation of context elements



(c) Legend for fig. a



(d) Legend for fig. b

Figure 5. Base event ontology v2.0

The presented base event ontology is work in progress. We are currently preparing a case study for sales domain. In our further work, we will perform several other case studies for different business domains to demonstrate the usefulness of this ontology in CE definition and, more importantly, automatic recognition of different types of CE.

REFERENCES

[1] D. Bugaite D. and O. Vasilecas, **"**Events Propagation from the Business System Level into the Information System Level". In Barry C. et al. (eds.). Information Systems Development. Springer US, p. 252, 2009.

[2] R. Adaikkalavan and S. Chakravarthy, "Event Specification and Processing for Advanced Applications: Generalization and Formalization," Proc. International Conference on Database and Expert Systems Applications, Regensburg, Germany, 2007, pp. 369-379.

[3] A. Sasa, O. Vasilecas, "Ontology-Based Support for Complex Events," Electronics and electrical engineering, vol. 113, no. 7, 2011.

[4] A. Tremblay, "A Cognitive Approach to Accomplishments, Fundamentally Imperfective Accomplishments, and Achievements with and without Preliminaries; A Cognitive Approach to Aspectual Categories," University of Alberta, 2007.

[5] H. Taylor, F. Martinez, A. Yochem, L. Phillips, Event-Driven Architecture: How SOA Enables the Real-Time Enterprise. Addison-Wesley, 2009.

[6] K. Chandy and W. Schulte, Event Processing: Designing IT Systems for Agile Companies, 1st ed. McGraw-Hill Osborne Media, 2009.

[7] D.C. Luckham, R. Schulte, "Event Processing Glossary – Version 1.1," July 2008. Available on: http://complexevents.com.

[8] W3C, OWL 2 Web Ontology Language, W3C, 2009. Available on: http://www.w3.org/TR/owl2-overview/.

[9] Cambridge Dictionary of British English, Cambridge Dictionaries Online. Available on: http://dictionary.cambridge.org/dictionary/british/.

[10] Z. Vendler, Facts and Laws, Doctoral disertation, Harvard University, 1959.

[11] J. Pustejovsky, "The syntax of event structure.," Cognition, vol. 41, no. 1-3, pp. 47-81, 1991.

 ISBN: 978-1-61208-184-7

A Top-Down-View on Intelligent Surveillance Systems

Yvonne Fischer* and Jürgen Beyerer*†
*Vision and Fusion Laboratory, Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
†Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Karlsruhe, Germany
Email: yvonne.fischer@kit.edu, juergen.beyerer@iosb.fraunhofer.de

***Abstract*—In today's surveillance systems, there is a need for enhancing the situation awareness of an operator. Supporting the situation assessment process can be done by extending the system with a module for automatic interpretation of the observed environment. In this article the information flow in an intelligent surveillance system is described and the separation of the real world and the world model, which is used for the representation of the real world in the system, is clarified. The focus of this article is on modeling situations of interest in a human-understandable way and how to infer them from sensor observations. For the representation in the system, concepts of objects, scenes, relations, and situations are introduced. Situations are modeled as nodes in a dynamic Bayesian network, in which the evidences are based on the content of the world model. Several methods for inferring situations of interest are suggested. Following this approach, even high-level situations of interest can be modeled by using different abstraction levels. Finally, an example of a situation of interest in the maritime domain is given.**

Keywords-surveillance system; data fusion; situation awareness; situation assessment; probabilistic reasoning.

I. Introduction

During the operation of complex systems that include human decision making, the processes of acquiring and interpreting information from the environment forms the basis for the state of knowledge of a decision maker. This mental state is often referred to as situation awareness [1], whereas the processes to achieve and maintain that state is referred to as situation assessment. In today's surveillance system, the situation assessment process is highly supported through various heterogeneous sensors and appropriate signal processing methods for extracting as much information as possible about the surveyed environment and its elements. Using these methods is, of course, an essential capability for every surveillance system in order to be able to observe a designated area and to detect and track objects inside this area. The approach of collecting as much sensor data as possible and extracting as much information as possible from it is termed bottom-up, or also data-driven processing.

However, this approach is not useful for the situation awareness of an operator, because his workload in interpreting all this information will be too high. The challenge of intelligent surveillance systems is therefore not only to collect as much sensor data as possible, but also to detect and assess complex situations that evolve over time as an automatic support to an operator's situation assessment process, and therefore enhancing his situation awareness. The approach of defining and presenting only relevant information about events and activities is termed top-down processing. However, there is a need for concepts and methrfid ods supporting higher level situation awareness, i.e., methods that are able to infer real situations from observed elements in the environment and to project their status in the near future.

The paper is structured as follows. In Section II, an overview of related work is given. As this article follows the top-down approach, the information flow in an intelligent surveillance system is highlighted in Section III. In Section IV, the methods of modeling situations of interest and inferring their existence are explained. In Section V, an example in the maritime domain is given.

II. Related Work

Working with heterogeneous sensors, the theories of multi-sensor data fusion [2] offer a powerful technique for supporting the situation assessment process. A lot of research has been done in combining object observations coming from different sensors [3], and also in the development of real-time methods for tracking moving objects [4]. Regarding data fusion in surveillance systems, the *object-oriented world model (OOWM)* is an approach to represent relevant information extracted from sensor signals, fused into a single comprehensive, dynamic model of the monitored area. It was developed in [5] and is a data fusion architecture based on the JDL (Joint Directors of Laboratories) data fusion process model [6]. Detailed description of the architecture and an example of an indoor surveillance application has been published in [7]. The OOWM has also been applied for wide area maritime surveillance [8].

First ideas of modeling situations in surveillance applications have been presented in our previous work in [9]. For the situation assessment process, probabilistic methods like hidden Markov models can be used, see for example [10]. In [11], Markov random fields are used to model contextual relationships and maximum a posteriori labeling is used to infer intentions of observed elements.

 ISBN: 978-1-61208-184-7

However, most of the methods used for situation assessment are based on machine learning algorithms and they result in models that humans are not able to understand. They are also strongly dependend on training data, which are not always available, especially not for critical situations. The contribution of this work is the modeling approach from a top-down perspective, which tries to model situations from a human perspective, i.e., what an operator wants to detect, and how to link them to methods for automatic interpretation.

III. Information Flow in Surveillance Systems

In surveillance applications, a spatio-temporal section of the real world, a so-called *world of interest*, is considered. The general information flow for intelligent surveillance systems is visualized in Figure 1, wherein information aggregates are represented by boxes, and processes are represented by circles. The information flow is as follows.

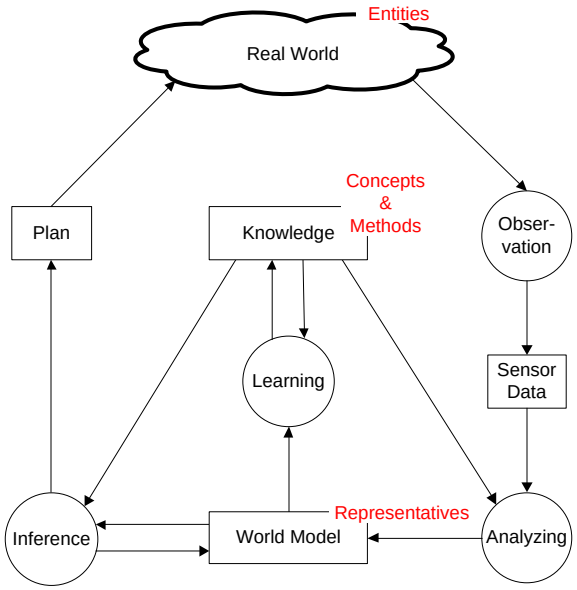


Figure 1. Information flow in a surveillance system represented by information aggregates *(boxes)* and processes *(circles)*.

First of all, all elements in the real world are termed *entities*. By the term entity, not only physical objects are meant, as entities can also be non-physical elements in the real world like relations or the name of a vessel. Thus, entities can represent observable or unobservable elements.

Sensor systems for observing the real world can be of extremely heterogeneous types, e.g., video cameras, infrared cameras, radar equipment, or radio-frequency identification (RFID) chips. Even human beings can act like a sensor by observing entities of the real world. Observing the world of interest with sensors results in sensor data, for example a radar image or a video stream. Sensor data is then analyzed by means of knowledge and the resulting information is transferred to the world model. Analyzing sensor data includes for example the detection and localization of moving vessels at sea from a video stream. Knowledge contains all information that is necessary for analyzing sensor data, for example specific signal-processing methods and algorithms used for the detection, localization and tracking of vessels in video streams.

The world model is a representation of entities in the world of interest and consists therefore of *representatives*. Every representative has a corresponding entity in the real world. The mapping between entities in the world of interest and representatives in the world model is structure-preserving and can therefore be interpreted as a homomorphism. Specific mappings are defined by *concepts* and are part of the knowledge. Concepts are for example used in the analyzing process by defining how an observed vessel is represented in the world model. As the world of interest is highly dynamic and changes over time, the history of the representatives is also stored in the world model. However, as mentioned before, some entities can't be observed directly. Therefore an inference process is reasoning about unobservable (and also unobserved) entities by means of knowledge. A simple inference process is for example to calculate an object's velocity from the last and current position. A more complex inference process would be to estimate if the intention of an observed vessel is benign or adversarial. Doing this way, the world model is always being updated and supplemented with new information by predefined inference processes.

Summing up, knowledge contains all information for analyzing sensor data, updating the world model and supplementing it with new information. Concepts are used for the representation of real-world entities in the world model. Characteristics of the knowledge are of course extremely dependent on the application domain. Additionally, knowledge is not static. The content of the world model can be used for acquiring new knowledge by a learning process, for example structure or parameter learning in graphical models.

To close the loop of the information flow, the result of an inference process could also include a plan of how to act further in the real world. This could be an action plan for an agent, for example to call the police, or a sensor management plan, for example a request for more detailed information from a special sensor.

IV. Modeling and Inferring Situations of Interest

Two problems are faced in this section: First, several concepts have to be defined, which means to define how the real-world entities can be represented in the world model. Second, the inference process has to be defined, which means to define how to reason about non-observable entities like situations or intentions from observed entities.

 ISBN: 978-1-61208-184-7

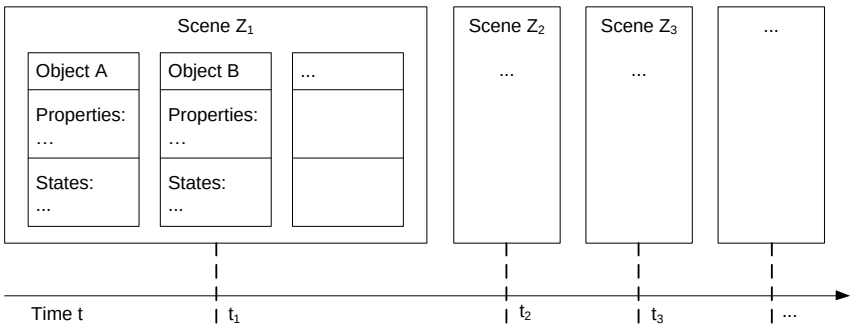


Figure 2. The concept of an object and a scene.

A. Concepts of world modeling

In this section, some basic concepts that can easily be used for the representation of real-world entities are defined. Addressed concepts here are objects, scenes, attributive relations, and situations. However, the world model can easily be extended by defining new concepts, e.g., for activities and events.

The concept of an *object* is defined as a physical entity of the real world. Regarding its spatial position, an object can be mobile, e.g., a vessel, or stationary, e.g., a land border. An object has several attributes, which can be divided into properties and states. Properties are time-invariant attributes, e.g., the length or the name of a vessel. State values can change over time and are therefore time-variant, e.g., the position or the velocity of a vessel. As the representation in the world model also has a memory, which means that the past states of an object are stored, the complete history of the observed object is always available. Furthermore, the representation of an object in the world model does not only include observed attributes, but also inferred ones. For example, based on observed positions of a vessel, the velocity can be inferred. Furthermore, attribute values can be quantitative or qualitative. For example, the absolute position and velocity of a vessel are quantitative attributes, and the attribute value that a vessel is made of wood is a qualitative one.

The concept of a *scene* is defined as the set of all observed and inferred object information at a point in time. A scene can therefore be interpreted as a time-slice, consisting of all objects and their attributes. To include the time aspect, a sequence of scenes can be defined, when the scenes are considered at several discrete points in time. However, a scene does not include any type of relations in an explicit way. This means, that it is for example not explicitly modeled that two vessels are close to each other. But implicitly, of course, this relation can be inferred by the positions of the two vessels. The concept of an object and a scene is visualized in Figure 2.

The *configuration space* is defined by all possibly occuring objects and their attributes. Thus, a scene, which is represented in the world model, can be identified by exactly one point in the configuration space. A sequence of scenes can be interpreted as a trajectory through the configuration space defined by a series of points in time.

The concept of *attributive relations* is defined as a statement about dependencies between at least two different attribute values of one or more objects. Similar to the attribute values of an object, relational values can be quantitative, e.g., the distance of two objects, or they can be qualitative, e.g., two objects are close to each other. Mostly, relational values are inferred, but some can also be observed, e.g., a measured distance by a laser. A relation can also exist between representatives of the same object in different scenes, e.g., the distance an object has covered between the two scenes.

The concept of a *group* is defined as set of object representatives that have the same values for a specific attribute. It is therefore a special case of an attributive relation and can also be interpreted as an equivalence-relation on a specific attribute value. Examples for groups are vessels that have the same size or vessels that are all in a certain area.

The concept of a *situation* is defined as a statement about a subset of the configuration space, which is either true or false. A specific situation of interest exists, if its statement was inferred to be true. Situations are therefore characterized by qualitative attribute values and their truth is inferred based on information in the world model. This means that situations have a higher level of abstraction and the level of detail included in the quantitative attribute values of objects and relations is getting lost. The simplest situation is a statement about qualitative attribute value of an object, e.g., that a vessel is made of wood. There are also situations, which can only be inferred by observing the real world over a period of time, e.g., the situation that a vessel is taking a straight course.

But although situations are also characterized by information collected over a time-period, they only exist at a special point in time. Their existence in the next time-point has to be verified again. However, there are a lot of dependencies between different situations. First of all, situations can be

 ISBN: 978-1-61208-184-7

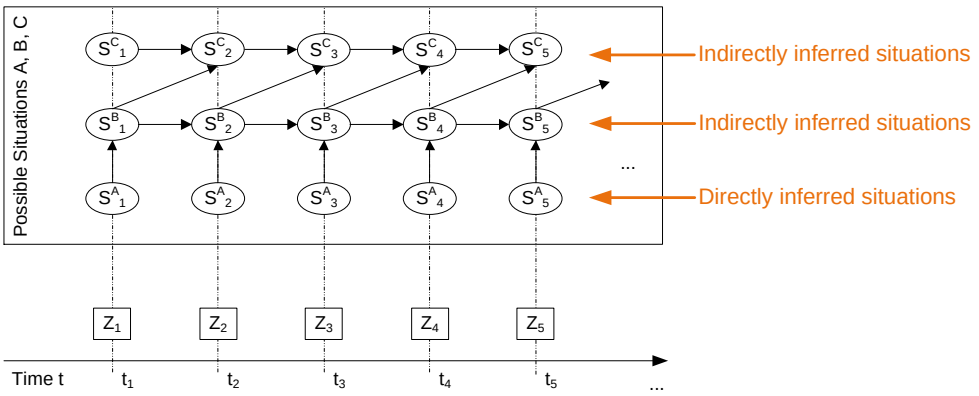


Figure 3. A network of situations, divided into directly and indirectly inferred situations.

inferred from other situations, e.g., if a vessel is heading in a certain direction and has a lot of peaple on board, the inferred situation could be that the vessel is carrying refugees on board. Furthermore, several situations can exist in parallel or the existence of one situation can exclude the existence of another situation. Mathematically, a situation at a time t can be modeled as a binary random variable S_t, such that

$$S_t(\omega) = \begin{cases} 1 & \text{if } \omega \text{ is true,} \\ 0 & \text{if } \omega \text{ is false,} \end{cases} \quad (1)$$

and ω is the statement of the situation of interest. Then, we are interested in the probability, that ω is true, and thus that the situation S_t exists at time t. We write this existence probability as $P(S_t = 1)$, or $P(S_t)$ in short.

For calculating this probability, the aforementioned dependencies between other situations have to be modeled. The following two cases can be distinguished:

- Directly inferred situations: the existence probability $P(S_t)$ can be inferred directly from the information content of a scene (or other concepts like relations or groups)
- Indirectly inferred situations: the existence probability $P(S_t)$ depends on the existence probability of other situations.

This also includes, that the existence probability of an indirectly inferred situation in future can for example be supported by the earlier existence of the situation itself, and the existence probability of a directly inferred situation cannot be supported over time. This concept of a network of situations is visualized in Figure 3.

B. Inferring Situations of Interest

Due to this modeling, the network of situations can be interpreted as a probabilistic graphical model, namely a Dynamic Bayesian network (DBN). In a simple Bayesian network, the basic idea is to decompose the joint probability of various random variables into a factorized form. Random variables are depicted as nodes and conditional probabilities as directed edges. The joint probability can then be factorized as

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)), \quad (2)$$

where $Pa(X_i)$ is the set of parents of the node X_i. If $Pa(X_i)$ is an empty set, then X_i is a root node and $P(X_i|Pa(X_i)) = P(X_i)$ denotes its prior probability.

A DBN [4] is defined as a pair $(B_0, 2TBN)$, where

- B_0 defines the prior distribution $P(\boldsymbol{X}_0)$ over the set $\boldsymbol{X}_0$ of random variables, and
- $2TBN$ defines a Bayesian network over two time slices with

$$P(\boldsymbol{X}_t | \boldsymbol{X}_{t-1}) = \prod_{i=1}^{n} P(X_t^i | Pa(X_t^i)), \quad (3)$$

where X_t^i is a node at time slice t and $Pa(X_t^i)$ is the set of parent nodes, which can be in the time slice t or in the time slice $t-1$.

Note that in the definition of a $2TBN$, $Pa(X_t^i)$ is never empty, i.e., every node in time slice t has at least one parent node and therefore the left side of equation (2) differs from the left side of equation (3). An example of a $2TBN$ with 3 nodes in each time slice is shown in Figure 4. The joint probability distribution of a DBN can then be formulated as

$$P(\boldsymbol{X}_{0:T}) = P(\boldsymbol{X}_0) \cdot \prod_{t=1}^{T} \prod_{i=1}^{n} P(X_t^i | Pa(X_t^i)). \quad (4)$$

As we want to model a network of situations by a DBN, the structure of the network has to fulfill the following assumptions:

- Stationarity: the dependencies within a time slice t and the dependencies between the time slices $t-1$ and t do not depend on t.
- 1st order Markov assumption: the parents of a node are in the same time slice or in the previous time slice.

 ISBN: 978-1-61208-184-7

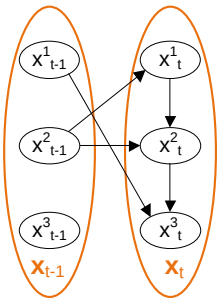


Figure 4. A example of a $2TBN$ defining dependencies between two time slices and dependencies between nodes in time slice t. Note that a $2TBN$ does not define the dependencies between nodes in time slice $t-1$.

- Temporal evolution: dependencies between two time slices are only allowed forward in time, i.e., from past to future.
- Time slice structure: The structure of one time slice is a simple Bayesian network, i.e., without cycles.

For modeling the situational network, the set of situations are divided into the set of directly inferable situations E and the set of indirectly inferable situations S, as described above. The state transition between two time slices satisfies the Markov assumption

$$P(S_t|S_{0:t-1}) = P(S_t|S_{t-1}), \quad (5)$$

and the dependencies between the directly and indirectly inferred situations is defined as

$$P(E_t|S_{0:t}, E_{0:t-1}) = P(E_t|S_t). \quad (6)$$

Due to this dependency, it is assumed that the values of the directly inferred situations are only dependent on the values of the indirectly inferred situations. The joint probability can then be calculated recursively by

$$P(S_{0:T}, E_{1:T}) = P(S_0) \cdot \prod_{t=1}^{T} P(S_t|S_{t-1})P(E_t|S_t). \quad (7)$$

By modeling the network of situations in this way, the following inference calculations are possible:

- Filtering: $P(S_t|E_{1:t})$ gives a solution to the existence probability of a set of situations S at the current time,
- Prediction: $P(S_{t+k}|E_{1:t})$ (with $k > 0$) gives a solution to the existence probability of a set of situations S in the (near) future,
- Smoothing: $P(S_k|E_{1:t})$ (with $0 < k < t$) gives a solution to the existence probability of a set of situations S in the past,
- Most likely explanation: $\text{argmax}_{S_{1:t}} P(S_{1:t}|E_{1:t})$ gives a solution to the most likely sequence of situations $S_{1:t}$.

Due to this modeling, the existence probability of a set of indirectly inferable situations can be calculated in a recursive way at each point in time. A situation is represented in the world model, if the corresponding existence probability is larger than an instantiation-threshold. If the existence probability in the next time step is below a deletion-threshold, it is assumed that the situation doesn't exist any longer and its representation is removed from the world model. This way, it is tried to keep an up-to-date representation of the existing situations of the real world.

V. Application Scenario in the Maritime Domain

For a representation of the world model, the OOWM-system as described in [8] was adapted to the maritime domain. The graphical user interface of the OOWM is depicted in Figure 5. It shows observed vessels at the Mediterranean Sea between the African coast and the island of Lampedusa. Sensor observations are simulated in the system, but they are assumed to be generated by coastal radar systems or signals from the automatic identification system (AIS). In Figure 5, an observed vessel is selected and its observed attributes can be seen on the left side of the user interface. These are exactly the attributes that are stored in the world model and are used for inferring situations of interest.

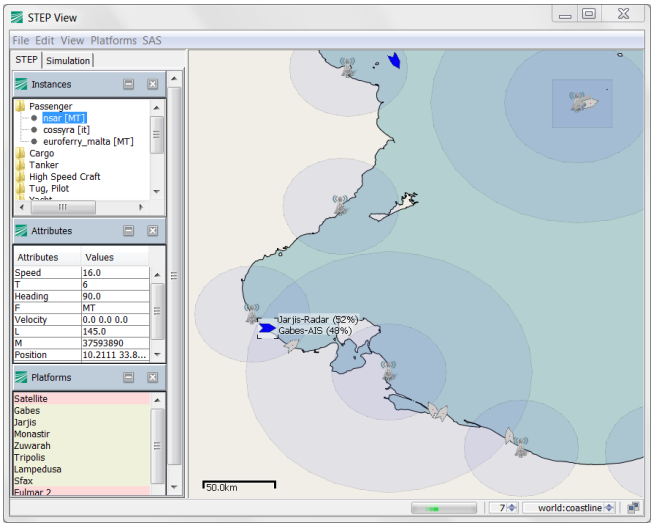


Figure 5. The OOWM system applied to the maritime domain

In the Mediterranian Sea, a situation of interest is the detection of vessels that carry refugees on board. Based on various statements by maritime experts, these vessels have the following (observable) characteristics: They start from the African coast (Tunisia or Libya), are heading towards Lampedusa, take a direct course, and don't send any AIS-Signal for identification. They are either wooden boats or motor-boats, where the wooden boats are slower and smaller than the motor-boats, and the motorboats often go the border, put the refugees into the water and make an emergency call.

An example of a dynamic Bayesian network representing the 3 situations of interest that an observed vessel is a refugee vessel, a wooden vessel, or a motor-vessel is shown

 ISBN: 978-1-61208-184-7

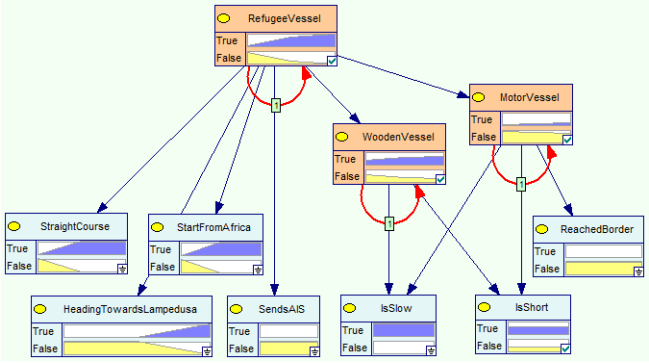


Figure 6. Dynamic Bayesian Network with 3 situations of interest (colored in orange). Temporal arcs over one time slice are marked with a "1" and colored in red.

in Figure 6. The 3 temporal arcs are pointing to the situations of interest themselves, respectively. The resulting existence probabilities (calculated by filtering) for the root node situation (refugee vessel) over 3 time steps are visualized in Figure 7. It can clearly be seen that due to the evidence that has been collected over time, the existence probability of this situation is increasing over time.

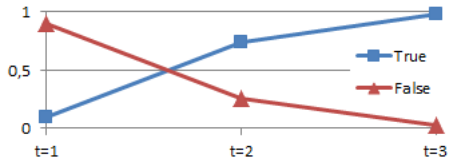


Figure 7. Resulting existence probabilities for the situation that an observed vessel is carrying refugees on board.

The challenges of designing the situational network are to model the structure and to determine the parameters, i.e., the conditional probabilities. Finally, the resulting probabilities for different configurations have to be interpreted (e.g., for the specification of the instantiation- and the deletion-threshold), which is often not straightforward.

VI. Conclusion and Future Work

In this article the information flow in an intelligent surveillance system was highlighted and it was described how situations of interest in surveillance applications can be modeled by concepts. For modeling a network of situations, the framework of dynamic Bayesian networks is suggested, in which the values of the directly inferable nodes are based on the content of the world model. This modeling fulfills the requirements resulting from the definition of situations and allows the application of efficient inference methods. An example of a situation of interest in the maritime domain was given. By extending the surveillance system with such a module for automatic interpretation of the observed environment it is able to support the situation assessment process of an operator and thus enhances his situation awareness.

Future work includes an experimental evaluation of the proposed method and an investigation on supporting the human operator in designing a situational network without having a detailed knowledge of the underlying method. Also the real-time capability of the proposed method when using a large amount of data has to be investigated.

Acknowledgment

This research is partially supported by the EU-FP7-Project WIMA^{2}S (Wide Maritime Area Airborne Surveillance), see http://www.wimaas.eu (last access: 11.12.2011).

References

[1] M. R. Endsley, "Towards a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, no. 11, pp. 32–64, 1995.

[2] D. L. Hall and S. A. H. McMullen, *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Inc., 2004.

[3] M. Baum, I. Gheta, A. Belkin, J. Beyerer, and U. D. Hanebeck, "Data association in a world model for autonomous systems," in *Proc. of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2010)*, 2010, pp. 187–192.

[4] A. Dore, M. Soto, and C. S. Regazzoni, "Bayesian tracking for video analytics: An overview," *IEEE Signal processing magazine*, vol. 27, no. 5, pp. 46–55, 2010.

[5] A. Bauer, T. Emter, H. Vagts, and J. Beyerer, "Object oriented world model for surveillance systems," in *Future Security: 4th Security Research Conference*. Fraunhofer Press, 2009, pp. 339–345.

[6] A. N. Steinberg, C. L. Bowman, and F. E. White, "Revisions to the JDL data fusion model," in *Sensor Fusion: Architectures, Algorithms, and Applications, Proceedings of the SPIE Vol. 3719*, 1999, pp. 430–441.

[7] J. Moßgraber, F. Reinert, and H. Vagts, "An architecture for a task-oriented surveillance system: A service- and event-based approach," in *Fifth International Conference on Systems (ICONS 2010)*, 2010, pp. 146–151.

[8] Y. Fischer and A. Bauer, "Object-oriented sensor data fusion for wide maritime surveillance," in *2nd International Conference on Waterside Security,IEEE,(WSS 2010)*, 2010, pp. 1–6.

[9] Y. Fischer, A. Bauer, and J. Beyerer, "A conceptual framework for automatic situation assessment," in *IEEE First International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA 2011)*, 2011, pp. 234–239.

[10] D. Meyer-Delius, C. Plageman, and W. Burgard, "Probabilistic situation recognition for vehicular traffic scenarios," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 459–464.

[11] R. Glinton, J. Giampapa, and K. Sycara, "A markov random field model of context for high-level information fusion," in *9th International Conference on Information Fusion*, 2006, pp. 1–8.

 ISBN: 978-1-61208-184-7

Designing a Fault-Tolerant Satellite System in SystemC

Kashif Javed
Department of Information Technologies
Abo Akademi University
Turku, FIN-20520, Finland
Kashif.Javed@abo.fi

Elena Troubitsyna
Department of Information Technologies
Abo Akademi University
Turku, FIN-20520, Finland
Elena.Troubitsyna@abo.fi

***Abstract*—Designing fault-tolerant satellite systems is a challenging engineering task. Often behavior of satellite systems is structured using notion of modes. Ensuring correctness of mode transitions is vital for guaranteeing safe and fault-tolerant functioning of a satellite. In this paper, we propose an approach to designing fault-tolerant satellite systems in SystemC. We demonstrate how to develop Attitude and Orbit Control System in SystemC and verify its correctness via model checking.**

Keywords-component; Fault-Tolerance; Mode-Rich Systems; Design; Verification

I. INTRODUCTION

Designing a system controlling a spacecraft is a challenging engineering task. The system should satisfy a large number of diverse functional and non-functional requirements. In particular, the designers should aim at building a fault-tolerant system, i.e., the system that should cope with faults of various system components. Often behavior of satellite systems is structured using the notion of modes – mutually exclusive sets of system behavior. Fault-tolerance is achieved by putting the system to some downgraded mode when an error occurs. In this paper, we consider an Attitude and Orbit Control System (AOCS) – a generic subsystem of a spacecraft [1]. We demonstrate how to achieve fault-tolerance via backward mode transitions.

AOCS is a complex control system consisting of several components. To ensure correctness of mode transition, we need to guarantee that all components reach a certain state. Moreover, when a component fails we need to guarantee that all other components make an appropriate backward transition.

In this paper, we propose an approach for designing more-rich system in SystemC programming language. We propose an algorithm defining mode-transition scheme of AOCS. To confirm correctness of our algorithm, we have converted it into Promela [6,7] and the results have been verified using SPIN model checker [7,8].

Section II presents architecture of the system. Unit branch state and state transitions have been explained in Section III and the controller phases & phase transitions of the AOCS are described in Section IV. Mode transitions and fault-tolerance procedures for correct functioning of the satellite under faulty conditions are illustrated in Sections V and VI respectively. Section VII explains verification of the implemented system and the paper is summarized in Section VIII besides giving direction for the future work.

II. ARCHITECTURE

The main purpose of AOCS is to control attitude and orbit [1] of a satellite. AOCS consists of a number of components -- AOCS Manager, FDIR (Failure Detection, Isolation and Recovery) Manager, Mode Manager and Unit Manager. The AOCS manager plays key role while dealing with the processing of sensor data, managing actuator movements relating to the units of Reaction Wheel (RW) and Thruster (THR) and doing computation for various controls. The responsibility of FDIR is to timely deal with such tasks as failure detection, isolation and recovery. Mode transitions are handled by the Mode Manager whereas the Unit Manager deals with unit reconfigurations and unit level state transitions [2,3]. Mode and Unit Manager Architectures are further elaborated in the following paragraphs.

A. Mode Manager

The responsibilities of mode include checking of mode transition preconditions, execution of mode transitions, management of controller phases and partially management of related units. There are six different types of controlled modes (i.e. Off, Standby, Safe, Nominal, Preparation and Science) in the mode manager and each mode has its own well-defined unique function. A brief summary of these modes is given below:

1) Off Mode: The satellite is immediately switched in the off mode as soon as the AOCS software booting is completed from the central data management unit.

2) Standby Mode: It is important to check and ensure successful separation of the spacecraft from the launcher and this work is continuously monitored and completed by the software process during the standby mode.

3) Safe Mode: Satellite enters this mode when the separation from the launcher is done. As soon as the system is in the safe mode, the relevant portions of Earth Sensor (ES), RW (Reaction Wheel) and Sun Sensor (SS) are switched to on state, the coarse pointing controller goes in the running phase and fine pointing controller is put in the idle phase. Initially the satellite acquires a stable attitude and then it achieves the coarse pointing.

 ISBN: 978-1-61208-184-7

4) Nominal Mode: When a mode transitions to nominal, the coarse pointing controller becomes idle and the fine pointing controller is set to the running phase. The selected branches of RW, Star Tracker (STR) and THR are switched to on state. In this mode, the satellite utilizes fine pointing control so that the Payload Instrument (PLI) in the AOCS is properly used for measurements.

5) Preparation Mode: The moment the mode is transitioned to the preparation, the concerned portion of Global Positioning System (GPS) is set to fine state, the relevant branch of PLI is switched to standby state and needed processes of RW, STR and THR go to on state. Thus, this mode ensures that the fine pointing control is reached and PLI gets ready for fulfilling its required tasks.

6) Science Mode: In science mode, the selected branch of GPS remains in the fine state, the concerned branch of PLI goes in the science state and the relevant parts of RW, STR and THR maintain their on state. Therefore, the PLI in this mode is ready to perform the tasks for which it has been designed. It stays in this mode till the completion of planned tasks.

B. Unit Manager

The AOCS consists of seven different units and internal state changes in these units are controlled by the unit manager. Mode manager controls the components of unit manager. Seven different controlled units are ES, SS, STR, GPS, RW, THR and PLI. Their brief description is as under:

1) ES is a device that measures the direction to the earth in the sensor's field of view. ES's internal state is either on and off.

2) SS is a tool to measure the direction to the sun in the sensor's field of view. It is also in the on or off state.

3) STR is an optical device that measures the position of stars in its field of view and performs pattern recognition on these stars in order to identify the portion of the sky at which it is looking. Two possible STR's operational states are on and off.

4) GPS is a sophisticated gadget that receives readings related to the satellite position and makes calculations to determine satellite's attitude. Two possible states of GPS operation are coarse navigation and fine navigation.

5) RW is a rotating wheel which is essentially required in order to apply the required torque to the satellite. It is achieved by accelerating or breaking the wheel. RW's state can be either on or off.

6) THR is a position actuator that is used to force the satellite to change its position and its orbit by emitting gas. It can also be in either on or off state.

7) The PLI is an instrument which provides required measurements pertaining to the specific mission. It can operate in standby or science state.

III. UNIT BRANCH STATE AND STATE TRANSITIONS

Every unit is implemented as a pair of identical devices to maintain the nominal branch and the redundant branch. For each unit, one and only one branch is selected at a time. Every selected branch is in on state and its status is locked. In other words, a branch in the off state is always allocated an unlocked status.

In total, there are six states of unit components (i.e. on, off, coarse, fine, standby and science). Whenever an unit state goes from off to on, the powering takes place. Similarly, when the unit switches from on to off state, un-powering takes place. Powering and un-Powering are associated with the states and state transitions of a branch of ES, SS, STR, RW or THR. Occurrence of such states and state transitions is shown in Figure 1. For the GPS unit, unit state goes from off to coarse state and coarse to fine state, then powering and upgrading is carried out respectively. In case of fine to off state transition, first downgrading is performed then un-powering is done. States and State Transitions of a Branch of GPS are depicted in Figure 2.

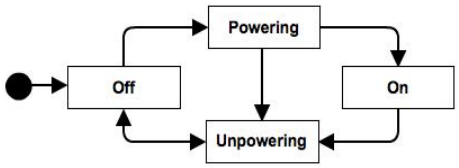


Figure 1: States and State Transitions of a Branch of ES, SS, RW, STR or THR [1]

In case of PLI unit, when the unit state goes from off to standby and from standby to science state, then powering and upgrading is achieved respectively. In case of science to off state transition, first downgrading occurs and then un-powering takes place. Figure 3 demonstrates states and their transitions of a branch of PLI.

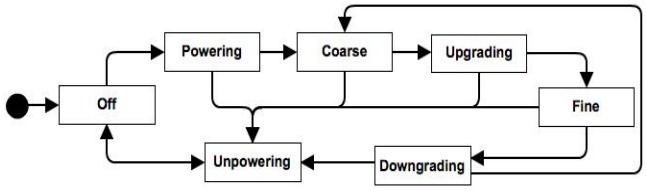


Figure 2:States and State Transitions of a Branch of GPS [1]

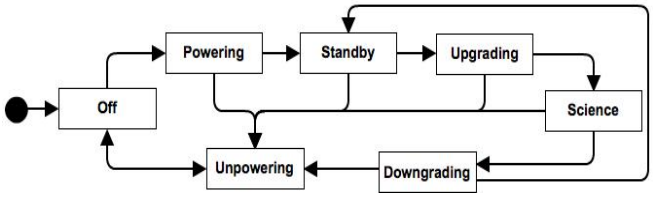


Figure 3: States and State Transitions of a Branch of PLI [1]

State transitions are very fast to accommodate time constrains for real-time satellite operations. Hence, any state transition to powering, un-powering, upgrading or downgrading takes less than one AOCS cycle. However, every state transition to off takes minimum three and maximum four AOCS cycles. Any state transition to on, coarse, fine, standby or science has a success condition if the transition gets completed during the first AOCS cycle when the condition is observed to hold. However, any state transition to on, coarse, fine, standby or science is overridden if the associated success condition is not observed to hold within a predefined number of AOCS cycles from start of the transition.

IV. CONTROLLER PHASES AND PHASE TRANSITIONS

The AOCS has two controllers -- Coarse Pointing Controller (CPC) and Fine Pointing Controller (FPC). The main objective of these two controllers is to direct the line of sight with a specified coarse accuracy and fine accuracy respectively. It is an essential requirement and must be met within given time limits. The following rules have to be observed during the controller phase transitions when a certain operational mode is reached:

1) Both controllers go to idle phase when the mode transition is set to off or standby state.

2) When the mode transition is switched to safe state, the CPC enters the running phase and the FPC remains in the idle phase.

3) When the mode transition shifts to nominal, preparation or science, the CPC goes in the idle phase and the FPC moves in the running phase.

Only one controller can be in non-idle phase at any point of time. When a controller phase has to switch from idle to running, first of all it is set to preparing. After predefined number of AOCS cycles, the controller is set to ready phase. Finally, the phase of controller is shifted to running as indicated in Figure 4. It can also be noticed that the controller can directly move to the idle phase from any of the other three phases (preparing, ready and running).

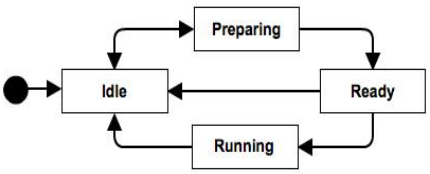


Figure 4: Phases and Phase Transitions of a Controller [1]

V. MODE TRANSITIONS

The following rules are imposed on mode transitions in order to ensure correct satellite function in nominal (fault-free) and faulty conditions:

1) When a mode transition to off or standby is completed, it is ensured that every branch in every unit is put in the off state.

2) On reaching to the safe mode, the selected branches of ES, RW and SS are set in the on state and all other branches pertaining to different units go to the off state.

3) In case of a transition to the nominal mode, the selected branch of GPS is turned in the coarse state, the concerned branches of RW, STR and THR are set to on state, and remaining every branch in every unit is put in the off state.

4) Completion of a mode transition to preparation ensures that the relevant branch of GPS is in the fine state, the chosen branch of PLI is in the standby state, the selected parts of RW, STR and THR are in the on state, and rest every branch in every unit is in the off state.

5) A mode transition to science requires that the needed branch of GPS is in the fine state, the selected branch of PLI is in the science state, the concerned branches of RW, STR and THR are in the on state, and all other branches pertaining to different units remain in the off state.

VI. FAULT TOLERANCE

Fault-tolerance should guarantee that the system continues to operate in predictable way even in case of failure of any of its components. Recovery from errors in fault-tolerant systems can be characterized as either roll forward or roll back. Forward error recovery aims at bringing the system to a new error-free state. Backward error recovery rolls back the system to some previous state before an error occurrence. In mode-rich systems, the backward error recovery is achieved via backward mode transition, i.e., mode downgrading. The mode down-gradation depends on various errors, which are explained below:

A. Branch State Transition Errors

A branch state transition error means that when some unit transitions to on state, the mode coarse, fine, standby or science gets overridden due to timeout condition. Because operation and state transition delays have to be avoided, we should time each mode transition. If a step of transition is not completed within a specified time limit, timeout signal is generated to get into a safe condition. The important error checks concerning to the branch state transitions are:

1) A branch state transition error on the redundant branch of ES, RW or SS causes a mode transition to off.

2) A mode transition to safe takes place when there is a branch state transition error on the redundant branch of GPS, STR or THR and there is no branch state transition error on the redundant branches of ES, RW and SS.

3) When a branch state transition error on the redundant branch of PLI occurs, it results into a mode transition to

 ISBN: 978-1-61208-184-7

nominal provided that there is no branch state transition error on the redundant branches of ES, SS, GPS, RW, STR and THR.

B. Phase Transition Errors

A phase transition error or an attitude error may arise during the computations done by the selected controller. An attitude error is generated when there is a problem in the execution of an AOCS algorithm. It means that an error occurs only when one of the two controllers (i.e. CPC and FPC) is in the running phase. The key factors relating to the attitude errors are:

1) If the current mode is safe, then a non-ignored attitude error causes a transition to the off mode.

2) In case the existing mode is nominal and a non-ignored attitude error occurs, a mode transition to safe takes place.

3) A mode transition to nominal takes place when the current mode is preparation and a non-ignored attitude error is generated.

4) The generation of a non-ignored attitude error moves the mode transition to preparation with the condition that the existing mode is science.

C. Unit Reconfiguration

Each logical unit consists of two hardware units known as nominal and redundant. Initially, the nominal unit works in the active role and provides all the necessary support for normal operation of the system. The redundant unit serves as a backup resource. When an error is detected in the nominal unit, it becomes “reconfigured”. It means that the nominal unit is switched off and the redundant unit takes over the operational tasks.

The important errors that take place during the unit reconfiguration are:

1) A branch state transition error on the nominal branch of ES, SS or RW causes a reconfiguration of the unit if there is no branch state transition error on the redundant branches of ES, SS and RW.

2) A branch state transition error on the nominal branch of GPS, STR, THR or PLI causes a reconfiguration of the unit if there is no branch state transition error on the redundant branches of ES, SS, GPS, RW, STR and THR.

Figure 6 shows detailed flow chart of the implemented system.

VII. VERIFICATION

We have implemented mode-transition algorithm in SystemC language. The SystemC Verification Standard provides API for transaction based verification, constrained and weighted randomization, exception handling, and other verification tasks [4,5]. SystemC supports the use of special data types which are often used by the hardware engineers. It comes with a strong simulation kernel to enable the designers to write good test benches for easy and speedy simulation. It is extremely important because the functional verification at the system level saves a lot of money and time.

The system architecture that is implemented in SystemC is verified in the SPIN model checker. SPIN [6,7,8] is often used to verify behavior of distributed and parallel systems. PROMELA (PROcess MEta LAnguage) is a high level language which is widely used to specify systems descriptions and is fully supported by SPIN for the purpose of verification of software-based applications. SPIN PROMELA is used to carry out detailed testing and verification of design and architecture of various systems.

The simplified system architecture for AOCS is shown in Figure 5.

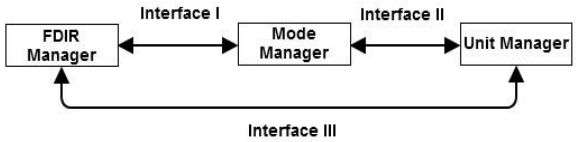


Figure 5: System Architecture [1]

An example of an interfaces between the FDIR Manager, Mode Manager and Unit Manager shown in Figure 5 are given below.

When failure occurs in the system, FDIR detects the error and issues the requests of mode transition, and then Mode Manager is responsible for mode transitions to the downgraded mode on the basis of error type. The following part of the code represents the Interface I scenario for Science Mode.

```
if (Mode==F) // Mode F: Science Mode
{       if (ES==off && SS==off && GPS==fine && STR==on &&
        RW==on && THR==on && PLI==science && CPC==idle
        && FPC==run)
        {/* The associated code describes that the conditions are valid
        for Science Mode. The current mode is Science. */}
        else if ((ES!=off || SS!=off || RW!=on) && STR==on &&
        GPS==fine && THR==on && PLI==science && CPC==idle
        && FPC==run)
        {/* The associated code describes that the conditions are not
        valid for Science Mode as error occurs on the unit branch of ES,
        SS or RW. It causes the mode transition to Off Mode. */}
        else if ((GPS!=fine || STR!=on || THR!=on) && ES==off &&
        SS==off && RW==on && PLI==science && CPC==idle &&
        FPC==run)
        {/* The associated code describes that the conditions are not
        valid for Science Mode as error occurs on the unit branch of
        GPS, STR or THR. It causes the mode transition to Safe Mode.
        */}
        else if (ES==off && SS==off && GPS==fine && STR==on
        && RW==on && THR==on && PLI!=science && CPC==idle
        && FPC==run)
        {/* The associated code describes that the conditions are not
        valid for Science Mode as error occurs on the unit branch of
        PLI. It causes the mode transition to Nominal Mode. */}
        else if (ES==off && SS==off && GPS==fine && STR==on
        && RW==on && THR==on && PLI==science &&
        (CPC!=idle || FPC!=run))
        {/* The associated code describes that the conditions are not
        valid for Science Mode as error occurs in the phase of Coarse or
```

Fine Pointing Controller. It causes the mode transition to Preparation Mode. */}
else
{/* The associated code describes that no transitions take place. */ } }
else
{/* The associated code describes that it is an invalid mode. Program is terminated.*/}

The SPIN's verification model successfully checks all the global mode transitions and the fault-tolerance of the system architecture. We have successfully verified forward and backward mode transitions and ensured correctness of global mode transitions with respect to component states.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an approach to designing fault tolerant mode-rich control systems. Our work aimed at demonstrating how to design satellite control system in SystemC and verify correctness using model checking. Our approach has been demonstrated by the design of Attitude and Orbit Control System – a generic subsystem of spacecrafts.

The proposed system has been implemented in SystemC language as it is being used as a defacto verification standard in embedded systems. SystemC specification was easily aligned with Promela which works as the input language to SPIN for model checking and verification.

We have presented the design of the system and verification steps pertaining to unit branch transition errors, controller phase transition errors and unit reconfiguration.

Our work complements research done on formal modeling of mode-rich satellite systems. The formal modeling undertaken in [9,10] aimed at enabling proof-based verification of mode-rich systems modeled in Event-B. In [11] the authors perform failure modes and effect analysis of each particular mode transition to systematically design mode transition scheme. Our work aims at building a gap between formal specification and code. This motivated our choice of SystemC as a design language and model-checking based verification.

As a future work, we are planning to investigate design and verification of decentralized mode-rich systems. In particular, we will study how to ensure correctness of mode transitions as a result of negotiation between several mode managers.

REFERENCES

[1] "DEPLOY Work Package 3 - Attitude and Orbit Control System Software Requirements Document", Space Systems Finland, Ltd., December 2010.

[2] M. Heimdahl and N. Leveson, "Completeness and Consistency in Hierarchical State-Based Requirements", IEEE Transactions on Software Engineering, Vol.22, No. 6, June 1996, pp. 363-377.

[3] N. Leveson, L. D. Pinnel, S. D. Sandys, S. Koga, and J. D. Reese, "Analyzing Software Specifications for Mode Confusion Potential", Proceedings of Workshop on Human Error and System Development, C.W. Johnson, Editor, March 1997, Glasgow, Scotland, pp. 132-146.

[4] C. Ip and S. Swan, "A tutorial introduction on the new SystemC verification standard", Technical report, www.systemc.org, 2003.

[5] L. Singh and L. Drucker, "Advanced Verification Techniques : A SystemC Based Approach for Successful Tapeout", Springer, 2004.

[6] J. Katoen, "Concepts, Algorithms and Tools for Model Checking", Lecture Notes, Chapter 1: System Validation, 1999.

[7] N. A. S. A. Larc, "What is Formal Methods?", NASA Langley Methods, http://shemesh.larc.nasa.gov/fm/fm-what.html, formal methods program, 2001.

[8] Kashif Javed, Asifa Kashif, and Elena Troubitsyna, "Implementation of SPIN Model Checker for Formal Verification of Distance Vector Routing Protocol", International Journal of Computer Science and Information Security (IJCSIS), Vol 8, No 3, June 2010, USA, ISSN 1947-5500, pp. 1-6.

[9] Alexei Iliasov, Elena Troubitsyna, Linas Laibinis, Alexander Romanovsky, Kimmo Varpaaniemi, Dubravka Ilic, and Timo Latvala. Developing Mode-Rich Satellite Software by Refinement in Event B . In Proceedings of FMICS 2010, the 15th International Workshop on Formal Methods for Industrial Critical Systems, September 2010, LNCS 6371. Springer.

[10] Alexei Iliasov, Elena Troubitsyna, Linas Laibinis, Alexander Romanovsky, and Kimmo Varpaaniemi, Pauli Väisänen. Verifying Mode Consistency for On-Board Satellite Software, 2010, LNCS 6351, Computer Safety, Reliability, and Security, Pages 126-141, Springer.

[11] Yuliya Prokhorova, Elena Troubitsyna, Linas Laibinis, Kimmo Varpaaniemi, and Timo Latvala. Derivation and Formal Verification of a Mode Logic for Layered Control Systems. Asia-Pacific Software Engineering Conference. IEEE Computer, December 2011.

 ISBN: 978-1-61208-184-7

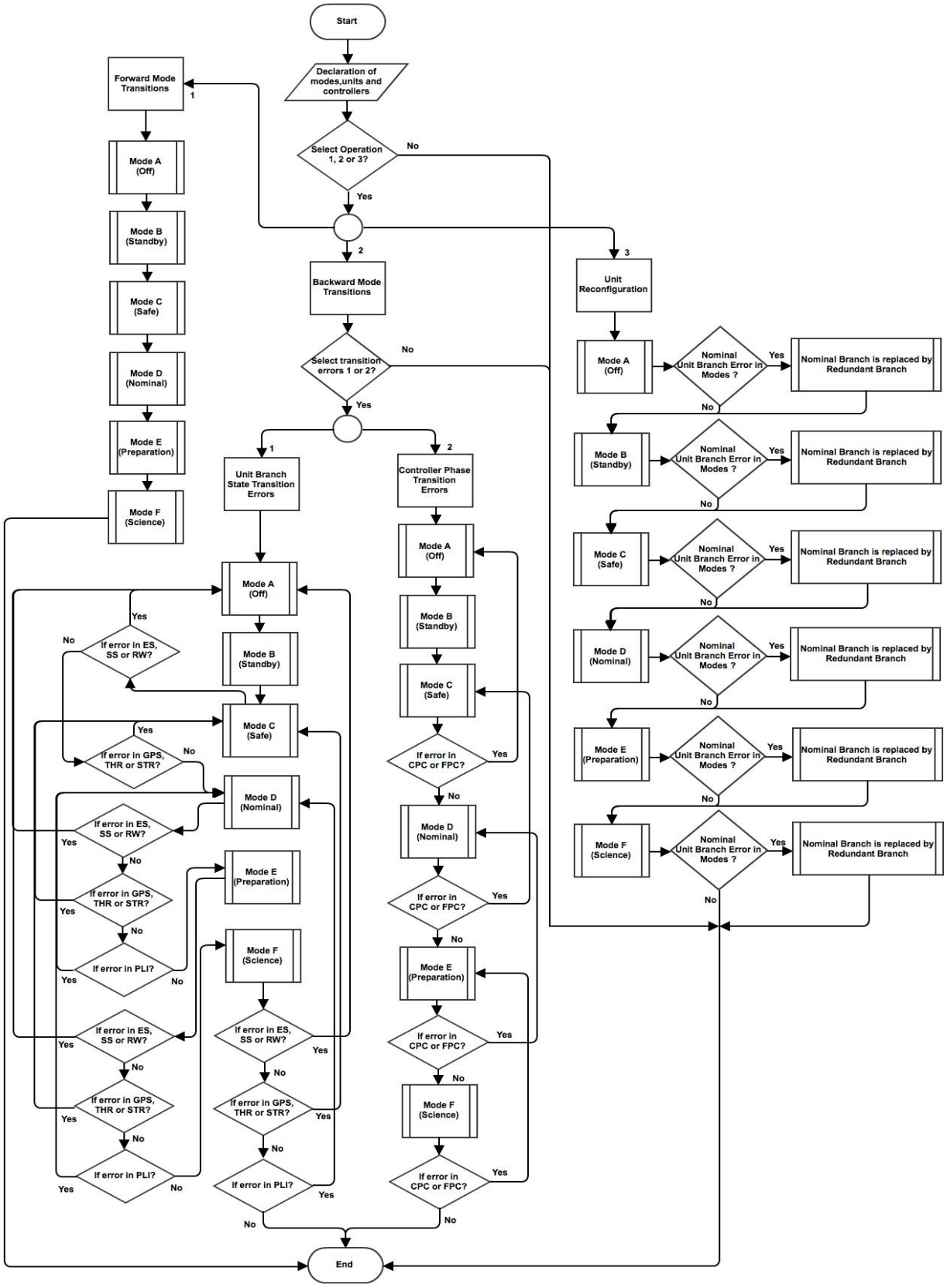


Figure 6: System Flow Chart

How About Agile Systems Development?

Hermann Kaindl, Edin Arnautovic, and Jürgen Falb
Institute of Computer Technology
Vienna University of Technology
Vienna, Austria
{kaindl, arnautovic, falb}@ict.tuwien.ac.at

***Abstract*—In recent years, a hype about "agile" *software* development has been growing. So, there may be some temptation to simply transfer such approaches to general systems development, for system including *hardware*. It is important to understand, however, that a core idea behind agile software approaches is *iterative and incremental development* (IID), actually an old and proven idea. Unfortunately, increments especially in rapid iterations face inherent limitations in hardware development. So, we claim that agile development for general systems involving hardware is much more difficult to achieve than what current folklore may assume. In order to address this issue, we constructively propose a development life cycle for general systems that takes these limitations into account. It distinguishes between iterations with major, minor and with almost no increments, in order to include hardware development realistically in IID. In this way, some of the promises of agile approaches may be kept in general systems development, whether it is called "agile" or not.**

***Keywords*-Agile development; iterative and incremental development; development life cycle**

I. INTRODUCTION

It is important to distinguish between *Agile Systems* engineering and agile *Systems Engineering* [1]. The former deals with *systems* that are flexible, reconfigurable, extensible, scalable, etc. The latter focuses on flexibility and speed in the *process* of conceiving, designing and implementing systems. It is about the ability of the process to respond to new requirements and information during system development.

In this latter spirit, actually *software* development approaches have become popular recently around the notion of "agility", where the most popular today is *Scrum* [2]. A major core idea of these approaches is *iterative and incremental development* (IID), which is actually an old and proven idea.

So, how about "agile" *systems development*? Is it possible to transfer such approaches directly from software to general systems development? Can hardware involved in such systems be developed incrementally in the same rapid iterations as software?

We try to answer these questions in this paper in the following manner. First, we sketch related work on this subject. Then we argue about inherent difficulties arising with hardware increments, which cause issues in agile systems development. Finally, we propose an IID life cycle that takes these limitations with hardware increments into account.

II. RELATED WORK

Research in agile systems engineering is still in its infancy, and most of the work investigates the potential utilization of agile approaches in general, or just emphasizes a need for such approaches.

By performing a series of interviews with industry representatives, Stelzmann [3] investigated under which conditions agile Systems Engineering could be used for the development of systems that have a major hardware portion. His results suggest that manufacturing, prototyping and testing of hardware is expensive and takes a lot of time, and implementing changes is hard. For these reasons, his interviewees did not apply agile development for hardware. However, his results also indicate that there is a strong need for more agility in systems engineering, in particular due to dynamic business environments. The industry representatives also claim that agile approaches would better support innovative products as well as complex systems that require more prototyping.

Development of a software-intensive system contains three important aspects according to [4]: business, system, and software. The business aspect is responsible for the economic and operational characteristics of the system including contracting, funding, operational requirements, and overall system delivery structure. The system aspect is responsible for the overall technical and technical management aspects of the system, and the software aspect deals with the software in the system. These authors state that, while some agile approaches have been introduced and executed for business and software aspects, there is a lack of such approaches for the system aspect. We agree with this observation, and claim that caution is needed in transferring these approaches to the engineering of general systems. In addition, these authors argue for development of an agile systems engineering framework but do not give any concrete details how such a framework would look like.

Haberfellner and de Weck [1] propose "Piecemeal Engineering" for the agile adaptation of existing, modular systems. They propose, e.g., to introduce new modules into

 ISBN: 978-1-61208-184-7

already existing systems first, and to use such modules for new systems afterwards. Another approach for more agility proposed in the same paper is "Set-Based-Design". Here, designers should work on a set of design alternatives in parallel, and the final design emerges over a sequence of steps (the "best" solution "wins"). However, they do not give any information about how such steps (iterations) should be organized and do not particularly address the specifics of software versus hardware development.

Taxen and Pettersson [5] present *Integration Driven Development* to combine plan-driven, incremental development with agile methods. They define "deltas", which are the incremental system's changes. This approach consists of rigorous planning, but agile realization of such deltas. They propose that such deltas should be implemented in a short time frame (e.g., a couple of weeks), and that this implementation results always in a new working version of the system. It seems, however, that also these authors focus only on software systems (even though large ones) and do not specify how to deal with hardware.

III. Inherent difficulties with hardware increments

Since in our view inherent difficulties with hardware increments pose a key issue for agile systems development, let us briefly sketch them.

First, it is important to understand the difference between iterations and increments in IID. Iterations mean repetitions in the process, whether they involve increments or not. Increments mean extensions of the (software) system in (small) portions.

Such iterations involving increments are at the core of current agile approaches to software development, in particular relatively short iterations. For instance, Scrum involves rapid iterations (with continuous customer input along the way), no longer than one month and usually more than one week.

However, certain inherent properties of hardware development constrain the development of increments. Such properties are, for example, that mechanical parts usually have to be fully built before they can be used in the system, and cabling, which usually has to be done at once for all planned subsystems. This, in turn, requires a final design and a manufactured frame or cover upfront. Further, these properties require a completed design of most of the components before building them. For example, it is necessary to have a good estimate of the power consumption of all electrical parts to be able to specify the required batteries and, in turn, their dimensions.

In general, hardware cannot be built in the timeframe of, e.g., a Scrum iteration. In contrast to software, evolutionary prototyping is usually not possible, since the flexibility required for hardware prototyping is too expensive for production.

Therefore, it is not really possibly to simply apply a current approach to agile software development for the development of general systems involving hardware.

IV. A life cycle model including hardware increments

For agile systems development, we present a life cycle model for systems engineering as illustrated in Figure 1 (we started our work in the context of a prototypical shopping robot [6]). This life cycle model emphasizes the different character of software and hardware development within systems development and recommends what should happen in a controlled development of general systems containing software and hardware. It is iterative and incremental but takes the differences between increments in software and hardware into account.

Although iterations are not necessarily tied to the increments (e.g., the iterations could only improve or polish existing features and not add something new), we focus only on the iterations that are related to increments and implement additional features.

We distinguish between *planned features* and *new features*. Planned features are typically already agreed between stakeholders and prioritized for implementation. New features typically arise from new stakeholder wishes caused by the feedback from the system's users. They are prioritized together with the remaining planned features.

We also distinguish between three types of iterations by the "grade" of the associated increment: iterations with *major*, *minor*, and *almost no* increments. Iterations with major increments are only possible in software as well as for the complete system when next product releases are planned. Iterations with minor, and almost no increments are typical for hardware development.

Let us now give some more process details about the life cycle model as illustrated in Figure 1. It starts with Elicitation of Stakeholder Wishes as a basis of System Requirements Engineering. Defined requirements serve as input for System Architecting, where the overall system architecture is developed. This activity includes also analysis of possible architectural patterns and variants, design space exploration, architectural constraints, etc. We have studied iterations between System Requirements Engineering and System Architecting before [7].

The System Decomposition activity focuses on the breakdown of the system into subsystems and allocation of the planned functionality to the subsystems. Such subsystems typically contain both software and hardware, and can be more or less complex. Depending on their complexity, they may be further decomposed recursively. Finally, each of these subsystems may be further decomposed into software and hardware, which are separately developed in Software Development and Hardware Development activities. Although the allocation of functionality to software and/or

 ISBN: 978-1-61208-184-7

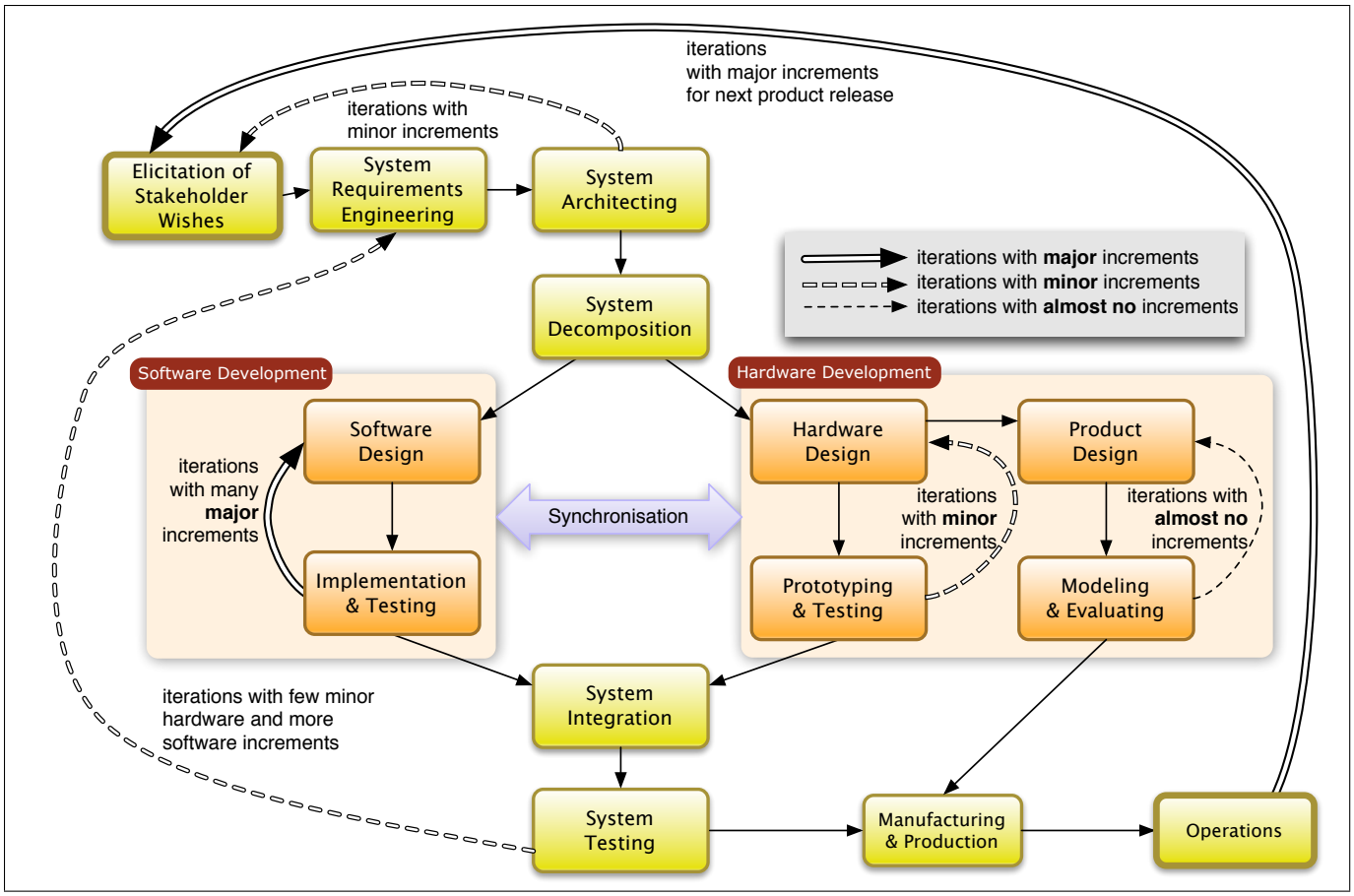


Figure 1. Life Cycle Model for Agile Systems Development

hardware might seem trivial at first (e.g., computations are done in software and movement is performed by drive trains and wheels), this allocation can be challenging. For example, image processing could be done in software but also in (programmable) hardware, and sophisticated techniques could be applied to find a good allocation.

The system architecture and its style, the separation into subsystems and allocation of functionality to hardware or software already define some limitations on increments. When considering software, e.g., the addition of increments in plug-in or component-based architectures is much easier than in a model-view-controller architecture.

The Software Development process might go through many iterations consisting of Software Design and Implementation & Testing activities with *major* increments. The number and size of increments depends mainly on the choice of the software development method. Using a rapid application development process results in more iterations and smaller increments than software development based on the Unified Process. Figure 1 expresses the possibility to include iterations with major increments in the software development process by the double line going from Implementation & Testing back to Software Design. This part of the development may well be a form of evolutionary prototyping. Especially when fast innovation is crucial, evolutionary prototyping can be useful for software development.

Electro-mechanical products are usually more limiting in regard to incremental development. Developing a hardware prototype typically involves the creation of a rather flexible hardware structure based on hardware toolkits, (e.g., mechanic toolkits, or electronic development boards). However, such toolkits are usually too expensive for mass production and do not support appealing product designs. Thus, developing a final product is even more resistive to incremental development. Therefore, we distinguish prototype development from final product development. The two parts in the Hardware Development box in the figure show this distinction.

The first activity, which we call Hardware Design, is dedicated to the development of the mechanical structure of the hardware according to its functionality and the design of the electronic hardware. Such hardware design is both the basis for prototype development built on hardware toolkits and for designing the final product. Prototypes developed

 ISBN: 978-1-61208-184-7

here will probably have to be "thrown away", i.e., it will generally not be possible to evolve them to become the final product. Hardware prototyping based on toolkits provides more flexibility and supports multiple iterations with minor increments. It is important to note that the architecture or basic design of the prototype must support increments. It is usually not possible to add major increments without a redesign.

Hardware and software development have to be synchronized in a way that increments added to software and hardware complement each other to allow integration and test of software and hardware components. The so-called anchor point milestones defined for spiral development [8], defining progress of development, may also serve for synchronization of system and software development efforts.

After System Integration, which includes the integration of subsystems into the complete system, there is System Testing. After that, the process can go back to System Requirements Engineering if some of the "planned features" are not yet implemented. Results from product integration and testing can also have other impact on the system requirements, leading to their changes or adding further requirements for the next iteration.

The Product Design activity creates a design of the final product including an appealing look and taking into account materials and guidelines for its manufacturing. Product Design works with producing models and evaluating them with stakeholders and users. Since these models have to give an impression of the final product, iterations have almost no increments. After having tested the functionality with prototypes and evaluated the usability and look based on models (Modeling and Evaluating), the product can be manufactured and put into Operations. For the overall product development life cycle, feedback from the product's users during Operations can lead to new stakeholder wishes and thus to new requirements and major increments for the next product release.

V. Conclusion

This paper argues that "agile" approaches to software development cannot be simply transferred to general Systems Engineering and development of systems involving hardware as well. The reason is that IID cannot be utilized in the same extensive manner as with software, since hardware increments have inherent limitations. Currently, there is a hype of agile software development. Thus it is important to make these inherent limitations clear, so that no naive expectations in "Agile Systems Engineering" may manifest themselves.

This paper also presents a proposal for an iterative and incremental life cycle for general Systems Engineering that takes the limitations of hardware increments into account. Of course, it will yet have to be shown how such an approach can be applied successfully in Systems Engineering practice, and its generality will also have to be investigated yet.

References

[1] R. Haberfellner and O. de Weck, "Agile SYSTEMS ENGINEERING versus AGILE SYSTEM engineering," in *Proceedings of the Fifteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE)*, July 2005, pp. 10–15.

[2] P. Deemer, G. Benefield, C. Larman, and B. Vodde, "The Scrum primer," 2010, version 1.2.

[3] E. Stelzmann, "Contextualizing agile systems engineering," in *Proceedings of the IEEE International Systems Conference (SysCon)*, April 2011, pp. 163–167.

[4] M. R. Kennedy and D. A. Umphress, "An Agile Systems Engineering Process: The Missing Link," *CrossTalk: The Journal of Defense Software Engineering*, vol. 4, no. 3, pp. 16–20, May/June 2011.

[5] L. Taxen and U. Pettersson, "Agile and Incremental Development of Large Systems," in *Proceedings of the 7th European Systems Engineering Conference (EuSEC 2010)*. Stockholm, Sweden: INCOSE, April 2010.

[6] H. Kaindl, J. Falb, E. Arnautovic, and D. Ertl, "Increments in an Iterative Systems Engineering Life Cycle," in *Proceedings of the 7th European Systems Engineering Conference (EuSEC 2010)*, Stockholm, Sweden, April 2010.

[7] H. Kaindl, E. Arnautovic, D. Ertl, and J. Falb, "Iterative requirements engineering and architecting in systems engineering," in *Proceedings of the Fourth International Conference on Systems (ICONS '09)*, March 2009, pp. 216–221.

[8] B. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, May 1988.

 ISBN: 978-1-61208-184-7

Towards Applying Normalized Systems Concepts to Modularity and the Systems Engineering Process

Peter De Bruyn, Herwig Mannaert
Department of Management Information Systems
University of Antwerp
Antwerp, Belgium
{*peter.debruyn, herwig.mannaert*}*@ua.ac.be*

***Abstract*—Current organizations need to be able to cope with challenges such as increasing change and increasing complexity. Modularity has frequently been suggested as a powerful means for reducing that complexity and enabling flexibility. As Normalized Systems (NS) theory has proven to introduce this evolvable modularity in software systems, this paper further explores the generalization of NS systems engineering concepts to modularity and the systems engineering process in general, and organizational systems in particular. After emphasizing the distinction between blackbox and whitebox perspectives on systems, we focus on the importance of employing exhaustively defined interfaces as a prerequisite to obtain 'true' black-box modules. Some aspects of the functional/constructional transformation are discussed. Finally, six additional interface dimensions are proposed as possible aspects to be included in such exhaustive interfaces when considering organizational modularity.**

***Keywords*-Normalized Systems, Modularity, Systems engineering, Evolvability, Systems theoretic stability**

I. INTRODUCTION

Current organizations need to be able to cope with increasing change and increasing complexity in many or all of their aspects. In this regard, modularity has previously been suggested as a powerful means for reducing that complexity by decomposing a system into several subsystems. Moreover, modifications at the level of those subsystems in stead of the system as a whole are said to be facilitating the overall evolvability of the system. More specifically, *Normalized Systems (NS) theory* has recently proven to introduce this evolvable modularity, primarily at the level of software systems. First, the theory states that the implementation of functional requirements into software constructs can be regarded as a *transformation* of a set of requirements $\mathcal{R}$ into a set of software primitives $\mathcal{S}$ [1], [2], [3]:

$$\{\mathcal{S}\} = \mathcal{I}\{\mathcal{R}\}$$

Next, in order to obtain evolvable modularity, NS theory states that this transformation should exhibit systems theoretic stability, meaning that a bounded input function (i.e., bounded set of requirement changes) should result in a bounded output values (i.e., a bounded impact or effort) even if an unlimited systems evolution is considered. Furthermore, Mannaert et al. [2] have formally proven that this implies that modular structures should strictly adhere to the following *principles*:

- *Separation of Concerns*, enforcing each change driver to be seperated;
- *Data Version Transparency*, enforcing communication between data in version transparant way;
- *Action Version Transparency*, requiring that action components can be updated without impacting calling components;
- *Separation of States*, enforcing each action of a workflow to be seperated from other actions in time by keeping state after every action.

As this results in very fine-grained modular structures, NS theory proposes to build information systems based on the aggregation of instantiations of five higher-level software software *elements*, i.e., action elements, data elements, workflow elements, trigger elements and connector elements [1], [2], [3]. Typical cross-cutting concerns (such as remote access, persistence, access control, etc.) are included in these elements in a way which is consistent with the above-mentioned theorems.

However, we claim that many other systems could also be regarded as modular structures. Both functional (i.e., requirements) and constructional (i.e., primitives) perspectives can frequently be discerned, modules can be identified and thus the analysis of the functional/constructional transformation seems relevant. Indeed, Van Nuffel has recently shown the feasibility of applying modularity and NS theory concepts at the business process level [4], [5] while Huysmans did so at the level of enterprise architectures [6]. However, the extension of NS theory to these domains has not been formalized yet. Consequently, as NS theory proved to be successful in introducing evolvable modularity in software systems, and as it is clearly desirable to extend such properties to other systems, this paper focuses on a first step towards generalizing NS theory concepts to systems engineering in general. More specifically we will focus on and stress the importance of a complete and unambiguous definition of the interface of modular subsystems as an

 ISBN: 978-1-61208-184-7

essential part of each systems engineering process. Next, by way of illustration, we discuss an initial attempt of applying our approach to organizational systems, motivated by the frequently observed discrepancy between the uttered need to more systematically engineer organizational artifacts [7], [8] and empirical findings suggesting that mostly only ad hoc approaches are employed in practice [9].

The remainder of this paper is structured as follows. Section II will discuss some extant literature on modularity, emphasizing the work of Baldwin and Clark. Next, we propose a more unambiguous definition of modularity after discussing functional and constructional perspectives on systems (Section III) and outlining some of the transformation properties (Section IV). Finally, some exemplifying additional interface dimensions when considering organizational modules will be suggested (Section V), as well as some conclusions and opportunities for future research (Section VI).

II. Related Work

The use of the concept of modularity has been noticed to be employed in several scientific domains such as computer science, management, engineering, manufacturing, etcetera [10]. While no single generally accepted definition is known, the concept is most commonly associated with the process of subdividing a system into several subsystems [11], [12]. This decomposition of complex systems is said to result in a certain degree of complexity reduction [13] and facilitate change by allowing modifications at the level of a single subsystem in stead of having to adapt the whole system at once [14], [10], [15].

As such, Baldwin and Clark defined modularity as follows: "*a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units*" [15, p. 63]. They conceive each system or artifact as being the result of specifying values for a set of design parameters, such as the height and the vessel diameter in case of a tea mug. The task of the designer is then to choose the design parameter values in such a way, that the 'market value' of the system as a whole becomes maximized. Some of the design parameters might be dependent on one another, as for example the value of the vessel diameter should be attuned to the value of the diameter of a mug. Consequently, modularization is conceived as the process in which groups of design parameters — highly interrelated internally, but loosely coupled externally — are to be identified as modules and can be designed rather independently from each other, such as for instance the drive system, main board and LCD screen in case of a simplified computer hardware design. A set of design rules (visible information) is used to secure the compatibility between the subsystems in order to be assembled into one working system later on, while the other design parameters are only visible for a module itself. Finally, they conclude that this modularity allows multiple (parallel) experiments for each module resulting in a higher 'option value' of the system in its totality. Instead of just accepting or declining one system as a whole, a 'portfolio of options' can be considered, as designers can compose a system by purposefully selecting among a set of alternative modules. Systems evolution is then believed to be characterized by the following six modular operators [15]:

- *Splitting* a design (and its tasks) into modules;
- *Substituting* one module design for another;
- *Augmenting*, i.e., adding a new (extra) module to the system;
- *Excluding* a module from the system;
- *Inverting*, i.e., isolating common functionality in a new module, thus creating new design rules;
- *Porting* a module to another system.

Typically, besides traditional phyisical products, many other types of systems are claimed to be able to be regarded as modular structures as well. First, all different programming and software paradigms can be considered as using modularity as a main concept to build software applications [1]. Furthermore, while Baldwin and Clark primarily illustrate their discussion by means of several evolutions in the computer industry, they also explicitly refer to the impact of product modularity on the (modular) organization of workgroups both within one or multiple organizations, and even whole industry clusters [15]. Also, Campagnolo and Camuffo [12] investigated the use of modularity concepts within management science and identified 125 studies in which modularity concepts arose as a design principle of organizational complex systems, suggesting that the principles of modularity offer powerful means to be applied at the organizational level.

Within the field of Enterprise Engineering, trying to give prescriptive guidelines on how to design organizations, modularity equally proved to be a powerful concept. For instance, Op't Land used modularity related criteria to merge and split organizations [16]. Van Nuffel proposed a framework to deterministically identify and delimit business processes based on a modular and NS theory viewpoint [4], [5], and Huysmans demonstrated the usefulness of modularity with regard to the study of (the evolvability) of enterprise architectures [6].

III. Towards a Complete and Unambiguous Definition of Modules

While we are obviously grateful for the valuable contributions of the above mentioned authors, we will argue in this section that the definition of modularity, as for example coined by Baldwin and Clark [15], already describes an ideal form of modularity (e.g., loosely coupled and independent). As such, we will first discuss the need to distinguish the functional and constructional perspectives of systems. Next, we will propose to introduce the formulation

 ISBN: 978-1-61208-184-7

of an exhaustive modular interface as an intermediate stage, being a necessary and sufficient condition in order to claim 'modularity'. The resulting modules can then be optimized later on, based on particular criteria.

A. Blackbox (Functional) versus Whitebox (Constructional) Perspectives on Modularity

When considering systems in general — software systems, organizational systems, etcetera — both a functional and constructional perspective should be taken into account [17]. The functional perspective focuses on describing what a particular system or unit does or what its function is [18]. While describing the external behavior of the system, this perspective defines input variables (what does the system need in order to perform its functionality?), transfer functions (what does the system do with its input?) and output variables (what does the system deliver after performing its functionality?). As such, a set of general requirements, applicable for the system as a whole, are listed. The structural perspective on the other hand, concentrates on the composition and structure of the system (i.e., which subsystems are part of the system?) and the relation of each of those subsystems (i.e., how do they work together to perform the general function and adhere to the predefined requirements?) [19].

Equivalently, one could regard the functional system view as a blackbox representation, and the constructional system view as a witebox representation. By blackbox we mean that only the input and output of a system is revealed by means of an interface, describing the way how the system interacts with its environment. As such, the user of the system does not need to know any details about the content or the inner way of working of the system. The way in which the module performs its tasks is thus easily allowed to change and can evolve independently without affecting the user of the system, as long as the final interface of the system remains unchanged. The complexity of the inner working can also be said to be hided (i.e., information hiding), resulting in some degree of complexity reduction. The whitebox view does reveal the inner way of working of a system: it depicts the different parts of which the system consists in terms of primitives, and the way these parts work together in order to achieve the set of requirements as listed in the blackbox view. However, each of these parts or subsystems is a 'system' on its own and can thus again be regarded in both a functional (blackbox) and constructional (whitebox) way.

The above reasoning is also depicted in Figure 1: both Panels represent the same system $SysA$, but from a conceptually different viewpoint. Panel (a), depicting the functional (blackbox) view, lists the requirements (boundary conditions) R_1, R_2, ... imposed to the system. These are proposed as 'surrounding' the system in the sense that they do not say anything about how the system performs its tasks, but rather discuss what it should perform by means of an interface in terms of inputs and outputs. Panel (b) depicts the constructional (whitebox) view of the same system: the way of working of an aggregation of instantiations of primitives P_1, P_2, ... (building blocks), collaborating to achieve the behavior described in Panel (a). Each of the primitives in Panel (b) is again depicted in a blackbox way and could, at their turn, each also be analyzed in a constructional (whitebox) way.

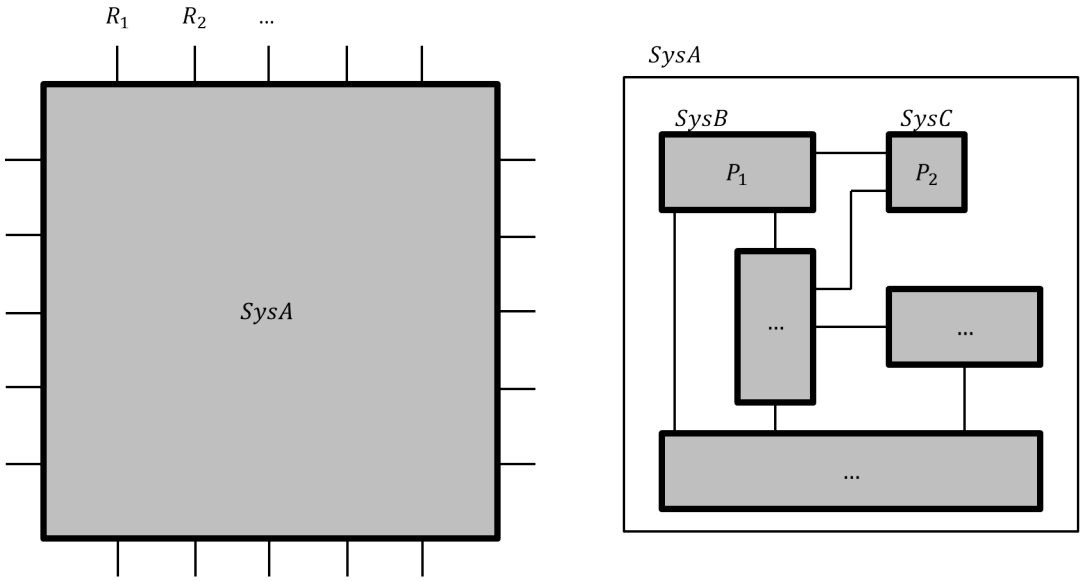
B. Avoiding Hidden Coupling by Strictly Defining Modular Interfaces

Before analyzing and optimizing the transformation between both perspectives, the designer should be fully confident that the available primitives can really be considered as 'fully fledged, blackbox modules'. By this, we mean that the user of a particular module should be able to implement it, exclusively relying on the available interface, thus without having any knowledge about the inner way of working of the concerned module. Stated otherwise, the interface of the module should describe any possible dependency regarding the module, needed to perform its functionality. Consequently, every interaction of the system with its environment should be properly and exhaustively defined herein. While this may seem rather straightforward at first sight, real-life interfaces are rarely described in such a way. Indeed, typical non-functional aspects such as technological frameworks, infrastructure, knowledge, etc. are consequently also to be taken into account (cf. Section V). Not formulating these 'tacit assumptions' results in hidden coupling: while the system is claimed to be a module, it actually still needs whitebox inspection in order to be implemented in reality, diminishing the pretended complexity reduction benefits.

Consider for instance a multiplexer for use in a typical processor, selecting and forwarding one out of several input signals. Here, one might conceptually think at a device having for example 8 input signals, 3 select lines and 1 output signal. While this is conceptually certainly correct, a real implementation on a real processor might for example require 120μ by 90μ CMOS (i.e., material) to make the multiplexer physically operational on the processor, while this is not explicitly mentioned in its conceptual interface. As such, this 'resource dimension' should be made explicit in order to consider a multiplexer as a real black box in the sense that the module can be unambiguously and fully described by its interface. A person wanting to use a multiplexer in real-life in a blackbox way, should indeed be aware of this prerequisite prior to his ability of successfully implementing the artifact.

A more advanced example of hidden coupling includes the use of a 'method' in typical object-oriented programming languages, frequently suggested as a typical example of a 'module' in software. Indeed, in previous work, it was argued to consider the multidimensional variability when

 ISBN: 978-1-61208-184-7



(a) Blackbox (functional) representation of $SysA$. R_1, R_2, etc. represent the requirements imposed to $SysA$ as a whole.

(b) Whitebox (constructional) representation of $SysA$. Requirements R_1, R_2, etc. are 'realised' through the collaboration of a set of instances of primitives P_1, P_2, etc. in $SysB$, $SysC$, etc. At their turn, $SysB$, $SysC$, etc. are represented here in a blackbox way.

Figure 1. Blackbox (functional) and whitebox (constructional) representations of system $SysA$.

analyzing the evolvability of programming constructs (such as data structures and processing functions) and that in typical object-oriented programming environments these dimensions of variability increase even furter as they make it possible to combine processing actions and data entities into one primitive (i.e., a single class) [2]. Hence, it was argued to start the analysis of object-oriented modular structures already at the level of methods instead of only considering a class as a possible 'module'. However, while it is usually said that such a method in object orientation has an interface, this interface is not necessarily completely, exhaustively and fully defined and thus such a method cannot automatically be considered as a 'real module' according to our conceptualization. Consider for example the constructor of the class in which the method has been defined. Typically, the constructor has to perform certain actions (e.g., making an instantiation (object) of the concerned class) before one can execute the concerned method. Also member variables of the class might introduce hidden coupling: first, they can be manipulated by other methods as well, outside control of the considered method. Second, they have to be created ('exist') before the module can perform its functionality. Finally, employing external libraries in case a method wants to be deemed a genuine module, would imply that ór the library should be incorporated into the module (each time) ór the external library should be explicitly mentioned in the interface.

Hence, in our view, one has a genuine module as soon as one is able to define a complete interface which clearly describes the boundaries and interactions of the subsystem and allows it to be used in a blackbox way. Modularization is then the process of meticulously identifying each dependency of a subsystem, transforming an ambiguously defined '*chunk*' of a system into a clearly defined *module* (of which the borders, dependencies, etc. are precisely, exante, known). Compared to the definition of Baldwin and Clark cited previously, we thus do not require for a module to exhibit already high intramodular cohesion and low intermodular coupling at this stage. Modules having these characterics are nevertheless obviously highly desirable. However, we are convinced that defining in a first phase such a complete interface, allows to 'encapsulate' the module in an appropriate way and avoid any sort of hidden coupling. Indeed, at least four out of the six mentioned modular operators in Section II require real blackbox (re)usable modules as a conditio sine qua non. More specifically, in order to use the operators Substituting, Augmenting, Excluding and Porting in their intended way, complete and exhaustively defined interfaces are a prerequisite. On the other hand, Splitting and Inverting concern the definition of new modules and design rules. Hence, they are precisely focused on the process of defining new modular interfaces themselves, thus usually involving some form of whitebox inspection.

Finally, while defining modules with such a strict interface will not directly solve any interdependency, evolvability, ... issues, it will at least offer the possibility to profoundly study and optimize the 'quality' of the modules (e.g., with regard to coupling and cohesion) in a next stage.

 ISBN: 978-1-61208-184-7

IV. Towards Applying Systems Engineering on the Function/Construction Transformation

In the same way as the implementation of software is considered as a transformation $\mathcal{I}$ of a set of functional requirements into a set of software primitives (constructs) in [2], the design or engineering of systems in general could be considered as a transformation $\mathcal{D}$ of a set of functional requirements R_j into a set of subsystems or primitives P_i:

$$\{P_i\} = \mathcal{D}\{R_j\}$$

This transformation $\mathcal{D}$ can then be studied and/or optimized in terms of various desirable system properties. In this section, we present a very preliminary discussion on the meaning of several important system properties in this respect.

Stability: As discussed in [2] for the software implementation transformation, any design transformation can be studied in terms of stability. This means that a bounded set of additional functional requirements results only in a bounded set of additional primitives and/or new versions of primitives. As elaborated by Mannaert et al. [2], this would require the absence of so-called combinatorial effects resulting in an impact of additional functional requirements that is proportional to the size of the system. An example of an unstable requirement is for instance a small software application in an "office" environment that needs to become highly secure and reliable, requiring a completely new and different implementation.

Scalability: Scalability would mean that the increase in value of an existing functional requirement has a clearly defined and limited impact on the constructional view. An example of such a scalable requirement is the amount of concurrent users of a website, which can normally be achieved by adding one or more additional servers. Examples of unscalable requirements in current designs are the increase in the number of passengers in an airplane, leading to the design of a completely new airplane, or the increase in the target velocity of a rocket, leading to the design of a totally different rocket.

Normalization: It seems highly desirable to have a linear design transformation that can be normalized. This would imply that the transformation matrix becomes diagonal or in the Jordan form, leading to a one-to-one mapping of functional requirements to (a set of) constructional primitives. Such a normalized transformation is explored in [1], [2] for the implementation of elementary functional requirements into software primitives (in this case elements as structured aggregations of primitives).

Isentropicity: Entropy is defined in statistical thermodynamics as the number of microstates for a given macrostate, corresponding to the uncertainty of the detailed internal system state with respect to the observable external state [20], [21]. In our view, an isentropic design would therefore imply that the external observable state of $SysA$ completely and unambiguously determines the states of the various subsystems. An example of such an isentropic design is a finite state machine where the various registers can be read. Indeed, the inputs and register values that are externally observable completely define the internal state of the finite state machine.

This approach to systems design or engineering also seems to imply that we should avoid to perform functional decomposition over many hierarchical levels, before starting the composition or aggregation process [22]. Studying and/or optimizing the functional to constructional transformation is a very delicate activity that can only be performed on one or two levels at a time. Therefore, the approach seems to imply a preference for a bottom-up or meet-in-the-middle approach, trying to devise the required system (i.e., the set of functional requirements R_j) in terms instantiations of a set of predefined primitives (i.e., P_i), over a top-down approach.

V. On a Complete and Unambiguous Definition of Organizational Modules

In Sections I and II we argued that not only software applications can be regarded as modular systems, but also many other types of artifacts, such as (for example) organizations. Hence, Sections III and IV focused on a first attempt to extend NS theory concepts to modularity and the systems engineering process in general. In this section, by means of example, we will illustrate some of the implications of our proposed engineering approach when applied to organizational systems. Indeed, several authors have argued for the need of the emergence of an Enterprise Engineering discipline, considering organizations as (modular) systems which can be 'designed' and 'engineered' towards specific criteria [7], [8], such as (for example) evolvability. More specifically, we will primarily focus our efforts here on the complete and unambiguous definition of organizational modules, as this is in our view a necessary condition to be able to study and optimize the functional/constructional transformation at a later stage.

Consequently, when also considering modules at the organizational level, a first effort should equally be aimed at exhaustively listing the interface, incorporating each of its interactions. For instance, when focusing on a payment module, not only the typical 'functional' interface such as the account number of the payer and the payee, the amount and date due, etc. (typical 'arguments') but also the more 'configuration' or 'administration' directed interface including the network connection, the personnel needed, etc. (typical 'parameters') should be included. As such, we might distinguish two kinds of interfaces:

- a *usage interface*: addressing the typical functional (business-oriented) arguments needed to work with the module;

 ISBN: 978-1-61208-184-7

- a *deployment interface*: addressing the typical non-functional, meta-transaction, configuration, administration, ... aspects of an interface.

Although some might argue that this distinction may seem rather artificial and not completely mutually exclusive, we believe that the differences between them illustrate our rationale for a completely defined interface clearly.

While the work of Van Nuffel [5] has resulted in a significant contribution regarding the identification and separation of distinct business processes, the mentioned interfaces still have the tendency to remain underspecified in the sense that they only define the functional 'business-meaning' content of the module but not the other dimensions of the interface, required to fully use a module in blackbox fashion. Such typical other (additional) dimensions — each illustrated by means of an imaginary organizational payment module — might include:

1) Supporting technologies: Modules performing certain functionality might need or use particular external technologies or frameworks. For example, electronical payments in businesses are frequently performed by employing external technologies such as SWIFT or Isabel. In such a case, a payment module should not only be able to interact with these technologies, but the organization should equally have a valid subscription to these services (if necessary) and might even need access to other external technologies to support the services (e.g., the Internet). An organization wanting to implement a module in a blackbox way should thus be aware of any needed technologies for that module, preferably by means of its interface and without whitebox inspection. Suppose that one day, the technology a module is relying on, undergoes some (significant) changes resulting in a different API (application programming interface). Most likely, this would imply that the module itself has to adapt in order to remain working properly. In case the organization has maintained clear and precise interfaces for each of its modules, it is rather easy to track down each of the modules affected by this technological change, as every module mentioning the particular technology in its interface will be impacted. In case the organization has no exhaustively formulated interfaces, the impact of technological changes is simply not known: in order to perform a confident impact analysis, the organization will have to inspect each of the implemented modules with regard to the affected technology in a whitebox way. Hence, technological dependencies should be mentioned explicitly in a module's interface to allow true blackbox (re)use.

2) Knowledge, skills and competences: Focusing on organizations, human actors clearly have to be taken into account, as people can bring important knowledge into an organization and use it to perform certain tasks (i.e., skills and competences). As such, when trying to describe the interface of an organizational module in an exhaustive way, the required knowledge and skills needed for instantiating the module should be made explicit. Imagine a payment module incorporating the decision of what to do when the account of the payer turns out to be insolvent. Besides the specific authority to take the decision, the responsible person should be able (i.e., have the required knowledge and skills) to perform the necessary tasks in order to make a qualitative judgment. Hence, when an organization wishes to implement a certain module in a blackbox way, it should be knowledgeable (by its interface) about the knowledge and skills required for the module to be operational. Alternatively, when a person with certain knowledge or skills leaves the company, the organization would be able to note immediately the impact of this knowledge-gap on the well-functioning of certain modules and could take appropriate actions if needed.

3) Money and financial resources: Certain modules might impose certain financial requirements. For example, in case an organization wants to perform payments by means of a particular payment service (e.g., SWIFT or Isabel), a fixed fee for each payment transaction might be charged by the service company. If the goal is to really map an exhaustive interface of a module, it might be useful to mention any specific costs involved in the execution of a module. That way, if an organization wants to deploy a certain module in a blackbox way, it may be informed about the costs involved with the module ex-ante. Also, when the financial situation of an organization becomes for instance too tight, it might conclude that it is not able any longer to perform the functions of this module as is and some modifications are required.

4) Human resources, personnel and time: Certain processes require the time and dedication of a certain amount of people, possibly concurrently. For example, in case of an organizational payment module, a full time person might be required to enter all payment transactions in the information system and to do regular manual follow-ups and checking of the transactions. As such, an exhaustive interface should incorporate the personnel requirements of a module. That way, before implementing a certain module, the organization is aware of the amount of human resources needed (e.g., in terms of full time equivalents) to employ the module. Equivalently, when the organization experiences a significant decline or turnover in personnel, it might come to the conclusion that it is no longer able to maintain (a) certain module(s) in the current way. Obviously, this dimension is tightly intertwined with the previously dicussed knowledge and skills dimension.

5) Infrastructure: Certain modules might require some sort of infrastructure (e.g., offices, materials, machines) in order to function properly. Again, this should be taken into account in an exhaustive interface. While doing so, an organization adopting a particular module knows upfront which infrastructure is needed and when a certain infrastructural facility is changed or removed, the organization

 ISBN: 978-1-61208-184-7

might immediately evaluate whether this event impacts the concerning module and vice versa.

6) Other modules or information: Certain modules might use other modules in order to perform their function. For example, when an organization decides to perform the procurement of a certain good, it will probably receive an invoice later on with a request for payment. While the follow-up of a procurement order might be designed into one module, it is reasonable to assume that the payment is designed in a distinct module, as this functionality might also return in other business functions (e.g., the regular payment of a loan). As such, when an organization is planning to implement the procurement module, it should be aware that also a payment module has to be present in the organization to finalize procurements properly. Hence, all linkages and interactions with other modules should be made explicit in the module's interface. When a module (including its interface), used by other modules, is changed at a certain point in time, the adopting organization then immediately knows the location of impact in terms of implemented modules and hence where remedial actions might be required.

Obviously, it is clear that exhaustively defining the technology, knowledge, financial resources, ... on which a module depends, will not suffice to solve any of the existing coupling or dependencies among modules. Also, one should always take into consideration that a certain amount of 'coupling' will always be needed in order to realistically perform business functions. However, when the interface of each module is clearly defined, the user or designer is at least aware of the existing dependencies and instances of coupling, knows that ripple-effects will occur if changes affect some of the module's interfaces (i.e., impact analysis) and can perform his or her design decisions in a more informed way, i.e., by taking the interface with its formulated dependencies into account. Consequently, once all forms of hidden coupling are revealed, finetuning and genuine engineering of the concerned modules (e.g., towards low intermodular coupling) seems both more realistic and feasible in a following phase. Indeed, one might deduct that Baldwin and Clark, while defining a module as consisting of powerfully connected structural elements, actually implicitly assumed the existence of an exhaustive set of formulated dependencies before modularization can occur. Our conceptualization is then not to be interpreted as being in contradiction with that of Baldwin and Clark, rather we emphasize more explicitly that the mapping of intermodular dependencies is not to be deemed negligible or self-evident.

VI. Conclusion and Future Work

This paper focused on the further exploration and generalization of NS systems engineering concepts to modularity and the systems engineering process in general, and organizational systems in particular. The current state-of-the-art regarding modularity was reviewed, primarily focusing on the seminal work of Baldwin and Clark. Subsequently, we argued that, first, a distinction should be made between blackbox and whitebox perspectives of systems. A system can then be considered as the transformation of (functional) requirements into (constructional) primitives. A preliminary discussion of some properties of this transformation was proposed. Next, in order to be able to fully (re)use those constructional primitives as 'blackbox building blocks', we proposed to define a module as a subsystem which can be completely described solely by its interface, thus indicating exhaustively all interactions and dependencies and hence avoiding hidden coupling. Finally, six additional organizational interface dimensions when considering organizational modules were suggested, implied by our approach. We concluded that our conceptualization is not in contradiction with that of Baldwin and Clark, but rather emphasizes an additional intermediate design stage when devising (organizational) modules.

A limitation of this paper is that no garantuee is offered that the identified additional interface dimensions will reveal all kinds of hidden coupling in every organization. Therefore, additional research (e.g., case studies) with regard to possible missing dimensions is required. In addition, our application of modularity and NS concepts to the organizational level was limited to the definition of completely defined organizational modules. The functional/constructional transformation on the organizational level was still out of scope in this paper. Furthermore, future research at our research group will be aimed at identifying and validating organizational black-box reusable modules, exhibiting exhaustively defined interfaces and enabling the bottom-up functional/constructional transformation.

Acknowledgment

P.D.B. is supported by a Research Grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[1] H. Mannaert and J. Verelst, *Normalized systems: re-creating information technology based on laws for software evolvability*. Koppa, 2009.

[2] H. Mannaert, J. Verelst, and K. Ven, "The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability," *Science of Computer Programming*, vol. Article in press, 2011.

[3] ——, "Towards evolvable software architectures based on systems theoretic stability," *Software Practice and Experience*, vol. Early View, 2011.

[4] D. Van Nuffel, H. Mannaert, C. De Backer, and J. Verelst, "Towards a deterministic business process modeling method based on normalized systems theory," *International Journal on Advances in Software*, vol. 3, no. 1-2, pp. 54–69, 2010.

 ISBN: 978-1-61208-184-7

[5] D. Van Nuffel, "Towards designing modular and evolvable business processes," Ph.D. dissertation, University of Antwerp, 2011.

[6] P. Huysmans, "On the feasibility of normalized enterprises: Applying normalized systems theory on the high-level design of enterprises," Ph.D. dissertation, University of Antwerp, 2011.

[7] J. L. G. Dietz, *Enterprise Ontology: Theory and Methodology*. Springer, 2006.

[8] J. Hoogervorst, *Enterprise Governance and Enterprise Engineering*. Springer, 2009.

[9] M. Indulska, J. Recker, M. Rosemann, and P. F. Green, "Business process modeling: Current issues and future challenges," in *CAiSE*, ser. Lecture Notes in Computer Science, P. van Eck, J. Gordijn, and R. Wieringa, Eds., vol. 5565. Springer, 2009, pp. 501–514.

[10] C. Y. Baldwin and K. B. Clark, "Managing in an age of modularity," *Harvard Business Review*, vol. 75, no. 5, pp. 84–93, 1997.

[11] H. Simon, *The Sciences of the Artificial*, 3rd ed. Cambridge, Massachusetts: MIT Press, 1996.

[12] D. Campagnolo and A. Camuffo, "The concept of modularity within the management studies: a literature review," *International Journal of Management Reviews*, vol. 12, no. 3, pp. 259 – 283, 2009.

[13] H. Simon, "The architecture of complexity," in *Proceedings of the American Philosophical Society*, vol. 106, no. 6, December 1962.

[14] R. Sanchez and J. Mahoney, "Modularity, flexibility, and knowledge management in product and organization design," *Strategic Management Journal*, vol. 17, pp. 63–76, 1996.

[15] C. Y. Baldwin and K. B. Clark, *Design Rules: The Power of Modularity*. Cambridge, MA, USA: MIT Press, 2000.

[16] M. Op't Land, "Applying architecture and ontology to the splitting and allying of enterprises," Ph.D. dissertation, Technical University of Delft (NL), 2008.

[17] G. M. Weinberg, *An Introduction to General Systems Thinking*. Wiley-Interscience, 1975.

[18] L. Bertalanffy, *General Systems Theory: Foundations, Development, Applications*. New York: George Braziller, 1968.

[19] M. Bunge, *Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems*. Boston: Reidel, 1979.

[20] L. Boltzmann, *Lectures on gas theory*. Dover Publications, 1995.

[21] Wikipedia. (2011) Entropy. [Online]. Available: http://en.wikipedia.org/wiki/Entropy

[22] P. De Bruyn, D. Van Nuffel, P. Huysmans, and H. Mannaert, "Towards functional and constructional perspectives on business process patterns," in *Proceedings of the Sixth International Conference on Software Engineering Advances (ICSEA)*, Barcelona, Spain, 2011, pp. 459–464.

 ISBN: 978-1-61208-184-7

In Search of Rules for Evolvable and Stateful run-time Deployment of Controllers in Industrial Automation Systems

Dirk van der Linden
Electro Mechanics Research Group
Artesis University College of Antwerp
Antwerp, Belgium
dirk.vanderlinden@artesis.be

Herwig Mannaert
Department of Management Information Systems
University of Antwerp
Antwerp, Belgium
herwig.mannaert@ua.ac.be

***Abstract*—Automation systems in the domains of smart grids, digital factories and modern process systems struggle to follow the permanent shift of their requirements. Hence, the most prominent non-functional requirement of a system seems to be evolvability. The recently proposed Normalized Systems theory has formulated constraints on the modular structure of software architecture in order to engineer evolvable systems. In this context, evolvability is related to systems theory stability as it is defined as the possibility to perform additional anticipated changes to the system of which the output remains bounded, even if an unlimited systems evolution is assumed. In this analysis, one considered the context of compile-time. However, this view becomes far more complex during run-time deployment, because some modules have several instances, others only one. The amount and complexity of connections during run-time is not straightforward visualizeable. In this paper, we introduce two new theorems, which are complementary with the existing four, to achieve a stateful run-time deployment.**

***Keywords*-Normalized Systems; Evolvability; Systems Theory; Modularity; Industrial Automation.**

I. INTRODUCTION

The non-functional requirement of evolvability is a very desirable characteristic for both information and production control systems. First, current information systems still struggle to provide high levels of evolvability [1]. Indeed, software maintenance is regarded as one of the most expensive phases of the software life cycle, and often leads to an increase of architectural complexity and a decrease of software quality [2]. However, contemporary organizations are increasingly faced with changing environments, which emphasizes the need for evolvability of software systems. The widely accepted shortage in programming manpower, and the disappointing success rate in business software development projects, call for a major gain in this kind of software development. Second, automation software should be able to evolve over time as well. This is a key requirement in the beginning age of decentralized energy generators and consumers prominently known as smart grid [3]. The upcoming of PLCs (Programmable Logical Controllers) some 40 years ago, has provided more flexibility to develop and maintain automation systems in terms of software in spite of hardware (i.e., electrical circuits). The dynamic interchange of software components of a PLC with near-to-zero downtime some years ago, has provided the flexibility to alter automation systems while staying in full service [4]. The modification of an automation system should be possible without affecting existing parts, even if running parts are reused in a so-called online change (i.e., downloading a new software part to the controller without stopping the system).

Normalized Systems theory has recently been proposed to contribute in achieving the characteristic of evolvability in systems. Requiring stability as defined in systems theory, four design principles or theorems are proposed. Systems built with modules, which comply with these theorems, can increase without losing control over the so-called combinatorial effects of a change. A bounded set of anticipated changes should result in a bounded amount of impacts to the (growing) system.

One can visualize an overview of a system by placing a number of modules on a surface, and connecting them through their interfaces. Since a good interface is roughly explaining the core functionality of a module, it may seem rather straightforward to consider the relations between the modules via their connections. However, from the moment the (compiled) code starts to run, obtaining this overview is even far more complex. For instance, some modules have in run-time several instances, others only one. The amount, complexity and dimensions of connections during run-time is not straightforwardly visualizeable. However, obtaining such overview of the runtime situation in an automation project is necessary to control evolvability, and predict combinatorial effects. There is need to minimize downtimes by dynamic reconfiguration of a system, without a complete shutdown. It is important to note the contrast with a static configuration, which does need a complete shutdown of the system. Such a static reconfiguration is very costly and should be replaced by a dynamic one [4].

In this paper, we introduce two new theorems, which are complementary with the existing four, to achieve a stateful and an evolvable run-time deployment.

The paper is structured as follows. In Section II, we

 ISBN: 978-1-61208-184-7

mention some related work. In Section III, the Normalized Systems theory will be discussed. In Section IV, we introduce the two new theorems. Finally, conclusions and future research are discussed in Section V.

II. Related Work

One of the motivations of Dijkstra to argue for the abolishment of the GOTO statement from all "higher level" programming languages was the finding that "...our intellectual powers are rather geared to master static relations and that our powers to visualize processes evolving in time are relatively poorly developed. For that reason we should do (as wise programmers aware of our limitations) our utmost to shorten the conceptual gap between the static program and the dynamic process, to make the correspondence between the program (spread out in text space) and the process (spread out in time) as trivial as possible." [8]

Most modern higher level programming languages do not allow GOTO statements, but we think more rules can contribute to address Dijkstra's ambition. Even without the GOTO statement the "dynamic process" (i.e., its dispersal in time) is still difficult to overview. Designing and developing modules, connected through interfaces with other modules, may seem rather straightforward when they are modeled in a static way (i.e., dispersed in text space or graphical visualizations). Studying the behavior of these modules in a dynamic view is even more complex. For example, one type or class definition results in run-time deployment in several instances. Some modules have a lot of instances, others only one. Consequently, representing the behavior of modules, together with their interrelations and instances during run-time is far from easy.

Additionally, visualizing and overviewing processes is harder if the system becomes more complex. Reducing complexity is a way to facilitate the formation of overviews. Employing meta data and information modeling contributes on this field. For example, Mahnke et al. have proposed classifications of types of information modeling standards for automation [9]. Also, they discuss the applicability of possible approaches to expose those models.

Further, Kuhl and Fay introduced an approach to modify automation systems by way of a middleware concept [4]. They focus on reconfiguration of systems, even during run-time, when a shutdown is not possible. This reconfiguration should not affect existing, running parts. Consequently, in such a system we have running instances, which are instantiated with a specific type version, and new constructions, which provide instances with a new type version. These instances should be able to co-exist, even when they are slightly different because of the different type versions they are based on.

Version transparency is one of the key points of the Normalized System theory to achieve stability [10]. As a consequence, different versions of both data entities and action entities can co-exist simultaneous. At first sight, the co-existency of different versions is not contributing in making the correspondence between the program and the run-time deployment trivial (i.e., what Dijkstra called for). However, allowing only one version (typically the most recent one) in an evolving system leads to unbounded combinatorial effects. The principle of version transparency is providing the possibility to overview nevertheless the different versions. The question is, how can we achieve an overview of run-time instances of these primitives, each constructed in one of the co-existing versions.

III. Normalized Systems

The law of Increasing Complexity (Lehman [11]) states: "As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it". This degradation of a system's structure over time is well known. More difficult to determine is the detailed cause of this deterioration. Which new parts of the system contribute in the effects of this law? In other words, why is a piece of code causing more costs in the *mature* stage of the lifecycle of a system, than exactly the same piece of code, is causing in the *beginning* stage of the project? The authors of the Normalized Systems theory combine Lehman's law of Increasing Complexity with the assumption of unlimited systems evolution: *The system evolves for an infinite amount of time, and consequently the total number of requirements and their dependencies will become unbounded*. These authors admit that in practice this assumption is an overstatement for the most commercial applications, but it provides a theoretic view on the evolvability issue, which is independent of time. The rather vague questions like "Is this change causing more troubles than another?" can be replaced by the fundamental question: "Is this change causing an unbounded effect?". The authors of Normalized Systems want to provide a deterministic and unambiguous yes/no answer on this question, by evaluation whether one of the theorems is violated or not.

A. Stability

The single postulate, from which the Normalized Systems theory is derived from, states that *a system needs to be stable with respect to a defined set of anticipated changes*. In systems theory, one of the most fundamental properties of a system is its stability: a bounded input function results in bounded output values, even for $T \rightarrow \infty$ (with T representing time).

Stability demands that the impact of a change only depends on the nature of the change itself. Conversely, changes causing impacts that are dependent on the size (or amount of changed or added requirements) of the system, are called *combinatorial* effects. To achieve stability, combinatorial effects should be abolished from the system. Systems that exhibit stability are defined as *Normalized Systems*. Stability

 ISBN: 978-1-61208-184-7

can be seen as the requirement of a linear relation between the cumulative changes and the growing size of the system over time. Combinatorial effects or instabilities cause this relation to become exponential (Figure 1). By eliminating combinatorial effects, this relation can be kept linear for an unlimited period of time, and an unlimited amount of (anticipated) changes to the system.

B. Design Theorems for Normalized Systems

Anticipating all the desired changes of the future might seem as a rather daunting task. Indeed, lots of system analysts get lost in this ambition. The authors of Normalized System's theory do not state they can do better in listing up all the functional requirements, possibly hidden in the present, or desired in the future. Fulfilling this task would be very complex and exceptional. These authors want to introduce another approach to achieve the same goal. The discussion about *anticipated changes* is not about changes, which are directly associated to recently expressed desires of the customers or managers to improve a system. Instead, anticipated changes focus on *elementary* changes, associated to software primitives. Typically, one real-life change corresponds with a lot of elementary changes, expressed in terms of software primitives. The Normalized System's theory is not focussing on the real-life changes, but on the elementary changes. In this section, we give an overview of the design theorems or principles of Normalized Systems, i.e., systems that are stable with respect to a defined set of anticipated (elementary) changes:

- A new version of a data entity;
- An additional data entity;
- A new version of an action entity;
- An additional action entity.

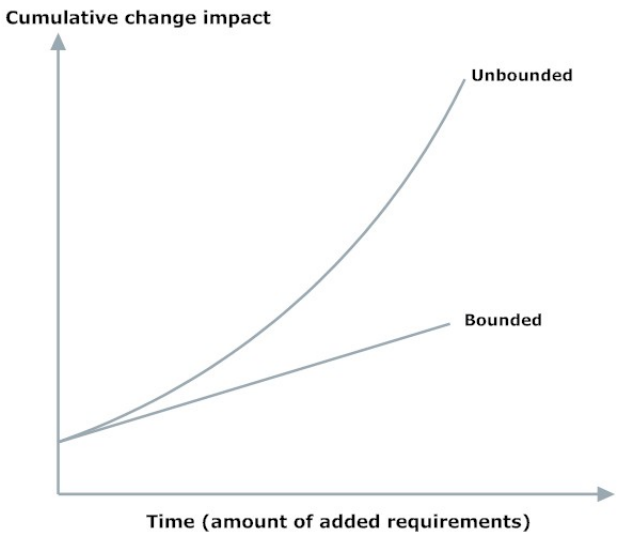


Figure 1. Cumulative impact over time

These changes are associated with software primitives in their most elementary form. Changes to meet "high-level requirements" that are obtained by system analysts from traditional gathering techniques (including interviews and use cases) [12] should be converted to these abstract, elementary anticipated changes. We were able to convert all high level changes in several case studies to one or more of these abstract anticipated changes [1][13][14]. However, some issues we encountered during the implementation of these proof of principles [14], have led to the introduction of the two new theorems of the next section. The systematic transformation of real-life requirements to the elementary anticipated changes is outside the scope of this paper. Note that already initial efforts are done in mapping the organizational requirements to the primitives of Normalized Systems [15].

1) Separation of concerns:

An action entity can only contain a single task in Normalized Systems.

This principle is focussing on how tasks are implemented within processing functions. Every *concern* or task has to be separated from other concerns. The identification of a task should be based on the concept of change drivers. A task is something that is subject to an independent change. A single change driver corresponds to a single concern in the application.

Proof: Consider a module M containing a task A and a second task B. The evolution of task B causes the introduction of N versions of task B. Since task B is part of module M, module M has also N versions. The introduction of a mandatory version upgrade of the task A will require to upgrade all N versions. According to the assumption of unlimited systems evolution, N will increase over time and will become unbounded, and so will the number of versions of task B. As a result, the number of additional version upgrades of the module M to implement a given change becomes unbounded.

2) Data version transparency:

Data entities that are received as input or produced as output by action entities, need to exhibit version transparency in Normalized Systems.

This principle is focussing on how data structures are passed to processing functions. Data version transparency is the property that data entities can have multiple versions, without affecting the processing functions that consume or produce them.

Proof: Consider a data structure D, that is passed through the interfaces of N versions of a processing function F. The introduction of a mandatory upgrade of the data structure D will require the adaptation of the code that accesses this data structure for N versions of F, unless D exhibits version transparency. According to the

 ISBN: 978-1-61208-184-7

assumption of unlimited systems evolution, N will increase over time and will become unbounded, and so will the number of versions of the processing function F. As a result, the number of additional adaptations of the code, which interfaces with the data structure, becomes unbounded.

3) Action version transparency:

Action entities that are called by other action entities, need to exhibit version transparency in Normalized Systems.

This principle is focussing on how processing functions are called by other processing functions. Action version transparency is the property that action entities can have multiple versions without affecting any of the other processing functions that call this processing function.

Proof: Consider a processing function P that is called by N other processing functions F. The introduction of an upgrade of P will require the adaptation of the code that calls P in the N functions F, unless the upgrade of function P exhibits version transparency. According to the assumption of unlimited systems evolution, N will increase over time and will become unbounded, and so will the number of calling functions F. As a result, the number of additional adaptations of the code, which are calling the function P, becomes unbounded.

4) Separation of states: The calling of an action entity by another action entity needs to exhibit state keeping in Normalized Systems.

This principle is focussing on how calls between processing functions are handled. The contribution of state keeping to stability is based on the removal of coupling between modules that is due to errors or exceptions. The (error) state should be kept in a separate data entity.

Proof: Consider a processing function P that is called by N other processing functions F. The introduction of an upgrade of P, possibly with a new error state, will require the N functions F to handle this error, unless the upgrade of function P exhibits state keeping. According to the assumption of unlimited systems evolution, N will increase over time and will become unbounded, and so will the number of calling functions F. As a result, the number of additional code to handle the new error in each function F, becomes unbounded.

IV. New theorems: entity instances

Modularity is a central concept in systems theory and has played a crucial role in software engineering since the 1960s. Doug McIlroy described a vision of the future of software engineering in which software would be assembled instead of programmed [16]. Studying evolvability of software in terms of its modular structure is widely accepted [1] and modularity is generally associated with use and reuse. Hence, when a module is used more than once during run-time we can call each use an *instance*.

Regev et al. proposed a definition of "Business Process Flexibility" [17]. We derive from this definition a more general interpretation of flexibility-to-change: "the capability to implement changes in a module's type and instances by changing only those parts that need to be changed and keeping other parts unchanged". In this interpretation, we specifically mean that a type change has influence on all upcoming instance creations. Meanwhile the existing (older) instances are not aware of this change, and should not be affected by this change. In other words, we consider the following sequence. An original version of a module's type is compiled on moment t=1. An instantiation of this module is created during the launch of the system on moment t=2. On moment t=3 we compile a new version of the module's type. We start a new part of the system, which is realizing a connection with an existing part. More specifically, on moment t=4 we create a new module instance based on the new version, without affecting the existing original module's instance in run-time (which existed since moment t=2). The instance, which showed up in run-time on moment t=4, should not affect the older instance, which was already launched in run-time on moment t=2.

For some instances of software primitives there are specific reasons to evolve, for other instances there are no such reasons. Moreover, even other instances have reasons to evolve on another way because of other specific (application dependent) reasons. Finally, we end up with an additional non-functional requirement, which can be designed on a similar way as evolvability: *support of diversity*. However, when initially identical instances of primitives evolve to a diversity of instances, the four theorems of Normalized Systems are not enough to prescribe how to manage instances. In search for a solution, we introduce two additional theorems, which focus on entity instances and how different versions can co-exist and used. Note that these new theorems are an *extension* of the theory. However, the prediction of the orignal authors of the existing Normalized Systems theory is not violated: these additions do not fundamentally alter the first 4 principles, they only suggest additional principles [7]. Moreover, the new theorems are *run-time equivalents* of the original theorems 2 and 3 (version transparency theorems). Besides, these existing theorems are proven by a simple *reductio ad adsurdum*, and the new ones are proven by evaluating a possible violation of the original theorems. Future research should provide experience reports in order to find possible empirical confirmations.

We define two additional anticipated changes:

- An additional data entity instance (known entity type version)
- An additional action entity instance (known entity type version)

 ISBN: 978-1-61208-184-7

We further posit the *assumption of unlimited systems evolution in both compile-time and run-time*, namely that the system evolves for an infinite amount of time. Note that the run-time evolution of the system is more complex than the compile-time evolution, because the run-time evolution also includes old instances, from which no module's type definitions exist any more in upcoming compile-time. One can imagine situations where new versions are just extending older version, and consequently the existence of the older module's type stops. Equivalently, the amount of both data entity instances and action entity instances will become (theoretically) unbounded. The amount of versions will become unbounded as well.

5) Data instance transparency: A data instance has to keep its own instance ID and the version ID on which it is based or constructed.

Proof (reductio ad adsurdum): When a data entity instance, constructed or generated with an old version is showing up in a more recent action entity instance, the action entity instance has to be able to decide whether it can
a) just process the data instance
b) use default values for non-existing fields
c) not process the request, but setting a 'data type obsolete' status.
Consider this action entity instance, which can not identify the version of the older data instance. Next, this action entity instance attempts consuming a non-existing field in the older data entity instance, and end up in an unexpected and/or stateless behavior. The latter would result in a violation of the fourth theorem (separation of states), and thus also of the postulate of Normalized Systems theory.

This principle is focussing on the interaction between data entity instances and action entity instances. It contributes to the problem that instances of the same data entity can differ in version. When a recent action entity meets an older data entity instance, this action has to know the version of the provided data instance, to be able to treat this instance in a correct way. It should for example not happen that the action entity is performing operations with a recently added data field, which is not available (yet?) in the provided data instance.

6) Action instance transparency: An action instance has to keep its own instance ID and the version ID on which it is based or constructed, preferably in a separate data entity instance

Proof (reductio ad adsurdum): Consider an action entity instance, which calls another action entity instance, without being able to identify the version of the called action entity instance. Next, the calling action entity instance requests to perform a non-existing (supporting) task of the called action entity instance, without being able to determine a correct state of the called operation. This would result in a violation of the fourth theorem (separation of states), and thus also of the postulate of Normalized Systems theory.

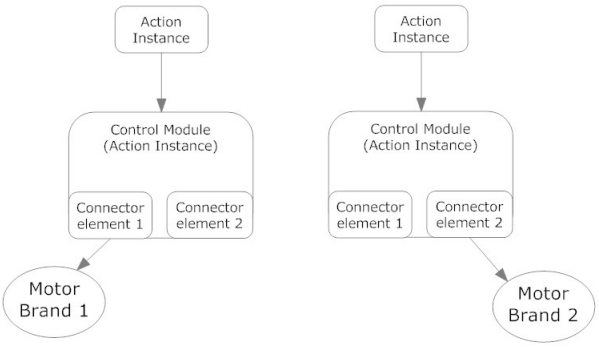


Figure 2. Two instances for different brands

This principle is focussing on the mutual interaction between action entity instances. It contributes to the problem that action instances of different versions are calling other action instances of different versions. The youngest instance has to be able to determine the version of the older instance, to be able to make a good decision concerning the operation of the requested functionality. Consider for example a control module, which is managing the control of a motor [13]. Suppose an existing motor instance is replaced by a motor of another brand, with different system functions. A new version of the control module would support a new connector element, which is managing the connection with the motor of the new brand (Figure 2). To comply to the theorem 'separation of concerns', a new connector element has to be added to the control module. However, besides the property of evolvability, support for diversity is needed too in this case. A calling action has to be able to specify and select which connector should be used in the operation. If, due to the absence of action instance transparency , the wrong (old) connector element is used, we will end up in unexpected behavior of the motor of the new brand.

V. Conclusion

Parts of a system, which are initially identical, will differ over time. In other words, some parts will evolve, others not (on the same way). Our ambition is to find rules to facilitate building systems, which are able to support both evolvability *and* diversity.

The Normalized Systems theory contributed in making heuristic knowledge explicit. We think the necessary knowledge for building evolvable systems is available, but often only in the form of tacit knowledge, which is often fragmented design knowledge [10]. The use of this tacit knowledge is very dependent of individuals, or of which individuals are supporting and coaching the development

 ISBN: 978-1-61208-184-7

process. For large systems, it is hardly manageable to only allow developers in the team, who have the required tacit knowledge. Making this tacit knowledge explicit can contribute in facilitating non-experienced engineers to develop evolvable systems.

This paper proposes two additional new theorems, which are in line with the existing theorems. In this contribution we focussed on the existing theorems "data version transparency" and "action version transparency". We specify that these original theorems apply explicitly for written code of data entity and action entities respectively. We suggest to interpret them as "data *type* version transparency" and "action *type* version transparency". Our two new theormes apply explicitly for *instances* during run-time for data entities and action entities. Therefore we call them "data instance transparency" and "action instance transparency".

REFERENCES

[1] Mannaert H., Verelst J., and Ven K., "Towards evolvable software architectures based on systems theoretic stability", Software, Practice and Experience, vol. 41, 2011.

[2] Eick S.G., Graves T.L., Karr A.F., Marron J., and Mockus A., "Does code decay? Assessing the evidence from change management data", IEEE Transactions on Software Engineering, vol 32(5), pp. 315-329, 2006.

[3] Amin S. M. and Wollenberg B. F. , "Towards a smart grid", IEEE Power and Energy Magazine, vol. 3, no. 5, pp. 34- 41, 2005.

[4] Kuhl I. and Fay A., "A Middleware for Software Evolution of Automation Software", IEEE Conference on Emerging Technologies and Factory Automation, 2011.

[5] Java platform enterprise edition. http://java.sun.com/javaee/.

[6] International Electrotechnical Commission, "IEC 61131-3, Programmable controllers - part 3: Programming languages", 2003.

[7] Mannaert H. and Verelst J., "Normalized Systems Re-creating Information Technology Based on Laws for Software Evolvability", Koppa, 2009.

[8] Dijkstra E., "Go to statement considered harmful", Communications of the ACM 11(3), 147-148, 1968.

[9] Mahnke W., Gössling A., and Graube M., "Information Modeling for Middleware in Automation", IEEE Conference on Emerging Technologies and Factory Automation, 2011.

[10] Mannaert H., Verelst J., and Ven K., "Exploring the Concept of Systems Theoretic Stability as a Starting Point for a Unified Theory on Software Engineering", IARIA ICSEA 2008.

[11] Lehman M.M., "Programs, life cycles, and laws of software evolution", Proceedings of the IEEE, Vol 68, pp. 1060-1076, 1980.

[12] Mannaert H., Verelst J., and Ven K., "The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability", Science of Computer Programming, 2010.

[13] van der Linden D., Mannaert H., and de Laet J., "Towards evolvable Control Modules in an industrial production process", $6th$ International Conference on Internet and Web Applications and Services, pp. 112-117, 2011.

[14] van der Linden D., Mannaert H., Kastner W., Vanderputten V., Peremans H. and Verelst J.,"An OPC UA Interface for an Evolvable ISA88 Control Module", IEEE Conference on Emerging Technologies and Factory Automation, 2011.

[15] van Nuffel D., Mannaert H., de Backer C., and Verelst J., "Towards a deterministic business process modelling method based on normalized theory", International journal on advances in software, 3:1/2, pp. 54-69, 2010.

[16] McIlroy M.D., "Mass produced software components", NATO Conference on Software Engineering, Scientific Affairs Division, 1968.

[17] Regev G., Soffer P., and Schmidt R., "Taxonomy of Flexibility in Business Processes", Proceedings of the 7th Workshop on Business Process Modelling, Development and Support, pp. 90-93, 2006.

 ISBN: 978-1-61208-184-7

Towards the Explicitation of Hidden Dependencies in the Module Interface

Dirk van der Linden
Electro Mechanics Research Group
Artesis University College of Antwerp
Antwerp, Belgium
dirk.vanderlinden@artesis.be

Herwig Mannaert, Peter De Bruyn
Department of Management Information Systems
University of Antwerp
Antwerp, Belgium
herwig.mannaert, peter.debruyn@ua.ac.be

***Abstract*—Balancing between the desire for information-hiding and the risk of introducing undesired hidden dependencies is often not straightforward. Hiding important parts of the internal functionality of a module is known as the *black box* principle, and is associated with the property of reusability and consequently evolvability. An interface, which is roughly explaining the core functionality of a module, helps indeed the developer to use the functionality without being forced to concentrate on the implementation details. However, some implementation details should not be hidden if they hinder the use of the module when the environment changes. These kind of implementation details can be called undesired hidden dependencies. An interesting question then becomes, which information should be hidden and which not? In this paper, we use the Normalized Systems theorems as a base to evaluate which details should be hidden versus transparent in order to improve reusability. In other words, which kind of information encapsulation contributes towards safe black box reuse?**

***Keywords*-Normalized Systems; Reusability; Evolvability; Systems Theory; Modularity; Black Box.**

I. INTRODUCTION

Modern technologies provide us capabilities to build large, compact, powerful and complex systems. Without any doubt, one of the major key points is the concept of modularity. Systems are built as structured aggregations of lower-level subsystems, each of which have precisely defined interfaces and characteristics. In hardware for instance, a USB memory stick can be considered a module. The user of the memory stick only needs to know its interface, not its internal details, in order to connect it to a computer. In software, balancing between the desire for information-hiding and the risk of introducing undesired hidden dependencies is often not straightforward. Experience contributes in learning how to deal with this issue. In other words, best practices are rather derived from heuristic knowledge than based on a clear, unambiguous theory.

Normalized Systems theory has recently been proposed [1] to contribute in translating this heuristic knowledge into explicit design rules for modularity. In this paper, we want to evaluate which information-hiding is desired and which is not with regard to the theorems of Normalized Systems.

The authors of this paper have each a different implementation focus (business process software versus automation control software), with different programming languages and development environments (JAVA [2] versus IEC 61131-3 [3]). In this collaboration we want to study fundamental principles, which should be independent of implementation focus. With regard to this independence, the different implementation focus of the authors might be an advantage. Moreover, at some point the need for combining these disciplines is arising. Automation systems have to be upgraded to new communication protocols and to provide new processing rules, as the interconnection of different grids will be forced in future [4].

Doug McIlroy called for *families of routines to be constructed on rational principles so that families fit together as building blocks. In short, [the user] should be able safely to regard components as black boxes* [5]. Decades after the publication of this vision, we have black boxes, but it is still difficult to guarantee that users can use them safely. However, we believe that all necessary knowledge is available, we only have to find all the necessary unambiguous rules to make this (partly tacit) knowledge explicit.

Scientific research groups contribute in converting tacit knowledge to theorems and fundamental rules, like the authors of Normalized Systems did. In addition, industrial working groups contribute in converting tacit knowledge to standards and specifications. For example, the OPC UA working groups provide the concept of OPC UA profiles. Profiles define the functionality of an OPC UA application [6]. Software Certificates contain information about the supported Profiles. OPC UA Clients and Servers can exchange these certificates via services.

The paper is structured as follows. In Section II, the Normalized Systems theory will be discussed. In Section lII, we give an overview of the most commonly discussed kinds of coupling, and evaluate whether they comply with the Normalized Systems theorems or not. In Section lV, we make suggestions on how we should deal with undesired hidded dependencies. Finally, conclusions and future research are discussed in Section V.

II. NORMALIZED SYSTEMS

The current generation of systems faces many challenges, but arguable the most important one is evolvability [7]. The

 ISBN: 978-1-61208-184-7

evolvability issue of a system is the result of the existence of Lehman's Law of Increasing Complexity which states: "As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it" [8]. Starting from the concept of systems theoretic stability, the Normalized Systems theory is developed to contribute towards building systems, which are immune against Lehman's Law.

A. Stability

The postulate of Normalized Systems states that *a system needs to be stable with respect to a defined set of anticipated changes*. In systems theory, one of the most fundamental properties of a system is its stability: a bounded input function results in bounded output values, even for $T \rightarrow \infty$ (with T representing time).

The impact of a change should only depend on the nature of the change itself. Systems, built following this rule can be called stable systems. In the opposite case, changes causing impacts that are dependent on the size of the system, are called *combinatorial effects*. To attain stability, these combinatorial effects should be removed from the system. Systems that exhibit stability are defined as *Normalized Systems*. Stability can be seen as the requirement of a linear relation between the cumulative changes and the growing size of the system over time. Combinatorial effects or instabilities cause this relation to become exponential (Figure 1). The design theorems for Normalized Systems contribute to the long term goal of keeping this relation linear for an unlimited period of time, and an unlimited amount of changes to the system.

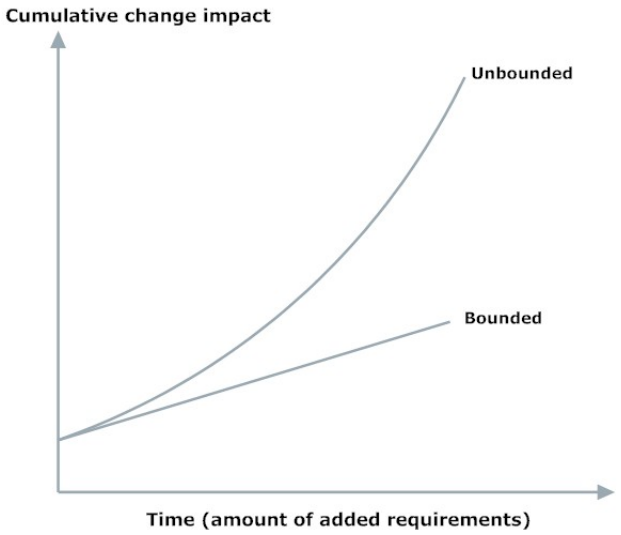


Figure 1. Cumulative impact over time

B. Design Theorems for Normalized Systems

In this section, we give an overview of the design theorems or principles of Normalized Systems, i.e., systems that are stable with respect to a defined set of anticipated changes:

- A new version of a data entity
- An additional data entity
- A new version of an action entity
- An additional action entity

Please note that these changes are associated with software primitives in their most elementary form. Real-life changes or changes with regard to 'high-level requirements' [9] should be converted to these abstract, elementary anticipated changes. We were able to convert all real-life changes in several case studies to one or more of these abstract anticipated changes [10][11][12]. However, the systematic transformation of real-life requirements to the elementary anticipated changes is outside the scope of this paper.

1) Separation of concerns:

An Action Entity can only contain a single task in Normalized Systems.

In this theorem, we focus on how tasks are structured within processing functions. Each set of functionality, which is expected to evolve or change indepently, is defined as a change driver. Change drivers are introducing anticipated changes into the system over time. The identification of a task should be based on these change drivers. A single change driver corresponds to a single *concern* in the application.

2) Data version transparency:

Data Entities that are received as input or produced as output by Action Entities, need to exhibit version transparency in Normalized Systems.

In this theorem, we focus on how data structures are passed to processing functions. Data structures or *Data Entities* need to be able to have multiple versions, without affecting the processing functions that use them. In other words, Data Entities having the property of data version transparency, can evolve without requiring a change of the interface of the action entities, which are consuming or producing them.

3) Action version transparency:

Action Entities that are called by other Action Entities, need to exhibit version transparency in Normalized Systems.

In this theorem, we focus on how processing functions are called by other processing functions. Action Entities need to be able to have multiple versions without affecting any of the other Action Entities that call them. In other words, Action Entities having the property of action

 ISBN: 978-1-61208-184-7

version transparency, can evolve without requiring a change of one or more Action Entities, which are connected to them.

4) Separation of states: The calling of an Action Entity by another Action Entity needs to exhibit state keeping in Normalized Systems.

In this theorem, we focus on how calls between processing functions are handled. Coupling between modules, that is due to errors or exceptions, should be removed from the system to attain stability. This kind of coupling can be removed by exhibiting state keeping. The (error) state should be kept in a separate Data Entity.

III. Evaluation of kinds of coupling

Coupling is a measure for the dependencies between modules. Good design is associated with low coupling and better reusability. However, lowering the coupling only is not specific enough to guarantee reusability. Classifications of types of coupling were proposed in the context of structured design [13]. The key question of this paper is whether a hidden dependency and therefore coupling is affecting the reusability of a module? In general, the Normalized Systems theorems identify places in the software architecture where high coupling is threatening evolvability [14]. More specifically, we will focus in this paper on several kinds of coupling and evaluate which of them is lowering or improving reusability.

A. Content coupling

Content coupling occurs when module A refers directly to the content of module B. More specifically, this means that module A changes instructions or data of module B. When module A branches to instructions of module B, this is also considered as content coupling.

It is trivial that direct references between modules prevent them from being reused separately. In terms of Normalized Systems, content coupling is a violation of the first theorem, separation of concerns.

Avoiding content coupling is not new, other rules than those of the Normalized Systems already made this clear. Decades ago, Dijkstra suggested to abolish the Go To statement from all 'higher level' programming languages [15]. Together with restricting access to the memory space of other modules, Dijkstra's suggestion contributed to exile content coupling out of most modern programming languages.

B. Common coupling

Common coupling occurs when modules *communicate using global variables*. A global variable is accessible by all modules in the system. If a developer wants to reuse a module, analyzing the code of the module to determine which global variables are used is needed. In other words, a white box view is required. Consequently, black box use is not possible.

In terms of Normalized Systems, common coupling is a violation of the first theorem, separation of concerns.

We add however, that not the *existence* but the *way of use* of global variables violates the separation of concerns theorem. In earlier work we used global variables in a proof of principle with IEC 61131-3 code, which complies with Normalized Systems [11]. The existence of global variables was needed for other reasons than mutual communication between modules (i.e., connections with process hardware). In this project, the global variables were passed via an interface from one module to the other. Some authors state that the declaration of global variables in the IEC 61131-3 environment is somewhat ambiguous [16], although we do not think this is crucial to determine the characteristic of reusability of a module (as long as there is no common coupling!).

Since the use of global variables in case of common coupling is not visible through the module's interface, each use of these global variables is considered to be a hidden dependency. And since common coupling is a violation of separation of concerns, this is an undesired hidden dependency with respect to the safe use of black boxes.

C. Control coupling

Control coupling occurs when module A influences the execution of module B by passing data (parameters). Commonly, such parameters are called flags. Whether a module with such a flag can be used as a black box depends on the fact whether the interface is explaining sufficiently the meaning of this flag for use. Obviously, if a white box view is necessary to determine how to use the flag, black box use is not possible. The evaluation of control coupling in terms of reusability is twofold. Adding a flag can introduce a slightly different functionality and improve the reuse potential. For example, if a control module of a motor is supposed to control pumping until a level switch is reached, a flag can provide the flexibility to use both a positive level switch signal and an inverted one (positive versus negative logic). On the other hand, extending this approach to highly generic functions, would lead in its ultimate form to a single function *doIt*, that would implement all conceivable functionality, and select the appropriate functionality based on arguments. Obviously, the latter would not hit the spot of reusability.

One of the key questions during the evaluation of control coupling is: how many functionalities should be hosted in one module? In terms of Normalized Systems, the principle 'separation of concerns' should not be violated. The concept of change drivers brings clarity here. A module should contain only one core task, eventually surrounded by supporting tasks. Control coupling can help to realize theorem 2 (data version transparency) and theorem 3 (action version

 ISBN: 978-1-61208-184-7

transparency). The calling action is able to select a version of the called action based on control coupling. We conclude that *control coupling should be used for version selection only*. More details about versions and their instances are part of very recent research of which the results are in a review process on the moment of the submissing of this paper.

D. Stamp coupling

Stamp coupling occurs when module A calls module B by passing a data structure as a parameter when module B does not require all the fields in the data structure.

It could be argued that using a data structure limits the reuse to other systems where this data structure exists, whereas only sending the required variables separately does not impose this constraint. However, we emphasize that the key point of this paper does not concern *reuse* in general. Rather, it focuses on *safe reuse* specifically. The research towards safe black box reuse is more about adding contraints or defining limitations than keeping or creating possibilities. When working with separated, simple datatypes as a set of parameters, every change requires a change of the interface of the module. Since we do not consider 'changing the interface' as one of our anticipated changes, stamp coupling is an acceptable form of coupling. With regard to the first theorem, separation of concerns, one should keep the parameter set (Data Entity), the functionality of the module (action) and the interface separated. Keeping the interface unaffected, while the Data Entity and Action Entity are changing, can be realized with stamp coupling.

E. Data coupling

Data coupling occurs when two modules pass data using simple data types (no data structures), and every parameter is used in both modules.

Realizing theorem 3 (action version transparency) is impossible with data coupling, since the introduction of a new parameter affects the interface of the module. This newer version of the interface would not be suitable for previous action versions, and could consequently not be called a version transparent update. The addition of a parameter in the module's interface would violate the separation of concerns principle. Changing or removing a parameter is even worse.

Note that the disadvantage of data coupling, affecting the module's interface in case of a change, does not apply on reusing modules, which are not evolving. This can be the case when working with system functions, e.g., aggregated in a system function library. However, problems can occur when the library is updated. We will give more details about this issue in the next section.

F. Message coupling

Message coupling occurs when communication between two or more modules is done via message passing. Message passing systems have been called 'shared nothing' systems because the message passing abstraction hides underlying state changes that may be used in the implementation of sending messages.

The property 'sharing nothing' makes message coupling a very good incarnation of the separation of concerns principle. Please note that asynchronous message passing is highly preferable above synchronous message passing, which violates the separation of states principle.

IV. Implicit Dependencies

We consider the case of one developer, who has programmed a part of a modular system with an acceptable amount of coupling. Every entity of the *subsystem* complies to the theorems of Normalized Systems. The code is fulfilling a *part* of the requirements of a bigger system, which has to be built by several developers. Another part of the system is programmed by a second developer, using the same programming language and the same platform. The question is, can both developers exchange the source code of their modules and reuse them as *safe black boxes*, i.e., without the need of a white box view of their colleagues' code? In this paper, we concentrate on the case where both developers used the same programming language, the same platform, but a different set-up of programming environment. In other words, they do not share the same - company standard - system functions in their respective programming environments.

We start our discussion with a second case, where two or more hardware developers share several ICs (Integrated Circuits) to build for example embedded systems. Can they just pick an IC out of the box and safely use it as a black box? On one hand, hardware engineers did a remarkable job with regard to the production of safe black boxes, because they do not need to know the internal details of the IC to use it. However, before usage, the user needs the information *that is printed on the IC* to estimate its expected behaviour. If the IC is very recently built as a prototype, with nothing printed on it, the user needs information of the prototype-builder to know how to use it. In other words, information about the interface has to be available in order to use the black box. Besides, the information — rarely more than a type number — which is typically printed on an IC, is referring to data sheets, explaining in more detail the black box use of the (hardware) module.

In software, we have the advantage that interfaces can be made roughly self-explaining. But in comparison with hardware, possible dependencies are introduced. Our two or more software developers who want to (re)use each others modules, made in another programming environment, should inform each other about the libraries they used. But even if they do this well, they might end up in a so-called dependency-hell. This is a colloquial term for the frustration of some software users who have installed

 ISBN: 978-1-61208-184-7

software packages, which depend on specific versions of other software packages. It involves for example package A needing package B & C, and package B needing package F, while package C is incompatible with package F.

This kind of compatibility conflicts is — in terms of Normalized Systems — caused by a violation of the first theorem, separation of concerns. Every external technology should be separated by a connector element. Note that a connector element is an Action Entity dedicated to connect a module with an external technology. To prevent version conflicts, every connector element should exhibit version transparency, like every other Action Entity should.

Let us consider the situation in which highly qualified software developers indeed implemented connector elements for every package or library they used. This makes the integrated system robust against anticipated changes, although it is obviously not delivering any garantee regarding the proper functioning of that external technology itself. In other words, if one or more packages or libraries are not properly linked to the development environment, the connector elements will deliver (on an asynchronous way) an error status.

Safe black box (re)use includes that a developer should be able to anticipate which conditions are necessary for (re)use. A self-explaining interface is a good start, but typically dependencies like packages or libraries are not included in the interface. We conclude that it should, and phrase the following rule.

In order to design safe black box (re)useable software components, every (re)use of a library or package in a module, should include a reference, path or link to the identification of the dependency, accompanied with the used version.

We make the reflection that there is a similarity between global variables and dependencies, which are not passed through the module's interface. Consequently, these dependencies cause common coupling. Remember it is not the *existence* of global variables, libraries or packages which is causing common coupling, but rather the fact that these variables, libraries or packages are *not passed via the module's interface*. As a result, these kind of dependencies violate the separation of concerns principle.

In searching ways to identify dependencies through the module's interface, the concept of OPC UA Profiles is interesting [17]. OPC UA Profiles define the functionality of an OPC UA application. As human-readable announcements, they inform users which parts of the OPC UA standard are implemented. In addition, this information can also be exchanged between OPC UA applications. This allows applications to accept or reject connection requests depending on which Profiles their counterpart is supporting. The concept of OPC UA Profiles is dedicated to exchange information about the interface and communication concepts of OPC UA applications. The functionality behind such an OPC UA interface is not exchangeable via OPC UA Profiles. In other words, the functionality, which can be expressed in standardized OPC UA Profiles, is limited to interface and communication functionality. This principle should be extended to a more general form to provide information about dependencies on different levels. For modules, directly connected to the internet, a worldwide accessible website could provide standardized information about well defined dependencies. For other modules, the same concept of reference could be introduced for specific application domains, or even vendor-dependent dependency information.

Remember the case of hardware engineers willing to share ICs for the development of embedded systems. The code printed on the ICs is referring to data sheets. This situation is similar to software modules, accompanied with a reference to dependency information in their interface. Whenever a user can not find the dependency information through a reference in the interface of a black box, it should be possible to reject the possible use of this module. This would result in an enforcement of the separation of states principle.

V. Conclusion

The reasons why properties like evolvability, (re)usability and safe black box design are difficult to achieve, have most likely something to do with a lack of making the existing knowledge and experience-based guidelines explicit. Undoubtedly, the theorems of Normalized Systems contribute on this issue by formulating unambiguous design rules at the elementary level of software primitives. However, on a higher implementation level, it is expected that not all implementation questions like those related to e.g., a dependency-hell, are easy to answer. Experienced engineers will find that these are violations of the theorems 'separation of concerns' and 'separation of states'. However, we aim that — on top of these fundamental principles — some derived rules can make these violations easier to catch, also for less experienced engineers.

In this paper we introduced the derived rule that, based on the 1st and 4th principle of Normalized Systems, any dependency should be visable in the module's interface, accompanied by its state and version. Of course, the way how this information is included in the interface, should be done in a version transparent way, to prevent violations of the 2nd and 3rd principle of Normalized Systems.

Acknowledgment

P.D.B. is supported by a Research Grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[1] Mannaert H. and Verelst J., "Normalized Systems Re-creating Information Technology Based on Laws for Software Evolvability", Koppa, 2009.

 ISBN: 978-1-61208-184-7

[2] Java platform enterprise edition. http://java.sun.com/javaee/.

[3] International Electrotechnical Commission, “IEC 61131-3, Programmable controllers - part 3: Programming languages”, 2003.

[4] Kuhl I. and Fay A., “A Middleware for Software Evolution of Automation Software”, IEEE Conference on Emerging Technologies and Factory Automation, 2011.

[5] McIlroy M.D., “Mass produced software components”, NATO Conference on Software Engineering, Scientific Affairs Division, 1968.

[6] Mahnke W., Leitner S., and Damm M., “OPC Unified Architecture”, Springer, 2009.

[7] Mannaert H., Verelst J., and Ven K., “Exploring the Concept of Systems Theoretic Stability as a Starting Point for a Unified Theory on Software Engineering”, IARIA ICSEA 2008.

[8] Lehman M.M., “Programs, life cycles, and laws of software evolution”, Proceedings of the IEEE, Vol 68, pp. 1060-1076, 1980.

[9] Mannaert H., Verelst J., and Ven K., “The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability”, Science of Computer Programming, 2010.

[10] Mannaert H., Verelst J., and Ven K., “Towards evolvable software architectures based on systems theoretic stability”, Software, Practice and Experience, vol. 41, 2011.

[11] van der Linden D., Mannaert H., and de Laet J., “Towards evolvable Control Modules in an industrial production process”, $6th$ International Conference on Internet and Web Applications and Services, pp. 112-117, 2011.

[12] van der Linden D., Mannaert H., Kastner W., Vanderputten V., Peremans H. and Verelst J.,“An OPC UA Interface for an Evolvable ISA88 Control Module”, IEEE Conference on Emerging Technologies and Factory Automation, 2011.

[13] Myers G., “Reliable Software through Composite Design”, Van Nostrand Reinhold Company, 1975.

[14] van Nuffel D., Mannaert H., de Backer C., and Verelst J., “Towards a deterministic business process modelling method based on normalized theory”, International journal on advances in software, 3:1/2, pp. 54-69, 2010.

[15] Dijkstra E., “Go to statement considered harmful”, Communications of the ACM 11(3), pp. 147-148, 1968.

[16] de Sousa M., “Proposed corrections to the IEC 61131-3 standard”, Computer Standards & Interfaces, pp. 312-320, 2010.

[17] OPC Foundation. “OPC Unified Architecture, Part7: Profiles”, Version 1.01, Draft 1, september 2010.

 ISBN: 978-1-61208-184-7

A Framework for Cyber-Physical Systems Design – A Concept Study

Ondrej Rysavy
Faculty of Information Technology
Brno University of Technology
Czech Republic
E-mail: rysavy@fit.vutbr.cz

Miroslav Sveda
Faculty of Information Technology
Brno University of Technology
Czech Republic
E-mail: sveda@fit.vutbr.cz

Radimir Vrba
Faculty of Electrical Eng.& Comm.
Brno University of Technology
Czech Republic
E-mail: vrbar@feec.vutbr.cz

Abstract — **The paper deals with principles of a launching research focused on cyber-physical systems (CPS) design environment. It refers to completed real-world CPS application projects aimed at smart data acquisition systems capable to store and present measured data wirelessly. The paper depicts a CPS design approach stemming from generic requirements on domain applications, reviews state of the art of the domain, mentions some experience with pilot projects and brings an outline of the intended CPS design framework.**

Keywords- Embedded system design, smart sensor, wireless communication, temperature and pressure measurement.

I. INTRODUCTION

The term Cyber-Physical Systems has come to describe the research and technological effort that will ultimately efficiently allow interlinking the real world physical objects and cyberspace. The integration of physical processes and computing is not new. Embedded systems have been in place since a long time to denote systems that combine physical processes with computing. The revolution is coming from communicating embedded computing devices that will allow instrumenting the physical world with pervasive networks of sensor-rich, embedded computation.

This paper deals with principles of a launching research focused on CPS design environment. It refers to completed real-world application projects aimed at smart data acquisition systems capable to store and present measured data wirelessly. The paper depicts a CPS design approach stemming from generic requirements on domain applications, state-of-the-art review, and discussing initial ideas on the proposed design and development tools interconnected to form a development environment.

II. REQUIREMENTS

The CPS research program aims to develop a unifying theory for the design and implementation of integrated cyber and physical resources that can be applied across multiple domains [12]: "... Currently, unrelated methods are used to separately develop cyber and physical subsystems. The differences between the two sides are manifest at the most fundamental levels: computer science builds upon discrete mathematics, whereas engineering is dominated by continuous mathematics. Even within the broad fields of engineering and computer science, multiple sub-disciplines use dissimilar concepts and tools. The lack of a unifying or composable theory makes it impossible to guarantee safety and performance by design. System validation requires extensive testing — an approach that is becoming intractable as systems become more complex. Despite progress in the development of increasingly more powerful technologies for networked embedded sensing and control, today's embedded computing systems are point solutions for specific applications. Current approach to hardware and software design, systems engineering and real-time control needs to be rethought in the unifying context of cyber-physical systems. For example, open, flexible, and extensible architectures for cyber-physical systems would enable and better influence advances in hardware and software components and subsystems. They should be analyzable and should support principled composition or integration. Run-time operation should exploit information-rich environments to enhance performance and reliability. In some applications, these systems should be context aware, with the ability to modify their behaviors to accommodate changing configurations, adapt to variations in the environment, sustain safe operation, and improve performance over time. For many applications, cyber-physical systems must be certifiable, i.e. new approaches are needed for the specification, verification, and validation of tightly integrated cyber and physical elements."

As is generally agreed, the effective design of cyber-physical systems requires research advances in methods and tools that support multiple views of integrated cyber and physical components. New programming languages are needed to handle complex interactions between cyber and physical resources and to deal with unstructured data and stringent requirements for responsiveness. Algorithms for reasoning about and formally verifying properties of complex integrations of cyber and physical resources are needed. Tools for implementing algorithms to support off-line and run-time optimization and control are also needed. Tools should support concurrent engineering of physical systems with sensing, communication, and control architectures. Methods and tools should enable new forms of analysis, testing, and validation of integrated discrete and continuous dynamics at multiple temporal and spatial scales and different levels of resolution. Tools should be open, interoperable, and highly expressive to enhance productivity and enable community use. They should also be extensible to leverage new results from the foundations research and

 ISBN: 978-1-61208-184-7

accommodate new technologies and capabilities as they become available.

III. STATE OF THE ART

Many of the embedded systems-related studies and efforts in the past have focused on the challenges the physical environment brings to the scientific foundations of networking and information technology, see [2] and [4]. However, the full scope of the change enabled by introducing CPS as a new branch of science and technology provides much more than restructuring inside this domain. The new approach can turn entire industrial sectors into producers of CPS. Actually, CPS is about merging computing and networking with physical systems to create new capabilities and improve product quality [11].

Cyber-physical systems denote a new modeling paradigm that promotes a holistic view on real-world – and therefore complex – systems. These systems have been studied before from various particular perspectives using paradigms like ubiquitous and distributed computing or embedded and hybrid systems. The above mentioned facts require also another approach to the design of such systems respecting from the beginning of design process the application domain that influences quality-of-service requirements such as real-time behavior, safety and security [17], [18], [14] and [15], but also precision, reliability and other non-functional properties affecting attributes specified usually by official standards [9].

In a CPS application, the function of a computation is defined by its effect on the physical world, which is in this case not only a system environment, but evidently also a component of the designed application system. Therefore, proper design environments should be used to improve or at least to enable efficiency of the design process. In cyber-physical systems the passage of time becomes a central feature — in fact, it is this key constraint that distinguishes these systems from distributed computing in general. Time is central to predicting, measuring, and controlling properties of the physical world: given a (deterministic) physical model, the initial state, the inputs, and the amount of time elapsed, one can compute the current state of the plant. This principle provides the foundations of control theory. However, for current mainstream programming paradigms, given the source code, the program's initial state, and the amount of time elapsed, we cannot reliably predict future program state. When that program is integrated into a system with physical dynamics, this makes principled design of the entire system difficult. Instead, engineers are stuck with a prototype-and-test style of design, which leads to brittle systems that do not easily evolve to handle small changes in operating conditions and hardware platforms. Moreover, the disparity between the dynamics of the physical plant and the program seeking to control it potentially leads to errors, some of which can be catastrophic.

IV. DESIGN FRAMEWORK

Perri and Kaiser [13] formulate a model of development environment employing three tool types: (1) structures in the role of reusable components embodied into developed systems; (2) mechanisms in the role of proper development tools used for development process but not included into developed systems, and (3) strategies as design and development methods. The design framework's concepts can stem from verifiable formal specifications of CPS as a launching paradigm and from reusability paradigm supporting all phases of the design process ranging from specification to implementation and testing.

The basic terms, excerpted originally from [24] and adapted according to [21] for computer-based systems and, particularly, for embedded systems application area purpose can be restated as follows:

- The environment is the portion of a real world relevant to the design project.
- The embedded system is a computer-based artifact that will be constructed and connected to the environment, as a result of the design project.
- A requirement is an embedded system's property intended to express the desires of the customer concerning the design project.
- A statement of domain knowledge is an environment's property intended to be relevant to the design project.
- A specification is an embedded system's property, intended to be directly implementable and to support satisfaction of the requirements.

Let S be the set of specifications, R be the set of requirements, and K be the relevant domain knowledge for a design project. Then S and K must be sufficient to guarantee that the requirements R are satisfied. The primary role of domain knowledge, K, is to bridge the gap between requirements, R, and specifications, S. Requirements that are not specifications are always converted into specifications with the help of domain knowledge. Application patterns, which embody domain knowledge, deal both with specifications and implementations.

It would seem, that the framework can be with no trouble reformulated for CPS and, after that, it can offer a starting point for specification and design. Evidently, such framework has to be refined to be useful for real applications. On the other hand, it should be general-enough to support broad application domains. The next section presents some experience based on completed CPS applications developed without special design environment, but demonstrating typical design cases.

V. LESSONS LEARNED

Starting with [23], [8], we collected some experience with designs of deeply imbedded CPS in frame of the following research outcomes: (1) mobile temperature data logger based on RFID system [17], (2) optoelectronic pressure and temperature sensory system based on dedicated Bluetooth network [17], and (3) optoelectronic pressure sensor system based on distributed architecture with Intranet TCP/IP [18]. After reviewing basic design concepts deployed, main attention is focused in all cases on the CPS artifacts' specification, design with respect to application

 ISBN: 978-1-61208-184-7

domain requirements, assembly, and appropriate communication services.

The paper [17] describes two CPS designs in more detail using two original research outcomes: mobile temperature data logger based on RFID system, and optoelectronic pressure sensory system based on dedicated Bluetooth network. The presented temperature data logger stands for an example of flexible, mobile and intelligent appliances fitting various industrial or medical applications. Similarly, the discussed sensor network represents a system architecture stemming from wireless smart pressure sensors connected by Bluetooth and from a network concentrator, which is based either on PDA personal digital assistant or on GSM SmartPhone.

The paper [18] describes a CPS example using an optoelectronic pressure sensor system based on distributed architecture with Internet/Intranet TCP/IP structure exploiting Ethernet 10/100 Mbps. After reviewing basic CPS concepts deployed, main attention is focused on a concrete optoelectronic pressure sensor design, assembly, and communication services in frame of the multi-sensor system fitting the application requirements. The networking configuration exemplifies in this case a real solution of a more complex networked embedded system application based on the IEEE 1451 family of standards and on actual software and hardware components developed by the authors and collaborators for a class of sensing and measurement embedded applications.

Both above mentioned papers strive to demonstrate application-driven designs of deeply embedded CPS cases that differ substantially in technology used. On the other hand, those cases enable also to identify typical commonalities in detailed CPS designs and, hence, to derive general requirements on design tools.

VI. CYBER-PHYSICAL SYSTEMS DESICN CONCEPTION

Design and development systems, see e.g. [4], [9], [5], [7], [22], have to support important concepts and methods by their tools for complete design and development life cycle of applications belonging to considered application domains. The toolset related to the discussed design framework will necessarily include also original methods and tools. At the beginning, the development means will target predominantly front-end parts of specification and design, namely formal specification, verification and rapid prototyping.

Conventional verification techniques to be used in the development environment have high memory requirements and are very computationally intensive. Therefore, they are unsuitable for real-world CPS systems that exhibit complex behaviors and cannot be efficiently handled unless we use scalable methods and techniques [20], which exploit fully the capabilities of new hardware architectures and software platforms [8]. High-performance verification techniques focus on increasing the amount of available computational power. These are, for example, techniques to fight memory limits with efficient utilization of external techniques that introduce cluster-based algorithms to employ aggregate power of network-interconnected computers, or techniques to speed-up the verification on multi-core processors.

Researching CPS models consist of capturing characteristics of CPS. We plan to study existing and to propose new models for common architectural and behavioral artifacts and communication patterns of the CPS domain.

To be more explicit, at the beginning we define models using Ptolemy II framework (see [14], [10]) extended by existing formal tools and we will study the possibility to integrate the formal verification methods for these hybrid models. It would require examining carefully the semantics bound in different models and define precise transformations to extract verifiable models from design models.

Domain specific modeling languages (DSML), contrary to the universal modeling languages, are specifically customized to the area of problems being solved. Using DSML approach, the modeling of a system is itself preceded by the phase of meta-modeling of the application domain. We plan to propose a DSML for the reliable real-time embedded devices in smart sensor and control networks domain and provide formal semantics for this language that should enable applications of formal methods for transformation and verification of CPS properties.

We will research possibility to apply existing formal methods to the models generated from the specifications written in CPS-DSML. The models describe the system being developed at different levels and views. Automated tools should support inter-model validation. Thus our primary concern is to demonstrate how tools based on formal methods can proof the inter-model consistency and property preservation. For instance, model of software components, which behavior is driven by discrete means of computing should be in consistency with lower level model of hardware processing units and also with same level model of abstract environment behavior. The difficulty and novelty lies in consideration that different models obey different means of computing.

Designed development environment prototype will include tools and methods that can be used to approach demonstration and experimenting with the selected application area. We assume that various methods will be experimentally implemented as software tools to show the capability of the approach on non-trivial use cases. New design patterns and components will be created and verified in frame of case studies. These case studies will serve to gather experience in development of CPS. The work should conclude by critical evaluation of the proposed approach, showing the strength aspects of considered method and revealing drawbacks that deserve further research.

VII. CONCLUSIONS

The paper deals with principles of a launching research focused on CPS design environment. It refers to completed real-world application projects aimed at smart data acquisition systems capable to store and present measured data wirelessly. The paper depicts a CPS design approach stemming from generic requirements on domain applications, continuing with brief review on state-of-the-art, and completed by initial ideas on the proposed design and

 ISBN: 978-1-61208-184-7

development tools interconnected to form a development environment.

This paper utilizes as model demonstrations three CPS designs rooted in original research outcomes: mobile temperature data logger based on RFID system, optoelectronic pressure sensory system based on dedicated Bluetooth network, and optoelectronic pressure sensory system based on Ethernet/IP/TCP network, published in more detail in frame of previous ICONS conferences.

Our research group is currently launching a related continuation research that aims at the formal tools support of CPS design [17], [18], [19]. Evidently, this new research domain requires not only formal specification and verification techniques extensions and modifications, but also novel approaches and adaptations of such general methods as model checking and proving, see e.g. [1], [2], [3], [9], [10], [16] and [22].

ACKNOWLEDGMENT

This project has been carried out with a financial support from the Czech Republic state budget through the *IT4Innovations Centre of Excellence*, EU, CZ 1.05/1.1.00/02.0070CEZ and through the MMT project no. MSM0021630528: *Security-Oriented Research in Information Technology*, by the Technological Agency of the Czech Republic through the grant no. TA01010632: *SCADA system for control and monitoring RT processes*, and by the Brno University of Technology, Faculty of Information Technology through the specific research grant no. FIT-S-11-1: *Advanced Secured, Reliable and Adaptive IT*. We also strive for the support by the Grant Agency of the Czech Republic through the grant proposal *Designing Cyber-Physical Systems.*

REFERENCES

[1] R. Akella and B.M. McMillin, Model-checking BNDC Properties in Cyber-Physical Systems, *Proceedings of the33rd INternational Computer Software and Applications Conference COMPSAC 2009*, IEEE CS, New York, NY, US, 2009, pp.660-663.

[2] B. Bonakdarpour, Challenges in Transformation of Existing Real-Time Embedded Systems to Cyber-Physical Systems *IEEE Symposium on Real-Time Systems RTSS RTSS - Ph.D. Forum on Deeply Real-Time Embedded Systems*, Tucson, Arizona, 2007, 2pp.

[3] M.C. Bujorianu and H.Barringer, An Integrated Specification Logic for Cyber-Physical Systems, *Proceedings of the 14th IEEE International Conference on Engineering of Complex Computer Systems*, Potsdam, Germany, 2009, pp.91-100.

[4] J. C. Eidson, E.A. Lee, S. Matic, S.A. Seshia and J. Zou, Time-centric Models For Designing Embedded Cyber-physical Systems, EECS Department, University of California, Berkeley, *Technical Report No. UCB/EECS-2009-135*, October 9, 2009.

[5] E.K. Jackson and J. Sztipanovits, Correct-ed through Construction: A Model-based Approach to Embedded Systems Reality. *Proceedings of the 13th Engineering of Computer-Based Systems*, IEEE Computer Society, Los Alamitos, CA, pp.164-173, 2006.

[6] J.E. Kim and Daniel Mosse, Generic framework for design, modeling and simulation of cyber physical systems, *SIGBED Review*, Vol. 5, No. 1, ACM, January 2008, 2pp.

[7] B.H. Krogh, E. Lee, I. Lee, A. Mok, R. Rajkumar, L.R. Sha, A.S. Vincentelli, K. Shin, J. Stankovic, J. Sztipanovits, W. Wolf and W. Zhao, *Cyber-Physical Systems, Executive Summary,* CPS Steering Group, Washington D.C., March 6, 2008. [http://www.nsf.gov/pubs/2008/nsf08611/nsf08611.htm]

[8] R. Kuchta, P. Steffan, Z. Barton, R Vrba and M. Sveda, Wireless Temperature Data Logger, *Proceedings of the 2005 Asian Conference on Sensors, and International Conference on new Techniques in Pharmaceutical and Biomedical Research*, 5-7 Sept. 2005, pp.208-212.

[9] E.A. Lee, Computing Needs Time, *Communications of the ACM*, Vol.52, No.5, pp.70-79, May 2009.

[10] E.A. Lee. *Finite State Machines and Modal Models in Ptolemy II*, Technical report, EECS Department, University of California, Berkeley, UCB/EECS-2009-151, December, 2009.

[11] E.A. Lee, CPS Foundations, *Proceedings of the DAC'10,* ACM, Anaheim, California, June 2010, pp.737-742.

[12] National Science Foundation, *Cyber-Physical Systems Program Solicitation, NSF 10-515*, Arlington, VA, US, March 11, 2010

[13] D.E. Perri and G.E. Kaiser, Models of Software Development Environment, *IEEE Transactions on Software Engineering*, Vol.17, 1991, pp.283-295.

[14] PtolemyII: http://ptolemy.berkeley.edu/ptolemyII

[15] L. Sha and J. Meseguer, Design of Complex Cyber Physical Systems with Formalized Architectural Patterns, *Software-Intensive Systems and New Computing Paradigms: Challenges and Visions,* Springer, 2008, pp 92-100.

[16] J.A. Stankovic, I. Lee, A. Mok and R. Rajkumar, Opportunities and obligations for physical computing systems, *IEEE Computer*, November 2005*,* pp.23-31.

[17] M. Sveda and R. Vrba, A Cyber-Physical System Design Approach, *Proceedings of The Sixth International Conference on Systems - ICONS 2011*, St. Maarten, AN, IARIA, 2011, pp.12-18.

[18] M. Sveda and R. Vrba, An Embedded Application Regarded as Cyber-Physical System, *Proceedings of the Fifth International Conference on Systems ICONS 2010*, Les Menuires, FR, IARIA, 2010, pp.170-174.

[19] M. Sveda and R. Vrba, Meta-Design with Safe and Secure Embedded System Networking, *International Journal On Advances in Security*, Vol. 2, No. 1, 2009, US, pp.8-15.

[20] M. Sveda and R. Vrba, Specifications of Secure and Safe Embedded System Networks, *8th International Conference on Networks Proceedings ICN 2009*, New York, NY, US, IARIA, IEEE CS, 2009, pp.220-225.

[21] M. Sveda, A Design Framework for Internet-Based Embedded Distributed Systems, *Proceedings of the International IEEE Conference and Workshop ECBS'2004*, Brno, Czech Republic, IEEE Computer Society Press 2004, pp.113-120.

[22] H. Tang and B.M. McMillin, Security Property Violation in CPS through Timing, *Proceedings of the 28th on Distributed Computing Systems IDCS 2008, Workshops*, IEEE CS, New York, NY, US, 2008, pp.519-524.

[23] R. Vrba, O. Sajdl, R. Kuchta and M. Sveda., Wireless Smart Sensor Network System. *Proceedings of the ICSE & INCOSE. Conference*, Las Vegas, Nevada: CRC Press LLC, 2004. pp.104-109.

[24] P. Zave and M. Jackson, Four Dark Corners of Require-ments Engineering, *ACM Transactions on Software Engineering and Methodology*, Vol.6, No.1, 1997, pp.1-30.

 ISBN: 978-1-61208-184-7

ISA-95 Tool for Enterprise Modeling

Dazhuang He, Andrei Lobov, Jose L. Martinez Lastra
FAST Lab, Department of Production Engineering
Tampere University of Technology
P.O. Box 600, 33101, Tampere, Finland
{Dazhuang.He, Andrei.Lobov, Jose.Lastra}@tut.fi

Abstract— **Enterprise information modeling is one of the major challenges for system integration in Factory Automation. Different standards exist to model information. This paper describes ISA-95 Tool that is developed based on internationally-used industrial standard ANSI/ISA-95. The tool makes it easier to automatically integrate product information with a production system. Up to the knowledge of authors, so far only ad hoc solutions were developed following ISA-95, which were failing to support in general the modeling of manufacturing systems. The tool can be used for production order specifications. An overview on ISA-95, B2MML (Business to Manufacturing Mark-up Language) and structures as SIIS (Software Intensive Industrial System) are described in the paper, which are followed by tool description and case study.**

Keywords- B2MML; ISA-95; Enterprise modeling.

I. INTRODUCTION

The information flow in industrial systems grows in terms of amount and structural complexity. At the moment factory information system implementation follows ad hoc solutions that may be based on some of the standards, i.e. ISA-95 or CAMX [1], but lack to have an adequate tools support for information *modeling*. The developing tendency of industrial systems shifts towards SIIS where software essentially influences to the design, construction and deployment of these systems. GAO, known as U.S. Government Accountability Office, attributes the poor success degree [2] of building software intensive systems to the management [3], in detail, ERP (Enterprise Resource Planning) and MES (Manufacturing Execution System) levels. From modules to methodologies, from languages to services, during last two decades an extensive research is performed to interconnect different enterprise systems and refine the SDLC (System Development Life Cycle).

For example, SOA (Service-Oriented Architecture) is a paradigm developed for organizing and utilizing distributed capabilities under different ownership domains [4]; OPC was designed to provide a common bridge for Windows based software solutions and process control hardware; the mechanism of loose coupling keeps different part of one system maintained own functionalities with communicating through well-defined interfaces [5].

ISA-95 developed by the Instrumentation Systems and Automation Society (ISA) defines a complete functional model for enterprise-control use as a reflection of an organizational structure of functions which can be replaced addressing different demands of the enterprise.

Following ISA-95 standard, this paper presents a tool that allows modeling of enterprise information. The tool can be used to allow adaptation to any specific demands of the enterprise. The models can be extended. The tool is demonstrated on manufacturing line producing mobile phones, where it is used to represent order information.

The paper is organized as follows: next section gives short introduction to ISA-95 standard and B2MML that provides XML representation for ISA-95, which is important for interoperability of IT systems at factories. Third section describes the tool. The use case is presented in fourth section that is followed by conclusions.

II. THE ISA-95 STANDARD & B2MML

A. The ISA-95 Standard

ISA-95 is originally a US standard which has been adopted as an international one under IEC/ISO 62246. As currently envisioned, the ANSI/ISA-95 series will consist of the 5 parts under the general title, Enterprise-Control System Integration:

- Part1:Models and terminology
- Part2:Object model attributes
- Part3:Activity models of manufacturing operation management
- Part4:Object models and attributes of manufacturing operations management
- Part5:Business to Manufacturing transactions

The latest versions of Part 1, 2 and 3 are released on 2010 while part 4 and part 5 are still under standardization. In this article, second latest version of ISA-95 and B2MML (v4010) were selected as a stable combination for their compatibility.

As a structured standard, ISA-95 includes 3 main information areas of producing products, capabilities, and actual production. Besides, the standard provides a reference model for system organizing, allocating business to the different systems and information flow between systems [6].

UML (Unified Modeling Language) models are used in the development of the tools following ISA-95. The 9 object

 ISBN: 978-1-61208-184-7

models, 86 objects and a whole set of attributes defined in ISA-95.00.02 are extensions to the information models defined in ISA-95.00.01.

The structure and the frame allow users of addressing own information inheriting the relationship between information blocks (known as classes in Object Models).

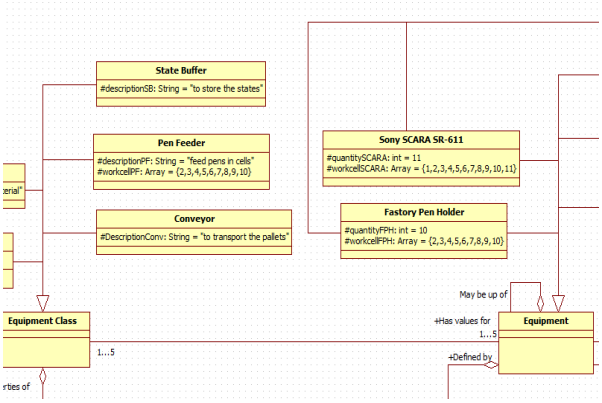


Figure 1. Part of Equipment Object Model for FASTory Line.

In an example of Equipment Object Model for the case study (Section IV), FASTory Line, the information of problem domain concepts such as “State Buffer”, “Pen Feeder”, “Conveyor” is well classified and the generalization between “Equipment Class” and them is also kept. In order words, standard UML models can be extended to fit the problem domain (Fig. 1).

The tool to be introduced in next part - “ISA-95 Tool” serves as an interface with all the classes beyond and under problem domain defined in ISA-95.00.02. In majority of the industrial systems, the problem domain consists of 4 models:

1) The personnel object model describes human resources defining different classes of personnel.

2) The equipment object model is structured similarly--the object model supports specifying requirements for different equipment classes.

3) The material model describes raw materials, intermediate products, and finished products.

4) A process segment is one step/task/unit of work that must be performed to complete a product.

The five other object models defined in ISA-95.00.02 beyond problem domain are production capability model, process segment capability model, product definition information model, production schedule model, production performance model.

B. B2MML

With a set of XML schemas written using the World Wide Web Consortium's XML Schema language (XSD), B2MML is treated as a complete XML implementation of ISA-95. The .xsd templates implement the data models in the ISA-95 standard. The final information carrier of ISA95-Tool will be an .xml file following B2MML’s .xsd template [7].

From the perspective of an SDLC, the link of support phase [8] in ISA-95 is still weak. This is reflected in the lack of tools and platform based on the standard. This stage of conceptualization greatly demands specific visualization to increase engineers’ efficiency on familiarizing and using this standard. Part III of this article depicts “ISA-95 Tool” as one solution addressing this demand.

III. INTRODUCTION TO ISA-95 TOOL

As mentioned in the previous part, there is a lack of visual-operating software as support phase for practical application of ISA-95`. The acknowledge degree still stays in the combination of models and attributes, which increases extremely the difficulty of application of the standard.

As one solution to the problem, “ISA-95 Tool” defines ”order” as the core concept and information carrier functioning in Manufacturing, Operation and Control level (level 3 in ISA standard family) and Business Planning and Logistics level (level 4 in ISA standard family) [9]. The process starts when the order is received from a customer and then transferred between system managers, analysts and operators. The process ends up with creating and transferring an .xml file to line controller that based on product needs should generate a production recipe. The recipe can be formalized in Business Process Execution Language (BPEL) [10].

However, system demands may vary from factory to factory and so not all models are necessary to keep the process running in a practical industrial use.

The first phase presented as a frame allows users to select models by their demands (Fig. 2).

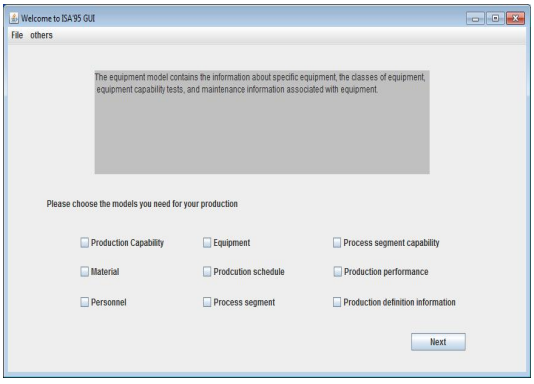


Figure 2. Model-selection phase of ISA-95 Tool.

The tool will list attributes and text fields under selected models (Fig. 3). The definitions of the attributes in the 9 models come from a minimum set of industry-independent information. The attributes are extensions to the object information model defined in ISA-95.00.01 and thus are

 ISBN: 978-1-61208-184-7

part of the definition of terms. The attributes and models define interfaces for enterprise-control system integration.

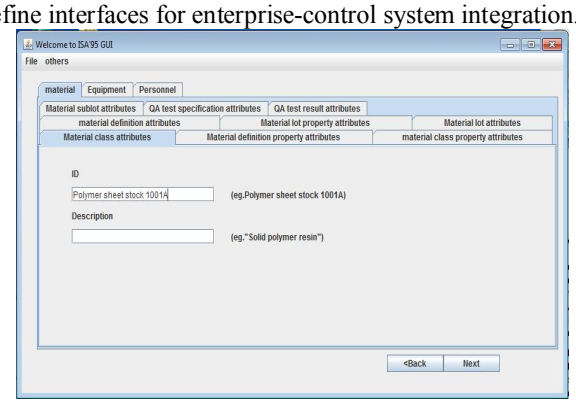


Figure 3. Attributes information phase of ISA-95 Tool.

An order list with information as Order ID and production start time can be created in third phase. A single cycle for all the operations in an order can be completed by pressing "Start" button (Fig. 4).

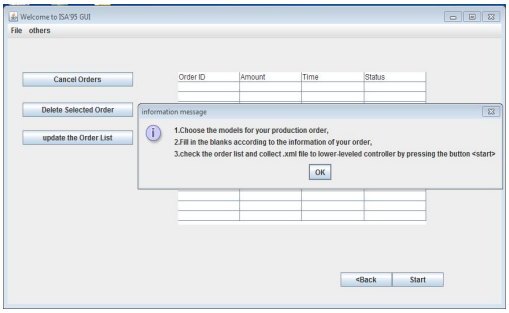


Figure 4. Order-checking phase of ISA-95 Tool.

After the operations cycle, an .xml file collecting inputs will be created by the tool and transferred to controllers at the production line level.

This is done by adding values to an existing .xml file template. As mentioned, from v02 of B2MML, .xsd files are available as part of the packages released by WBF (The Organization for Production Technology). The template here is created following .xsd file format. Little changes as adding root elements to .xsd files are required if the format transformation is completed by an xml software.

IV. CASE STUDY: FASTORY LINE

A. *FASTory Line*

FASTory-Line is an assembly line used for research purposes in FAST Lab, Tampere University of Technology (Fig. 5). In order to simulate the production (due to costs reasons), drawings of components and products are created by robots. The main advantage is that, different drawings of components are used to simulate parts of the assembly and different colors for increasing the complexity of the systems as well as production customization. As a result, there is no need disassemble ready products, however the actual robots need to perform pick and place operation and follow unique sequence depending of the variant of the product.

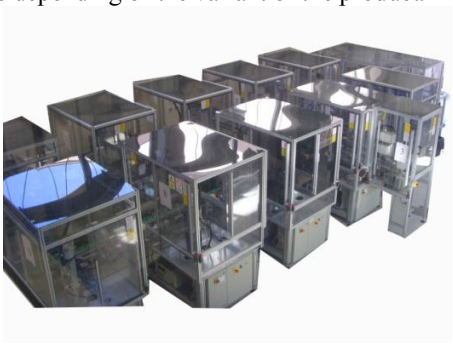


Figure 5. FASTory Line

The drawing consists of 3 components: frame, screen and keyboard, 3 different formats for each component and 3 different pen colors (red, green and blue) for each format thus the product has 729 variants altogether. All the drawing robots can take the task of drawing any part. It is also possible to make the complete mobile phone or finish only one part to bring larger flexibility of the line. The material to be used in the production will be paper and pens.

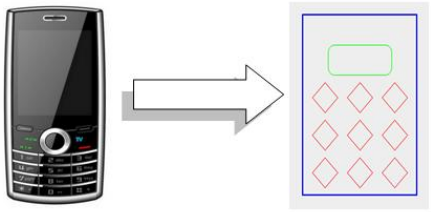


Figure 6. Cell-phone simulation.

B. *Modified version of "ISA-95 Tool"—"FASTory GUI"*

FASTory line is a typical production line with strong demands on models in problem domain, mainly for equipment model, personnel model and material model. The traditional solution with "ISA-95 Tool" is to choose problem domain models in first phase, to fill attributes forms in second phase before final order is created. However, a different scenario is presented below as an alternative solution.

"FASTory GUI" is developed as a modified version of "ISA-95 Tool" being optimized for FASTory Line. Starting from the action flow, the user set input by choosing radio button groups and making selections in combo boxes in the first phase (Fig. 7).

Instead of text-fields, the material information is represented directly in the form of component formats. The volume of the ink in different colors changes as the user chooses different cell phone formats and colors. The product segment information in accordance with the choice will be displayed in a table simultaneously.

 ISBN: 978-1-61208-184-7

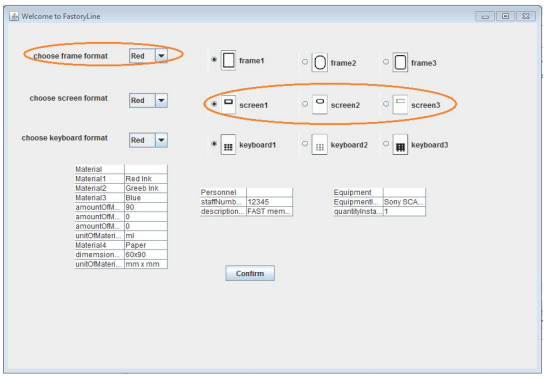


Figure 7. Cell-phone format selection phase of FASTory GUI.

After confirmation, the user can check the information as production segment rules, production schedules and even a preview of the cell-phone product (Fig. 8). The user adds orders list after the correction of the mistakes (if any).

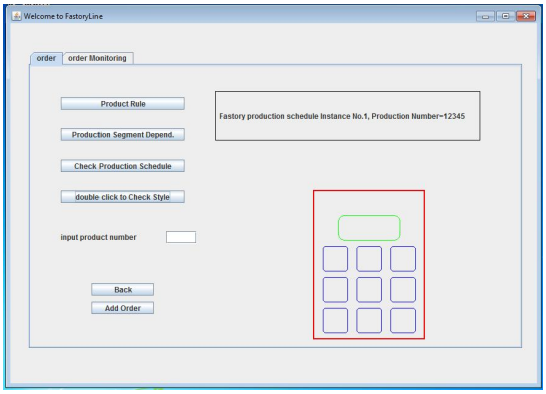


Figure 8. Production Segment Information phase of FASTory GUI.

The same as with "ISA-95 Tool", if the order does not reach the requirement, the user can delete unwanted order in monitoring tab or exit the procedure by cancelling the orders.

Also, an .xml file collecting user's choices and carrying order information will be created and transferred to line level controllers. Again, here an alternative method is presented instead of feeding values to .xml templates.

In an SIIS as FASTory Line, not all data are in top level of importance, which means only part of information can keep the system processes running. Another reason on reducing the amount of the information is that irrelevant elements, null elements and long headers increase workload and difficulty for line controllers on information analysis. Here a minimum set of elements are chosen from the template artificially and an example of this has been tested on FASTory GUI. An example of a part of the .xml code looks as follows:

```
<amount>99</amount>
<time>Tue Sep 20 12:58:58 EEST 2011</time>
<Formats>
<frameFormatColor="1">1</frameFormat>
<screenFormatColor="0">0</screenFormat>
<keyboardFormatColor="1">1</keyboardFormat>
</Formats>
```

As depicted in this scenario, the formats and colors are represented by integers in elements and attributes; the programming solution reduces the time needed analyzing and extracting information from files.

V. CONCLUSION

The ISA-95 standard is an important basis for the development connecting control system and enterprises. B2MML is selected as implementation language for the standard to allow interoperability of industrial IT systems. "ISA-95 Tool" allows visualization of the models and attributes staring from abstract concepts, refining and placing them into practical industrial use. FASTory tool is a specialized version based on "ISA-95 Tool" taking FASTory Line as a study case. It is also a good example of how "ISA-95 Tool" can be extended to fit factories, enterprises in different size and types as separate solutions, though it is already sufficient and powerful enough working as an independent tool.

The further development and research can focus on the modeling of production performance that can be checked after at least one single process segment and return information back to ISA-95 models. Thus an .xml file containing performance model information is needed and can be generated as a result of feedback information coming from line and low-level controllers. This addition will require some changes on current web service interface between the tool and the controllers of the line.

Another issue is that majority of targeted users of current software products on ISA-95 application are "solution architects", "analysts", and "engineers" but not the managers who do not have flexibility to update the model or change the schedule. This indirectly increases the operating requirement even if there are no problems other than the format and the access of the information. Thus an extra step could be added transferring created .xml file to .xls file. The format of an Excel table which avoids the specific knowledge requirements of B2MML is considered for the future work.

REFERENCES

[1] CAMX Reference, http://gocamx.com/camx, Great Technologies Collaborations, Inc. (retrieved: January, 2012)

[2] R. N. Charente, "Why Software Falls," IEEE Spectrum,vol. 42, no. 9, pp. 36-43

[3] P. I. Sosnin, "Conceptual solution of the tasks in designing the software intensive systems," 14th IEEE Mediterranean Electrotechnical Conference (MELECON 08), pp.293-298, 5-7 May 2008, doi: 10.1109/MELCON.2008.4618450

[4] "Service Oriented Architecture (SOA)", OASIS, http://xml.coverpages.org/soa.html (retrieved: January, 2012)

 ISBN: 978-1-61208-184-7

[5] L. Kerschberg, "The role of loose coupling in expert database system architectures," 5th International Conference on Data Engineering, pp. 255-256, 6-10 Feb 1989, doi: 10.1109/ICDE.1989.47222

[6] I. M. Delamer and J. L.Martinez Lastra, "Factory Information Systems in Electronics Production", Tampere University of Technology, ISBN 978-952-15-1937-6, 2006

[7] ANSI/ISA-95.00.02-2001, "Enterprise-Control System Integration Part2: Object Model Attributes", ISA, 2001

[8] J.W. Satzinger, R.B. Jackson, and S.D. Burd, "Object-oriented Analysis & Design with the Unified Process," Thomson, ISBN 0-619-21643-3, 2005

[9] ANSI/ISA-95.00.03-2005, Enterprise-Control System Integration Part 3: Activity Models of Manufacturing Operations management, ISA 2005

[10] J. Puttonen, A. Lobov, and J.L. Martinez Lastra "An application of BPEL for service orchestration in an industrial environment," IEEE International Conference on Emerging Technologies and Factory Automation, (ETFA 08)., pp. 530-537, 15-18 Sept. 2008, doi: 10.1109/ETFA.2008.4638450

 ISBN: 978-1-61208-184-7

Evaluating Service-Oriented Orchestration Schemes for Controlling Pallet Flow

Johannes Minor, Jorge Garcia, Jaacan Martinez, Andrei Lobov, Jose L. Martinez Lastra
Tampere University of Technology
Tampere, Finland
{johannes.minor, jorge.garcia, jaacan.martinez}@tut.fi, {lobov, lastra}@ieee.org

***Abstract*—Incorporating web services at the device level is expected to improve many aspects of automated manufacturing systems, including scalability, reusability, reconfigurability, compatibility between equipment from different vendors, and cross-layer integration. There is much ongoing research into architectural issues and enabling technologies for service-oriented automation, but the body of knowledge surrounding service identification and optimal orchestration schemes needs improvement. In this project, the performance of a test system, consisting of conveyors with embedded web service-enabled devices, with three different orchestration schemes is evaluated. The three schemes have similar message exchange patterns, but differ in terms of the degree of involvement of a central orchestrator. The schemes are compared in terms of pallet transfer timing, and communications load. Performance is similar for the small test system, but the results predicted scalability problems for the schemes with high reliance on a central orchestrator.**

Keywords-SOA; orchestration; control schemes.

I. INTRODUCTION

The case for applying Service Oriented Architecture (SOA) to industrial control systems has been made in many previous research projects [1,5,10]. The benefits that Service Orientation at the device level is expected to bring to industrial applications include increased business agility, easier and less costly equipment reusability and reconfigurability, and improved cross-layer integration. In addition, building systems around open standards reduces a business' reliance on proprietary protocols, systems, and data formats, and can drastically decrease the effort with which systems from multiple vendors can be integrated, allowing factories to choose the best-of-breed components for all systems, without worrying about interoperability.

Much of the research into service orientation in industrial automation has focused on architectural issues, and the enabling technologies and standards. Still missing, however, are the practical implementation details, best practices, and system design methodologies that can help bridge the gap between systems theory and a working SOA deployment that exhibits some or all of the benefits touted by SOA advocates.

Although the benefits of service orientation have been established, additional research is required to expand the body of knowledge surrounding the field. Specific issues that still need to be addressed are:

- How to combine scan-based and event-based systems?
- Down to what level is it reasonable to have web services?
- How best to compose services over a number of distributed devices to execute business processes?
- How to integrate legacy equipment into a SOA-based system?

This paper proposes and evaluates some possible orchestration schemes for handling pallets on a system of conveyors, in an attempt to provide some answers to the following questions:

- What is the best approach for composing atomic services into a more complex task?
- How can we combine event- and scan-based SOA manufacturing systems?

A. Orchestration and Choreography

When discussing SOA in automation, orchestration typically refers to the practice of composing a set of exposed services, with a pre-defined interaction pattern that defines a business or manufacturing process [1]. The process can be described using a language such as Web Services Business Process Execution Language (WS-BPEL). The orchestration engine executes the application logic, sequencing and synchronizing service invocations to reach the business goal [15].

The focus of orchestration is on a high-level view of the process workflow, whereas choreography considers the lower level rules that define the message exchange sequences. The W3C candidate recommendation Web Service Choreography Description Language v1.0 (WS-CDL) [8] can be used to describe peer to peer collaboration by defining their globally observable behavior, where business goals are realized by peer to peer message exchange.

B. Previous Research

EU project SODA [16] investigated the eco-system required to build, deploy, and maintain a SOA application in many domains. The SIRENA [17] project demonstrated the feasibility of extending SOA to the device level, and produced some proof-of-concept device-level services [10]. FP6 SOCRADES [18] evaluated a number of solutions for SOA at the device level, and demonstrated a SOA solution for automating electronics assembly.

Much research has been done on system modeling, control, and decision making at higher levels, with Petri Net-based approaches [2,4] and Timed Net Condition/Event

 ISBN: 978-1-61208-184-7

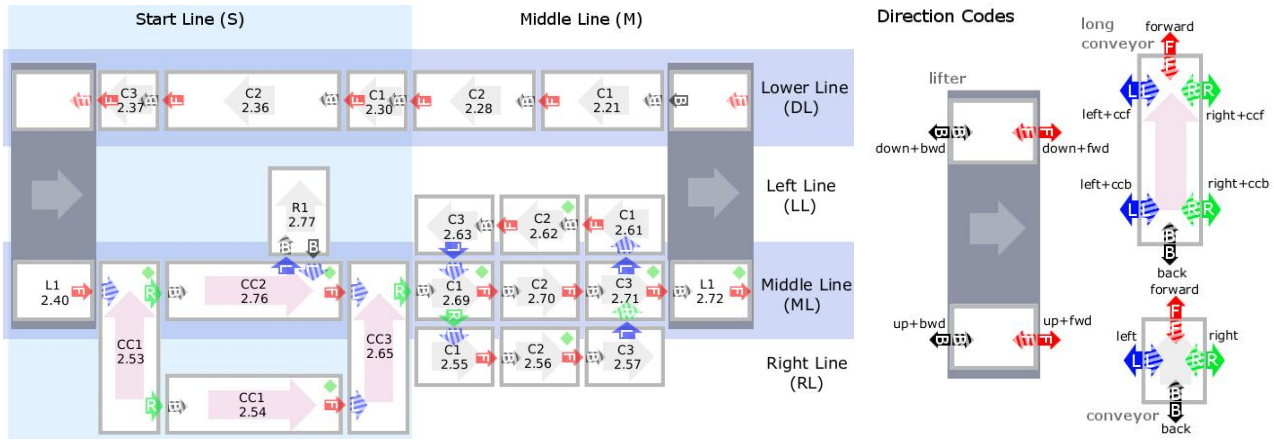


Fig. 1. System of conveyor segments used for testing pallet flow control schemes.

Systems (TNCES) [3]. Other research presents designs for orchestration engines for controlling systems composed of web service-enabled embedded devices [3,11].

Some software engineering approaches to combining, modeling, and optimizing service orchestration and choreography have been proposed [6,7], but the focus has been on eliminating redundant data transfers to minimize process execution time in IT systems. Time-consuming data transfers between devices are not the primary concern on the factory floor.

This paper describes some preliminary research into optimal strategies for composing device-level, fine-granularity, atomic services, to synchronize device interactions to efficiently complete manufacturing processes. This research aims to find a balance between device-to-device interactions, and master-slave control schemes, with long-term goal of providing an easier transition path

Sections II and III introduce the test system, and describe the control schemes. Section IV analyzes the data, and Section V presents conclusions and future work.

II. SYSTEM DESCRIPTION

For this study, three different pallet transfer control schemes are tested using a varying number of pallets on the system shown in Fig. 1.

The system consists of 21 conveyor segments. An intelligent Web Services-enabled device controls each segment. This line uses the InicoTech S1000 Smart RTU [13]. The main loop is made up of the middle and lower lines, connected at either end by a lifter, 13 segments in total. The continuously moving loop acts as a buffer for the robot and manual workstations off the main line. Pallets can be introduced to the system at loading stations in the branches off the main loop.

Each pallet has a unique RFID code, read when a pallet is loaded at a loading station. To transfer a pallet from one conveyor segment to the next, the following message exchange takes place, independent of the control scheme. This pattern is shown in Fig. 2.

1) Reservation Request/Response: A message containing the pallet ID and input transfer direction is sent to the conveyor or lifter. The reservation status, and the ID of the pallet holding the reservation are returned. If the pallet IDs match, and the status is "RESERVED," the transfer can safely proceed. If the device is not in a position to accept a pallet at the requested input position (e.g., reservation is requested at the upper end of a lifter while the lifter is down), the reservation status is "PENDING." Reservation Requests are sent until the reservation is successful.

2) Transfer In Request/Response: A message containing the pallet ID and the input transfer direction are sent. If they match the reservation information, "ACCEPTED" is returned.

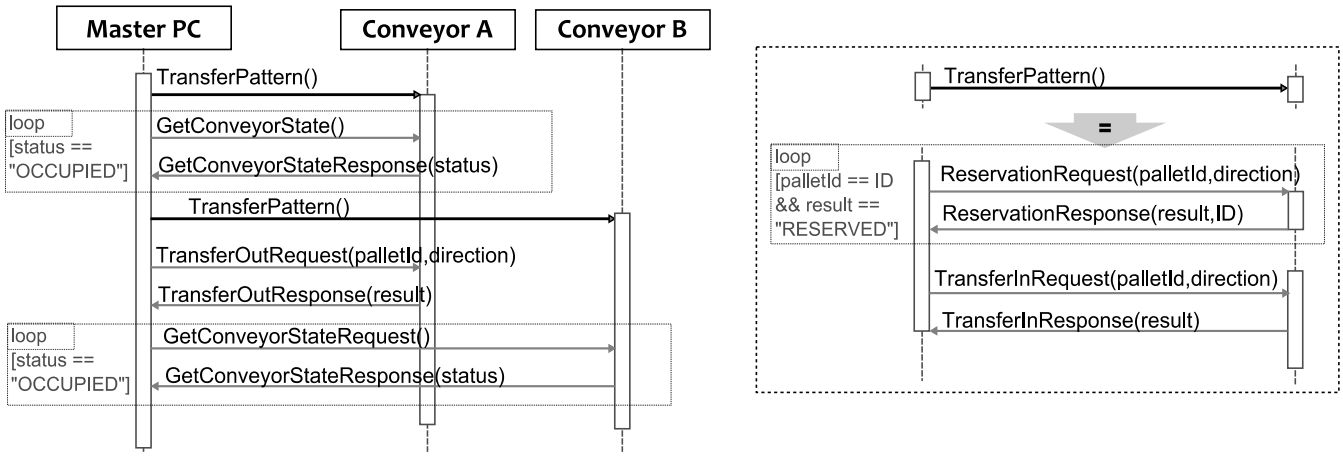


Fig. 2. Master-Slave Control Scheme

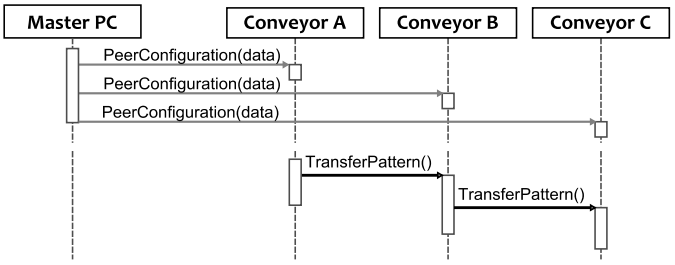


Fig. 3. Peer to Peer Control Scheme, using TransferPattern from Fig. 2.

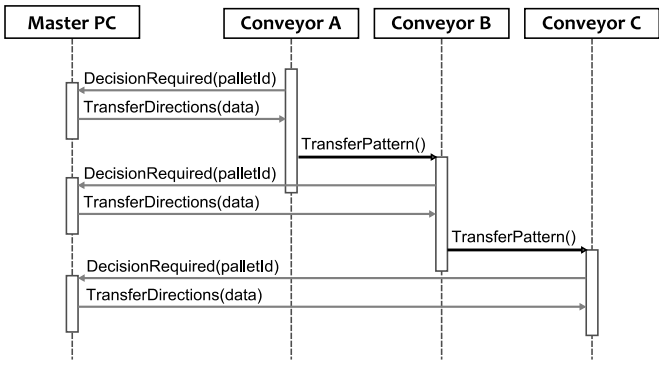


Fig. 4. Combined Control Scheme, using TransferPattern from Fig. 2.

III. CONTROL SCHEME DESCRIPTIONS

For this study, varying numbers of pallets are added to the main loop, driven by different control schemes. The three schemes chosen for comparison represent different balances between device-to-device interaction, and top-down orchestration.

A. Master-Slave: All interactions through Orchestration Engine

Each remote device is a DPWS server, and a single master DPWS client controls the pallet flow by invoking actions on all remote devices. The remote devices themselves do not autonomously invoke actions on other remote devices. Each remote device supports the same set of operations:

1) Reserve: A reservation request message is sent, containing a unique RFID (Radio Frequency IDentification) Code to identify the pallet, and the input transfer direction. Direction codes supported by each device can be determined from metadata in the WSDL read from the device in the discovery phase. This operation returns the reservation state (free, pending, reserved), and the ID of the pallet that the conveyor is reserved for, if any.

2) TransferIn: The transfer-in request message contains the same data as the reservation request. The operation is accepted if the request message matches the reservation data, or rejected if it does not.

3) GetConveyorState: After the *TransferIn* action is invoked, the conveyor state is polled at some interval. The operation is complete when the conveyor state request returns "occupied."

4) TransferOut: When a downstream conveyor is reserved, the transfer-out operation is invoked. The request message contains the pallet Id and the output direction code.

5) ReadRfidTag: Certain devices have an RFID tag reader, which is used to read the unique RFID code of the pallet when it is first loaded, and for consistency-checking during operation.

One process is required on the master PC running the Orchestration Engine for each pallet. The message exchange pattern is shown in Fig. 2. This exchange pattern can be described in WS-BPEL.

B. Peer to Peer: Devices handle pallet transfer autonomously

The devices cooperate autonomously with each other to achieve common goals. Each remote device acts as a combined DPWS client and server, each with the ability to invoke actions on other remote devices. Each device supports the same set of services, and follows the same interaction pattern to allow for collision-free pallet flow. At startup, a supervisory control system dynamically discovers the devices present in the network, and based on some layout map, and invokes a configuration service on each device, containing the following information:

- Output transfer direction code
- Service Address of the downstream peer device
- Input transfer direction to the downstream peer device

When pallet is introduced to the system at a loading station, neighboring devices negotiate pallet transfer with the message exchange pattern shown in Fig. 3, containing the same *Reserve* and *TransferIn* operations as the Orchestrator control scheme. No *TransferOut*, *GetConveyorState*, or *ReadRfidTag* operations are required.

C. Hybrid Approach: Peer to Peer with Decision Request Notifications

In this approach, lower-level pallet transfer control

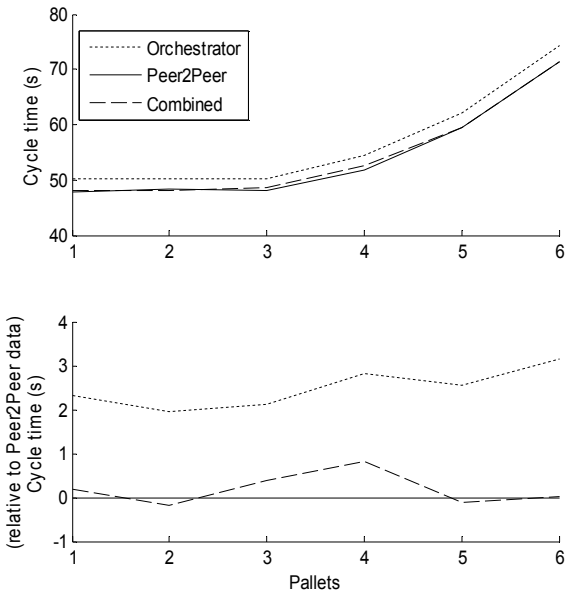


Fig 5. Main Loop Cycle (Lap) Times for different Control Schemes.

 ISBN: 978-1-61208-184-7

messages (*ReservationRequest*, *TransferIn*) are exchanged peer to peer, as in the previous scheme, but the device receives transfer instructions (downstream peer service address, output transfer direction, peer input transfer direction) from the master after the device publishes a notification that a pallet has been received (*DecisionRequest*). This exchange pattern is shown in Fig. 4.

When the system is stared, the master PC dynamically discovers all devices, and subscribes to the *DecisionRequest* notifications. When a notification is received, some logic executed on the master PC determines the appropriate next step, and sends the appropriate transfer instructions.

IV. EXPERIMENT AND RESULTS

The tests were conducted as follows:

1) A configuration file describing the layout of the conveyor system is loaded in the control software running on the master PC. The master PC subscribes to the "*PalletInformation*" notifications on each device, used for logging purposes, common to all control schemes.

2) A single pallet is transferred from conveyor to conveyor around the loop.

3) After fifteen minutes, a new pallet is placed at a loading station, and introduced into the main loop with the existing pallets

4) Additional pallets are introduced at approximately equal intervals for 90 minutes

A. Average Pallet Lap Time

The average time for a pallet to complete one lap of the main loop for the three schemes is shown in Fig 5. The results are plotted relative to the data from the Peer to Peer test, because it had the fastest lap times, on average. These results are not surprising. Although all control schemes use the same interaction pattern for reservation and pallet transfer (i.e. polling at a 500ms interval for reservation), the Master-Slave scheme has an additional, approximately 350ms polling cycle to determine when the TransferIn operation is complete. With thirteen devices in the loop, we would expect the lap time to be approximately 13×(350/2)ms = 2.275s longer for a single pallet.

The performance bottlenecks in the system were the lifters, because of time taken to transfer a pallet between the upper and lower lines. While fewer than four pallets are in the loop, transfer times stay constant, because the pallets do not interfere with each other. For four pallets, the marginally faster performance of each transition using the Peer to Peer approach results in noticeably faster average times for four pallets, but the advantage fades for five or more pallets. This behavior is coincidental, not inherent to the control scheme, and would likely change if the conveyor line layout were changed.

We can conclude that the orchestration scheme chosen has a negligible impact on the timing of pallet transfers.

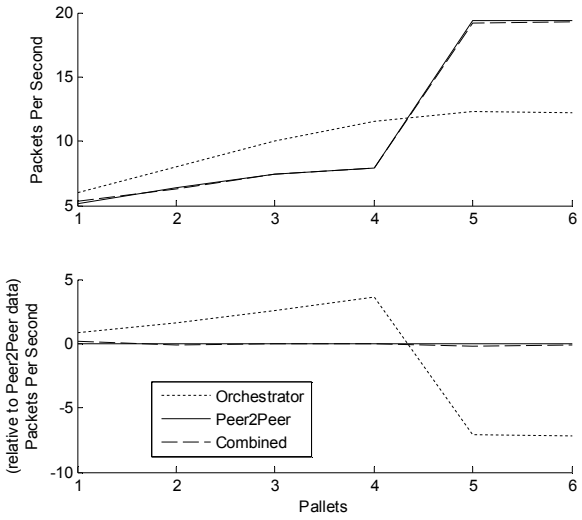


Fig 6. Packets sent per second by a device before the lifter bottleneck

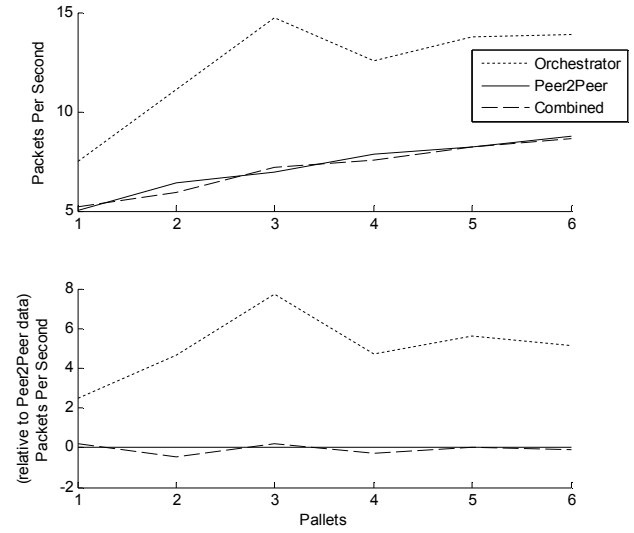


Fig 7. Packets sent per second by a device after the lifter bottleneck

B. Communication Load

Access to network statistics for the devices was limited to the number of packets sent and received since startup. Although this is not useful as an absolute measure, it can be used to compare relative performance. Fig. 8 shows a typical data set for one test for one device. There is a spike in activity when a new pallet is introduced, and then it stabilizes.

Fig. 6 and Fig.7 show the average load for devices before and after the lifter bottleneck. For lower pallet numbers, the communication load is lower devices operating under the peer to peer and hybrid schemes. This is likely because there is no polling taking place. Reservation requests are event-based (i.e. sent when a pallet is received). When the pallets start to interfere with each other, the peer to peer and combined approaches create communication loads almost twice that of the orchestrator approach. However, it is important to note

 ISBN: 978-1-61208-184-7

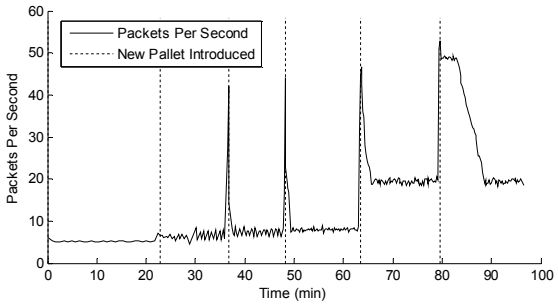


Fig 8. Typical "Packets Per Second" data

that the orchestration engine running on the PC is experiencing a load roughly equivalent to the difference, because the devices are not communicating directly with each other. This raises scalability questions, because the communications load from all devices is concentrated at the orchestrator.

For devices after the bottleneck, as in Fig. 7, communication load remains lower for the event-driven peer-to-peer and combined control strategies.

V. CONCLUSIONS AND FUTURE WORK

This research shows that for small systems, timing of physical operations, such as pallet transfers, are not very sensitive to the choice of orchestration scheme. However, analysis of the communication loads on the remote devices and the orchestrator (or master PC) suggests that performance will degrade with increasing system size for schemes that rely heavily on a central orchestrator. Additional research is required into the scalability of these methods.

This research lays the groundwork for designing and testing more complex systems. A system with autonomous devices interacting requires robust supervisory control and monitoring. Implementing Complex Event Processing (CEP) for monitoring and decision support for orchestration can be also considered. Complex Event Processing, used in conjunction with various formal system modeling methods [2,12], is a promising approach for providing detailed information about the state of the system, as well as fault prediction, prevention, and detection, and deadlock prevention in flow control.

REFERENCES

[1] F. Jammes, H. Smit, J.L. Martinez Lastra, and I.M. Delamer, "Orchestration of service-oriented manufacturing processes," IEEE 10th Conference on Emerging Technologies and Factory Automation (ETFA 05). vol. 1, pp. 617-624, 19-22 Sept. 2005

[2] C. Popescu and J.L. Martinez Lastra, "An incremental Petri Net-derived approach to modeling of flow and resources in service-oriented manufacturing systems," IEEE 8th International Conference on Industrial Informatics (INDIN 10), pp. 253-259, 13-16 July 2010

[3] A. Lobov, J. Puttonen, V.V. Herrera, R. Andiappan, and J.L. Martinez Lastra, "Service oriented architecture in developing of loosely-coupled manufacturing systems," 6th IEEE International Conference on Industrial Informatics (INDIN 08), pp. 791-796, 13-16 July 2008

[4] J.M. Mendes, P. Leitao, A.W. Colombo, and F. Restivo, "Service-oriented process control using High-Level Petri Nets," 6th IEEE International Conference on Industrial Informatics (INDIN 08), pp. 750-755, 13-16 July 2008, doi: 10.1109/INDIN.2008.4618202

[5] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," IEEE Transactions on Industrial Informatics, vol. 1, no. 1, pp. 62-70, Feb. 2005

[6] B. Haopin, S. Meina, X. Huiyang, W. Qian, and F. Lingyun, "An Optimized Design of Service Orchestration," Third International Conference on Pervasive Computing and Applications (ICPCA 08), vol. 2, pp. 980-984, 6-8 Oct. 2008

[7] J. Sun, Y. Liu, J. Song Dong, G. Pu, and T. Hut Tan, "Model-Based Methods for Linking Web Service Choreography and Orchestration," 17th Asia Pacific Software Engineering Conference (APSEC 10), pp. 166-175, Nov. 30-Dec. 3 2010, doi: 10.1109/APSEC.2010.28

[8] Web Services Choreography Description Language Version 1.0 http://www.w3.org/TR/ws-cdl-10/ (retrieved: January, 2012)

[9] OASIS Web Services Business Process Execution Language V2.0, 11 APR 2007, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html (retrieved: January, 2012)

[10] A.W. Colombo, F. Jammes, H. Smit, R. Harrison, J.L. Martinez Lastra, and I.M. Delamer, "Service-oriented architectures for collaborative automation," 31st Annual Conference of IEEE Industrial Electronics Society (IECON 05), pp. 2649-2654, 6-10 Nov. 2005, doi: 10.1109/IECON.2005.1569325

[11] Y.S. Park, T.D. Kirkham, P. Phaithoonbuathong, and R. Harrison, "Implementing agile and collaborative automation using Web Service orchestration," IEEE International Symposium on Industrial Electronics (ISIE 09), pp. 86-91, 5-8 July 2009, doi: 10.1109/ISIE.2009.5213759

[12] D. Cachapa, R. Harrison, and A.W. Colombo, "Monitoring functions as service composition in a SoA-based industrial environment," 36th Annual Conference on IEEE Industrial Electronics Society (IECON 10), pp. 1353-1358, 7-10 Nov. 2010, doi: 10.1109/IECON.2010.5675485

[13] InicoTech Technologies LTD; S1000 User Manual; http://www.inicotech.com/doc/S1000%20User%20Manual.pdf (retrieved: January, 2012)

[14] J. M. Garcia Izaguirre, A. Lobov, and J.L. Martinez Lastra, "OPC-UA and DPWS Interoperability for Factory Floor Monitoring using Complex Event Processing," 9 th IEEE International Conference on Industrial Informatics (INDIN 11), pp. 205-211, 26-29 July 2011 doi: 10.1109/INDIN.2011.6034874

[15] J. Puttonen, A. Lobov, and J.L. Martinez Lastra, "An application of BPEL for service orchestration in an industrial environment," IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 08), pp. 530-537, 15-18 Sept. 2008

[16] SODA Project Profile, http://www.ims.es/pdf/eng/downloads/publications/SODA_profile_oct-06.pdf (retrieved: January, 2012)

[17] SIRENA Project, http://www.sirena-itea.org/ (retrieved: January, 2012)

[18] SOCRADES Project, http://www.socrades.eu/Home/ (retrieved: January, 2012)

 ISBN: 978-1-61208-184-7

Exploring Entropy in Software Systems: Towards a Precise Definition and Design Rules

Herwig Mannaert, Peter De Bruyn, Jan Verelst
Department of Management Information Systems
University of Antwerp
Antwerp, Belgium
{*herwig.mannaert, peter.debruyn, jan.verelst*}*@ua.ac.be*

***Abstract*—Software systems need to be agile in order to continuously adapt to changing business requirements. Nevertheless, many organizations report difficulties while trying to adapt their software applications. Normalized Systems (NS) theory has previously been able to introduce a proven degree of evolvable modularity into software systems, based on the systems theoretic notion of stability. In this paper, we explore the applicability of this other fundamental property of systems (i.e., entropy) to the issues of software maintenance and evolvability. The underlying concepts in entropy definitions will be explained and applied to software systems and architectures. Further, the considerable complexity of running multi-tier multi-threading software systems and the relation with entropy concepts is discussed and illustrated. Finally, the concordance of design rules for controlling that entropy with previously formulated NS principles is explored.**

***Keywords*-Normalized Systems, Entropy, Systems engineering, Evolvability**

I. INTRODUCTION

Current organizations need to be able to cope with increasing change and increasing complexity in most of their aspects and dimensions. As a consequence, all constructs and artifacts of an organization have to be able to swiftly adapt to this agile and complex environment, including its business processes and organizational structure, as well as its supporting information systems. Indeed, also the software applications an organization employs, should be able to evolve at an equivalent pace as the business requirements of the organization they are embedded in.

However, many indications are present that most modular structures in software applications do not exhibit this required evolvability, flexibility, etcetera. One very early indication of this phenomenon was expressed by the formulation of Manny Lehman's Law of increasing complexity, stating that "*As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it.*" [1, p. 1068]. Interpreting entropy as a measure for uncertainty or the degree of absence of structure (i.e., disorder) in a system, one could conceive Lehman's law as referring to the irreversable tendency of software applications to build up entropy (i.e., structure deterioration and degradation) troughout their lifecyle and hence become more and more complex while being less and less maintainable as time goes by. Indeed, Lehman himself initially proposed his law as an instance of the second law of thermodynamics [1]. Therefore, the law can also be interpreted as describing ever increasing entropy or disorder in the structure of software systems, unless effort is done in order to reduce the amount of entropy. In a similar way, Frederick Brooks stated that program maintenance is an inherently entropy increasing process, and that even its most skilfull execution is only able to delay the subsidence of the system into unfixable obsolescence [2]. In practice, manifestations of this ever increasing difficulty in maintaining software is for instance reflected in terms of ever growing IT departments and rising IT maintenance costs [3], [4].

Specifically focusing on these issues of software maintenance and evolvability, *Normalized Systems (NS) theory* has recently proven to introduce a degree of proven ex-ante evolvability at the level of software systems. To start with, the theory states that the implementation of functional requirements into software constructs can be considered as the *transformation* of a set of requirements $\mathcal{R}$ into a set of software primitives $\mathcal{S}$ [5], [6], [7]. In order to reach evolvable modularity, NS theory demands that this transformation should exhibit systems theoretic *stability*. Mannaert et al. have formally proven in [6] that this assumption implies that the modular software structure should strictly and systematically adhere to four design *principles* as a necessary condition. In this paper, our goal is to explore the applicability of this other fundamental property of systems, i.e. *entropy*, to software maintenance and evolvability. Therefore, we propose a more precise definition of entropy in software systems, and explore how this notion of entropy might provide guidance to the software engineering process, in order to control the amount of entropy continually built up during the lifecyle of a software application. As an initial step towards design rules, we attempt to relate this guidance to the design principles of NS theory.

The remainder of this paper is structured as follows. In section II, we briefly discuss some related work. A concise overview of Normalized Systems Theory is presented in a

 ISBN: 978-1-61208-184-7

third section. In section IV, an attempt is made to derive an unambiguous definition of entropy in software architectures. Next, the feasibility of controlling the entropy in software systems is discussed in a fifth section, and the implications for software design are related to NS design principles. Finally, we present some conclusions and future work in Section VI.

II. Related Work

Entropy as expressed in the second law of thermodynamics, is considered to be a fundamental principle. There are many versions of this second law, but they all have the same intent, which is to explain the phenomenon of irreversibility in nature [8]. Moreover, mathematical derivations of the principle of entropy start in general from a formula describing the number of possible combinations. In statistical thermodynamics, entropy was defined by Boltzmann in 1872 as the number of possible microstates corresponding to the same macrostate [9]. In information theory, entropy was defined in 1948 by Shannon as the number of possible combinations or uncertainty associated with a random variable [10].

It has been attempted in many areas to apply and operationalize the concept of entropy, both inside and outside engineering sciences. In [11], Janow has studied organizations and productivity based on entropy. Janow concluded that entropy offered an interesting means to explain why organizations tend to become gradually more slow in their decisionmaking processes, as well as lose productivy and speed as they become larger over time. In the computing area, entropy is defined as the randomness collected by an operating system or application for use in cryptography or other uses that require random data [12], [13]. This randomness is often collected from hardware sources, either pre-existing ones such as mouse movements or specially provided randomness generators [14].

In software engineering, earlier attempts have been made to apply entropy concepts to software. For example, Harrison argued for an entropy-based metric for measuring the complexity of software applications, based on the information theory perspective on entropy [15]. Based on an analysis of existing complexity metrics for software, Bianchi et al. [16] propose a class of metrics aimed at assessing the amount of software degradation as an effect of continuous change.

Further, Manny Lehman considered his law of increasing complexity as an instance of the second law of thermodynamics [1]. Therefore, the law can also be interpreted as describing ever increasing entropy or disorder in the structure of software systems. This disorder or structure degradation hampers future adaptations of the software system, unless effort is done in order to reduce the amount of entropy [1], [17]. In a similar way, Frederick Brooks related the severe complexities of software engineering to the concept of entropy, and stated that program maintenance is an inherently entropy increasing process [2]. Consequently, as the theory on Normalized Systems [5], [6], [7] is aimed at understanding and controlling the law of increasing complexity, it can also be interpreted as an approach to controlling entropy, as will be further highlighted below.

III. Normalized Systems Theory

Specifically focusing on the issues of software maintenance and evolvability, *Normalized Systems (NS) theory* has recently proven to introduce a degree of proven ex-ante evolvability at the level of software systems. To start with, the theory states that the implementation of functional requirements into software constructs could be regarded as a *transformation* of a set of requirements $\mathcal{R}$ into a set of software primitives $\mathcal{S}$ [5], [6], [7]:

$$\{\mathcal{S}\} = \mathcal{I}\{\mathcal{R}\}$$

In order to limit the complexity of the evolvability analysis, it is argued in [6] that it is not a conceptual limitation to limit the software constructs to those of procedural programming languages, distinguishing data structures S_m and processing functions F_n. However, in order to study the evolvability, an additional variable needs to be introduced to represent the version of the programming constructs, both for data structures $S_{m,i}$ and processing functions $F_{n,j}$.

Further, in order to obtain evolvable modularity, NS theory demands that this transformation should exhibit systems theoretic stability, meaning that a bounded input function (i.e., bounded set of requirement changes) should result in a bounded output values (i.e., a bounded impact or effort) even if an unlimited time period and systems evolution is considered (in which the number of primitives and their dependencies becomes unbounded). Applied to information systems, this means that the impact of a change can only be dependent on the nature of a change itself. Alternatively, changes having impacts also depending on the size of the system are called *combinatorial effects* and thus should be avoided in order to obtain a stable software architecture. In fact, one could observe that the behavior of combinatorial effects seems to be very similar to the ever increasing complexity issue as coined by Lehman: a continuously growing number of changes including combinatorial effects, each of them exhibiting an ever increasing impact of N, would contribute to the ever increasing complexity. As such, Mannaert et al. [6] have formally proven that this implies that the modular software structure should strictly and systematically adhere to the following *principles* as a necessary condition:

- *Separation of Concerns*, enforcing that each change driver becomes seperated;
- *Data Version Transparency*, enforcing that communication between data is performed in a version transparant way;

 ISBN: 978-1-61208-184-7

- *Action Version Transparency*, requiring that action components can be updated without impacting their calling components;
- *Separation of States*, enforcing that each action of a workflow becomes seperated from other actions in time, by keeping state after every action.

These design principles show that current software constructs, such as for example functions and classes, by themselves offer no real mechanisms to accomodate anticipated changes in a stable way. Moreover, as the systematic application of these principles results in very fine-grained modular structures, NS theory proposes to build information systems based on the aggregation of instantiations of five higher-level software *elements*, i.e., action elements, data elements, workflow elements, trigger elements and connector elements [5], [6], [7]. The internal structure of every of these elements has been described in a very fine-grained way, proven to be free of combinatorial effects. Hence, by building normalized software applications based on an aggregation of instances of the different elements, the stable software structure of an application can be expanded, based on the internal structure of the elements.

While NS theory thus originally originated from applying the concept of systems theoretic stability on the transformation of (elementary) functional requirements into constructional software primitives, it seems reasonable to expect that the NS theory concepts can also be related to entropy, and that the principles can be interpreted as a means of controlling the amount of entropy built up during the lifecyle of a software application.

IV. Towards a Definition of Software Entropy

In this section we will first discuss some underlying concepts in general entropy definitions. Next, we will apply these concepts on software systems by proposing a definition for their macrostates and microstates.

A. Underlying Concepts in Entropy Definitions

Consider in more detail the statistical perspective on entropy as introduced by the Austrian physicist Ludwig Boltzmann. In statistical thermodynamics, the aim is to understand and to interpret the measurable macroscopic properties of materials – the macrostate – in terms of the properties of their constituent parts – the microstates – and the interactions between them. In Boltzmann's definition, entropy is a measure of the number of possible microstates of a system, consistent with its macrostate. It is basically the number of possible combinations of individual microstates that yield the same macrostate [18]. This notion of entropy can be seen as a measure of our lack of knowledge about a system. Consider for example a set of 100 coins, each of which is either heads up or tails up. The macrostates are specified by the total number of heads and tails, whereas the microstates are specified by the facings of each individual coin. For the macrostate of 100 heads of 100 tails, there is exactly one possible configuration, so our knowledge of the system is complete. At the opposite extreme, the macrostate which gives us the least knowledge about the system consists of 50 heads and 50 tails in any order, for which there are 10^{29} possible microstates. It is clear that the entropy is extremely large in the latter case because we have no knowledge of the internals of the system. An obvious mechanism to decrease the entropy or complexity, is to increase the *structure* and therefore knowledge of the internals. Suppose we would have 10 groups of 10 coins, each with 5 heads and 5 tails, the number of possible combinations or microstates would only be 2520 [18]. In summary, structure can be used to control entropy, in the sense that the microstates are known, which leads to less uncertainty and therefore, a kind of determinism.

Another example which is often suggested is that of a container with a boundary in the middle and containing a gas in the left part of the container. The macrostate is characterized by observable variables such as pressure and temperature, while the microstate is the union of the position, velocity, and energy of all gas molecules in the container. Once the boundary division is removed, the gas molecules will spread out over the entire container, increasing the number of possible microstates and therefore the amount of entropy. In summary, entropy increase is typically related to a process of *spreading out*, and intermediate boundary structures are a typical mechanism to avoid the increase of entropy.

B. Macrostates and Microstates in Software Systems

Starting from the generic definition of entropy as *the number of microstates for a given macrostate*, we first need to propose a definition for the concepts of macrostate and microstates. As a *macrostate* is in general an observable state of the system, and all source code is observable in a transparent way, it seems more promising to apply the concept of entropy to the run-time analysis of a software system, than to the compile-time analysis of the source code. Therefore, the observable macrostate seems to be related to information that is visible in output streams such as loggings, or observable system states such as database entries.

Concerning *microstates*, they could be defined in terms of the elementary instructions and data registers of the processor. However, as our purpose is to establish principles to guide the software engineering and architecture process, we propose to define the concept of microstates at the level of the constructs of the programming languages. As the correct and faultless execution of software programs is in general considered to be a worthwhile pursuit, we propose to define microstates as binary values representing the correct or erroneous execution of a construct of a programming language. However, it is important to clearly specify the *molecules* at this level, and to define clear boundaries between them. For instance, the run-time execution of a

 ISBN: 978-1-61208-184-7

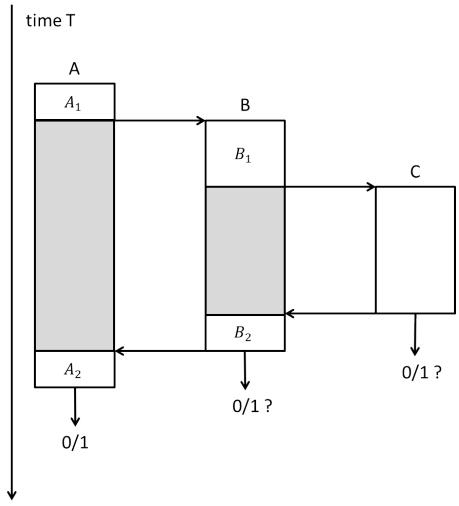


Figure 1. Synchronous pipelines.

procedural function can be defined as the execution of a specific procedural function operating on specific instances of the argument data structures, and resulting in a specific instance of an output data structure.

Moreover, these boundary divisions are not only needed between the various programming constructs at a certain point in time, but should also avoid coupling or leakage in the temporal dimension. For instance, the operations of a specific function could also be influenced by the values of global variables that each have received their specific value from other functions, or influence other functions by assigning values to such variables. Only when avoiding these effects, this function will not influence any other software molecule in any other way than through the specified output data arguments. In an object-oriented programming environment, all class member variables behave like global variables for the various methods of the class.

Consider Figure 1, representing an action entity A calling action entities B and (indirectly) C in a stateless way. Each of the seperate action entities A, B and C will generate a correct (0) or erroneous (1) outcome. In our representation, the seperate action entities can be regarded as the 'software molecules'. The correct or erroneous outcome of each of the subparts is defined as their microstate. The final outcome of action entity A, after executing B and C, can be regarded as the macrostate. Interestingly, one can note that the boundary division between the individual modules exhibits leakage in the temporal dimension. Indeed, action entity B (A) can only be succesfully completed after the execution of action entity C (B) and are hence interdependent. This inherently results in an increase in the amount of entropy: suppose for example that an error has occured as final outcome of A. This situation generates an uncertainty effect regarding where the actual error did occur. Did the returned final value originate in one of the two genuine parts of A, one of the two genuine parts of B, or C (or possibly a combination of them)? In other words, multiple combinations of microstates (correct or erroneous execution of the different (sub)activities) might generate the same macrostate (the resulting error as a final outcome of activity A), causing an increased degree of entropy or uncertainty.

In accordance with the evolvability analysis in [6], we only distinguish data structures $S_{m,i}$ and processing functions $F_{n,j}$. However, in order to represent the run-time state of the system, an additional variable k needs to be introduced representing the actual instance or value of the data structure $S_{m,i,k}$, and a variable l representing the program thread that is executing the processing function $F_{n,j,l}$. In order to control the complexity, which is widely accepted to be related to the concept of entropy, it seems reasonable to adhere to the straighforward extension of this model to an object-oriented programming environment, as proposed in [6], [7]. This means that a dedicated class is defined for every processing function, which is then the central method of this class, and for every data structure, which fields are the member variables of this class. Moreover, it is proven in [6], [7], in accordance with the *Separation of Concerns* principle, that no other functionalities should be added to those classes.

V. Towards Isentropic Software Architectures

In the previous section we discussed by means of a pedagogical example how entropy (and the according micro and macrostates) can be defined in the context of software systems. In this section, we will extend our analysis by illustrating the degree of possible entropy in real-life running software systems and which design rules might be formulated in order to control this entropy.

A. Entropy in a Running Software System

In order to illustrate the complexity of running software systems and its relation to our attempts in defining entropy, we start from the programming patterns for basic data entry operations in a multi-tier JEE (Java Enterprise Edition) architecture, as described and elaborated in [19], [20], [7]. Every data element or table `Obj` uses the same simple patterns for the various operations manipulating data, such as create, update, retrieve, delete, and search. For instance, in order to create a new instance, the webtier MVC (Model View Controller) framework calls the method `act` on an `ObjEnterer` class. The call is related to the application tier through a method `create` in an `ObjAgent` class, calling remotely through RMI (Remote Method Invocation) the `create` method in an `ObjBean` class residing in the application server. The various calls pass to each other instances of a serializable `ObjDetails` class, that contains the actual values of the various fields or attributes.

 ISBN: 978-1-61208-184-7

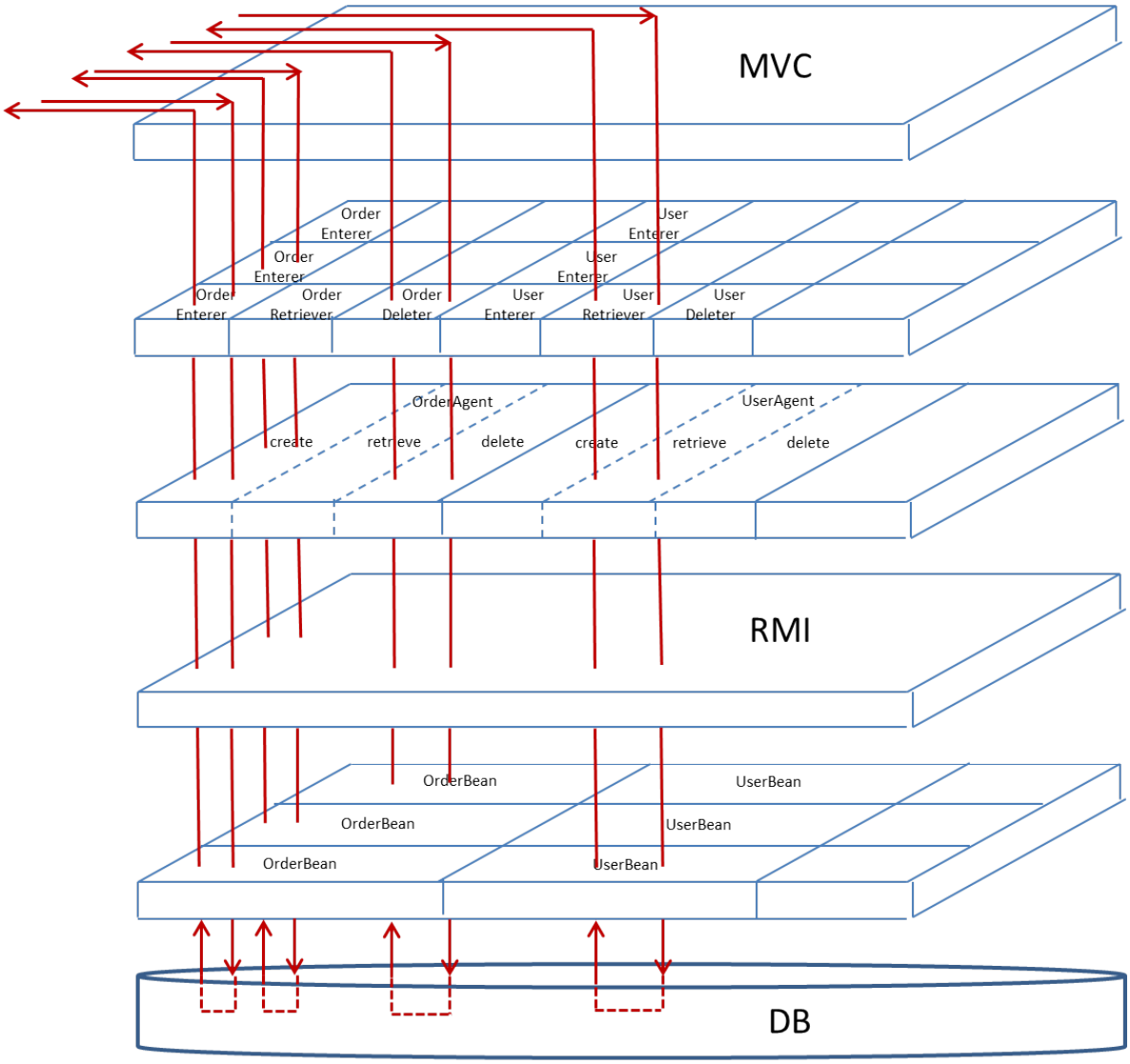


Figure 2. 3D visualization of a *state space* representing a running data entry system.

While this programming pattern is quite straightforward at compile-time, it is much more complicated at run-time in a multi-tier and multi-threaded system. Figure 2 shows a 3D visualization of the state space of such a running system, distinguishing three dimensions:

- The *functional dimension* horizontally, representing the various data objects (Order, User, ...) and operations (create, retrieve, delete, ...).
- The *multi-tier dimension* vertically, representing the various tiers: the webtier controlled by the MVC, the application tier called through RMI, and the data tier.
- The *multi-thread dimension* in the depth, representing the various threads of the incoming calls.

The drawing also represents the thread instantiation pattern of this simple software pattern for data entry. Every incoming thread leads to the creation of a new and separate instance of the appropriate action class, like `OrderEnterer`. The agents, like `OrderAgent`, are designed as singletons, and are a single point of access for that data element to relay the calls to the application tier. In the application tier, every call leads again to the creation of a new and separate instance of a bean class, like `OrderBean`, though the same class groups the various operations on a single data element or table.

The aim of this section in general and this drawing in particular is not to present a high quality pattern that should be preferred over other existing or conceivable patterns. Its goal is to explain that the run-time state of a software system in general and a programming pattern in particular, has many facets that are in general not explicitated in a compile-

 ISBN: 978-1-61208-184-7

time software pattern. And it is this complicated run-time statespace that needs to be analyzed and mastered, in order to control the number of possible microstates, and therefore the entropy of the software system.

B. Design Rules for Controlling Entropy

As our aim is to use the concept of entropy as guidance for software engineering and architecture, we have proposed to use the instantiations of programming constructs, such as functions or class methods, as the molecules or basic building blocks representing the microstates of the system. Knowing that local variables cease to exist after the processing function has been completed, and assuming that we do not allow hidden coupling or information leakage through all sorts of global variables, the microstate of such a function — being a correct or erroneous execution — can be studied in isolation from the rest of the system.

The synchronous pipelines that exist in Figure 2 in relaying the calls through the various tiers of the architecture, seem to pose a problem. Indeed, as the `act` method of the `OrderEnterer` is only completed after the calls to the other tiers, its microstate cannot be studied in isolation from these other processing methods. Splitting the method in its two parts, before and after the remote call, would introduce coupling between those parts through all the local variables of the method. Therefore, the *Separation of States* principle, as derived in [6], [7], seems to be in accordance with the concept of software entropy control. Indeed, if seperation of states is not adhered to (e.g., synchronous pipelines), it is not clear or uncertain where exactly a certain error or exception has occured in the modular structure. In such architectures, the macrostate (i.e., an error has occured) can be explained through many possible microstates (i.e., many possible causes related to many possible atomic tasks which are not seperated by keeping state) and by definition the amount of entropy in the software system increases. Of course, the use of synchronous pipelines in this example is quite innocent, as it is limited to relaying a simple data entry call through the various tiers of the architecture.

Also, because the essence of controlling entropy is the reduction of uncertainty, and the operations of a software system are described in terms of the various *tasks* or *concerns*, the *Separation of Concerns* principle (as derived in [6], [7]) seems to be in full accordance with the proposed concept of software entropy as well. Indeed, encapsulating every task or concern in a separate programming construct, would allow us in general to externalize the state of every task. Exporting every such microstate — through detailed loggings for instance — to the observable macrostate, would avoid the creation of multiple microstates for the same macrostate, and would by definition avoid the creation of entropy. Finally, while from a stability point of view the identification of concerns should be based on so-called change drivers, the identification of concerns from an entropy point of view should be based on so-called uncertainty drivers. While theoretically, concerns identified from one point of view or the other might turn out to be different, we anticipate that generally both perspectives will lead to the identification of the same concerns in practice.

Two other principles derived in [6], [7], *Data Version Transparency* and *Action Version Transparency* manifest themselves at compile-time, and therefore seem less relevant in this run-time analysis. However, the isentropic requirement demanding the ability to study every microstate of a processing function in isolation of the rest of the system, calls for the export of all microstate details to an observable macrostate. More specifically, this implies the following rules.

- *Data Instance Traceability:* the actual version and values of every instance of a data structure serving as an argument, need to be exported to an observable macrostate.
- *Action Instance Traceability:* the actual version of every instance of a processing function and the thread it is embedded in, need to be exported to an observable macrostate.

These observable macrostates may be implemented through a wide range of possibilities, ranging from simple loggings to more elaborate database entries.

VI. Conclusion and Future Work

Triggered by the identified need for evolvable software applications, Normalized Systems theory has previously been able to introduce a proven degree of evolvable modularity into software systems, based on the systems theoretic notion of stability. In this paper, we explored the applicability of this other fundamental property of systems, i.e. entropy, to the issues of software maintenance and evolvability. First, the underlying concepts of the statical perspective on entropy were applied to and defined specifically in the context of software systems. More specifically, the macrostate was related to observable output streams of the software system, and the microstates were defined as a set of binary values representing the correct or erroneous execution of programming constructs. Next, the complexity of real-life multi-tier software applications in terms of entropy was illustrated. In order to control the amount of entropy, some design rules were explored and related to the existing design principles of NS theory. This suggested an initial concordance regarding principles for software maintainability based on applying concepts from (1) systems theoretic stability and (2) entropy from thermodynamics.

Acknowledgment

P.D.B. is supported by a Research Grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

 ISBN: 978-1-61208-184-7

REFERENCES

[1] M. Lehman, "Programs, life cycles, and laws of software evolution," *Proceeding of the IEEE*, vol. 68, no. 9, pp. 1060–1076, 1980.

[2] F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, 1975.

[3] B. P. Lientz and E. B. Swanson, "Problems in application software maintenance," *Communications of the ACM*, vol. 24, pp. 763–769, 1981.

[4] Standish Group, "The standish group report: Chaos," Tech. Rep., 1994.

[5] H. Mannaert and J. Verelst, *Normalized systems: re-creating information technology based on laws for software evolvability*. Koppa, 2009.

[6] H. Mannaert, J. Verelst, and K. Ven, "The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability," *Science of Computer Programming*, vol. Article in press, 2011.

[7] ——, "Towards evolvable software architectures based on systems theoretic stability," *Software Practice and Experience*, vol. Early View, 2011.

[8] Wikipedia. (2011) Second law of thermodynamics. [Online]. Available: http://en.wikipedia.org/wiki/Second_law_of_thermodynamics

[9] L. Boltzmann, *Lectures on gas theory*. Dover Publications, 1995.

[10] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[11] R. Janow, "Shannon entropy and productivity: Why big organizations can seem stupid," *Journal of the Washington Academy of Sciences*, vol. 90, 2004.

[12] Wikipedia. (2011) Entropy (computing). [Online]. Available: http://en.wikipedia.org/wiki/Entropy_(computing)

[13] C. Cachin, "Entropy measures and unconditional security in cryptography," Ph.D. dissertation, Swiss Federal Institute of Technology Zürich, 1997.

[14] Z. Gutterman, B. Pinkas, and T. Reinman, "Analysis of the linux random number generator," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, 2006.

[15] W. Harrison, "An entropy-based measure of software complexity," *Software Engineering, IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 1025–1029, nov 1992.

[16] A. Bianchi, D. Caivano, F. Lanubile, and G. Visaggio, "Evaluating software degradation through entropy," in *Proceedings of the 7th International Symposium on Software Metrics*, ser. METRICS '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 210–219. [Online]. Available: http://dl.acm.org/citation.cfm?id=823456.823991

[17] G. Visaggio, "Assessing the maintenance process through replicated, controlled experiment," *The Journal of Systems and Software*, vol. 44, no. 3, pp. 187–197, 1999.

[18] Wikipedia. (2011) Entropy. [Online]. Available: http://en.wikipedia.org/wiki/Entropy

[19] H. Mannaert, J. Verelst, and K. Ven, "Towards rules and laws for software factories and evolvability: A case-driven approach," in *Proceedings of the First International Conference on Software Engineering Advances (ICSEA)*. IEEE Press, 2006.

[20] ——, "Exploring concepts for deterministic software engineering: Service interfaces, pattern expansion, and stability," in *Proceedings of the Second International Conference on Software Engineering Advances (ICSEA)*. IEEE Press, 2007.

 ISBN: 978-1-61208-184-7

MDE-based QoS Management Framework for RTDB Management Systems Development

Salwa M'barek, Leila Baccouche, Henda Ben Ghezala
RIADI-GDL laboratory
INSAT, National Institute of Applied Science and Technology
C.U. Nord, B.P 676, Tunis Cedex 1080, Tunisia
salwa.mbarek@riadi.rnu.tn leila.baccouche@insat.rnu.tn henda.bg@cck.rnu.tn

***Abstract*—This paper sets out a framework for real-time database management systems (RTDBMS) model design integrating QoS management. We use a Model Driven Engineering (MDE) approach based on model transformation techniques. The aimed systems apply the feedback control scheduling for QoS management which gives a robust and controlled behavior of the system even in transient overloads. The framework provides metamodels and processes to extend, reuse and transform RTDBMS models for different QoS requirements and different real-time applications. A RTDBMS design tool has been developed based on EMF (Eclipse Metamodeling Framework) and Kermeta metamodeling and transformation language.**

Keywords-real-time database management systems; QoS management; feedback control scheduling; MDE; model transformation ; Kermeta.

I. INTRODUCTION

The real-time database management systems (**RTDBMS**) are database management systems manipulating real-time data and real-time transactions with time constraints [4, 15].

The *real-time data* must be updated periodically by *real-time update transactions* to reflect the real world state at any time, otherwise they become unfresh which may cause a disaster.

The *real-time user transactions* which have to access real-time data, must be executed within a deadline otherwise they become useless for the application.

Recent works in RTDBMS [1, 2, 3, 8, 9, 10, 16] propose QoS management architectures and QoS management algorithms based on the *Feedback Control Scheduling Architecture* (**FCSA**) to give a robust and controlled behavior of the RTDB even during transient overloads and when we have inaccurate run-time estimates of the transactions [12].

We propose a model design framework for RTDBMS using FCSA, which is based on the **MDE** (*Model Driven Engineering*) approach. This new approach for software systems engineering is centered on the models and not on the implementation [11].

Our aim is to support designers to easily set up the appropriate model of the RTDBMS with a QoS management approach based on the Feedback Control Scheduling. Moreover, to satisfy new real time requirements, persistent RTDBMS models can be easily reused, extended and combined based on model transformation techniques.

The framework provides metamodels and processes for RTBDMS model and code generation.

This paper begins in Section 2 with an overview of the MDE (*Model Driven Engineering*) approach. Section 3 sets out the Feedback Control Scheduling Architecture (FCSA) for QoS management in RTDBMS. The metamodel of Feedback Control Loops is presented in Section 4. Section 5 gives a metamodel of QoS management approaches in RTDBMS. Section 6 illustrates the proposed MDE-based framework for RTDBMS development. The Model transformation to integrate the QoS management to basic RTDBMS is explained in Section 7. We conclude this paper by a summary of contributions and perspectives.

II. MODEL DRIVEN ENGINEERING

The *model-driven engineering* (**MDE**) approach has allowed several significant improvements in the development of complex systems by putting the focus on a more abstract concern than the classical programming. It is a form of generative engineering in which (all or part of) an application is generated from models [5, 6].

Modeling allows the generation of parts of an application instead of implementing the source code manually. This increases the development speed and even more importantly, it increases the implementation quality. Models can be checked for consistency before source code is created from them. If an application evolves, changes only have to be applied in the model, while the source code can be re-generated automatically.

Models provide a higher level of abstraction than source code. Developers can focus on key aspects of an application, instead of dealing with the complexities inherent in a programming language. The creation of custom models, so-called Domain-Specific Languages (DSL), can make the application understandable without a background in programming.

The Eclipse Modeling Top-Level Project facilitates MDE for Eclipse and is one of the biggest and most active areas in the Eclipse ecosystem. The Eclipse Modeling Framework (EMF) builds the foundation for a variety of modeling

 ISBN: 978-1-61208-184-7

technologies such as the Graphical Modeling Framework (GMF) or textual modeling (XText), and has become a widely used standard for modeling worldwide.

III. FEEDBACK CONTROL SCHEDULING ARCHITECTURE

The *Feedback Control Scheduling Architecture* (**FCSA**) as visualized in Figure 1, gives a robust and controlled behavior of the RTDBMS even during transient overloads and when we have inaccurate run-time estimates of the transactions [12].

This architecture is based on the following principle: "***observation then auto-adaptation***". The database administrator defines some parameters and their thresholds to give the QoS specification. For instance the *Miss Ratio (MR)* is a QoS parameter which measures the percentage of user transactions that missed their deadlines (MR ≤ 20%). The observation consists on measuring periodically the *QoS parameters* to compute the system performance.

The *auto-adaptation* consists on adjusting the system behavior, using *control loops*, *update policies* and *QoS management algorithms.*

The Figure 1 shows the FCSA as proposed in [10]. It consists of several main components:

Sources generate *user transactions* to be submitted to the system. Each *Update Stream* periodically submits an *update transaction* for a certain temporal data object (real-time data). *Admission control* is applied to *user transactions*.

Transaction handler consists of a *concurrency controller* (CC), a *freshness manager* (FM) and a *basic scheduler*.

Update transactions with *highest importance* are scheduled in the *high priority ready queue* while *user transactions* and *Update transactions* with *lowest importance* are scheduled in the *low priority queue*. In each queue, transactions are scheduled in **EDF** (Earliest Deadline First) manner. A transaction can be aborted and restarted by CC. It can also be preempted by a higher priority transaction. *Freshness manager* (FM) checks the freshness of real-time data before the initiation of a *user transaction*. FM blocks the corresponding transaction if an accessing data is currently stale. The blocked transaction(s) will be transferred from the block queue to the ready queue as soon as the corresponding update commits.

Monitor periodically measures ***QoS parameters*** (miss ratio, utilization, and perceived freshness) and reports the statistics to the *feedback QoS controllers* (MR/Utilization Controllers) and *QoD manager*. *QoS controllers* compute the ***control signals*** **ΔU** based on the current performance error using the **PID control** (Proportional Integral Derivative control) [14, 15].

QoD Manager adapts the *update policy*, if necessary. It informs the *admission controller* of the new ***control signal*** **ΔU** **(ΔU_{new})** after potential QoD adaptations. *Update scheduler* decides whether or not to schedule an incoming

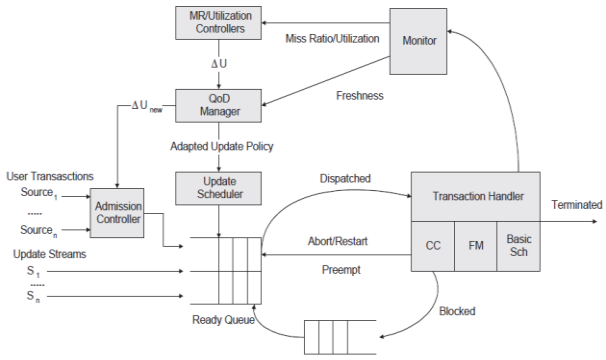


Figure 1 : Feedback Control Scheduling Architecture (FCSA)

update depending on the selected *update policy*.

IV. THE FEEDBACK CONTROL LOOPS METAMODEL

An important step in designing the FCSA of an RTDBMS is to decide the following concepts: controlled variables, performance reference, control signal, manipulated variable, control loop and control function.

1. *Controlled variables* are the performance metrics controlled by the scheduler. Controlled variables of a real-time system may include the *deadline miss ratio M*(*k*) and the *CPU utilization U*(*k*) (also called miss ratio and utilization, respectively), both defined over a time window ((*k*-1)*W*, *kW*), where *W* is the *sampling period* and *k* is called the *sampling instant*. The miss ratio *M*(*k*) at the k_{th} sampling instant is defined as the number of deadline misses divided by the total number of completed and aborted tasks in a sampling window ((*k*-1)*W*, *kW*). Miss ratio is usually the most important performance metric in a real-time system.

The utilization *U*(*k*) at the k_{th} sampling instant is the percentage of CPU busy time in a sampling window ((*k*-1)*W*, *kW*). CPU utilization is regarded as a controlled variable for real-time systems due to cost and throughput considerations. CPU utilization is important because of its direct linkage with the deadline miss ratio [1, 3, 10].

2. *Performance reference* is a target value specified by the DBA for a specific *controlled variable*. Each controlled variable must converge to its performance reference (reference). For instance: in steady state, the controlled variable MR must be less than 30% so its reference is MR_r =30%. An overshoot noted Mp is allowed in transient overloads; so that

$$MR \leq MR_r \times (M_p + 100)\%$$

3. *Control loop is a closed loop* using a *control function* to generate a performance adjusting signal called a *control signal*. The entry of the control loop is the error e(t) between the target value of a controlled variable and its current value. There is a control loop for each controlled variable. The unique control signal generated by the QoS Controller is derived from all control signals generated by its control loops.

$$\mathrm{e(t)} = controlled\ variable\ reference - measured\ controlled\ variable$$

4. *Control signal* (generally noted ΔU) is provided periodically by control loops of the ***QoS Controller***. It is computed by a certain *control function* based on the error e(t) between the target values of the controlled variables (performance reference) and their measured values. In [1], the control signal ΔU which is the requested CPU utilization adjustment is computed as follows.

$$\Delta U(t) = \boldsymbol{k_p}.\mathrm{e}(t) + \boldsymbol{k_i}.\int_0^{\mathrm{t}} \mathrm{e}(t).\mathrm{dt} + \boldsymbol{k_d}.\frac{\mathrm{de}(t)}{\mathrm{dt}}$$

k_p, k_i, k_d are PID parameters

$$\mathrm{e(t)} = MR_r - MR(t)$$

5. *Control function* represents the relation between the *control signal* $\Delta U(\mathrm{t})$ and the *controlled variable error e(t).*

$$\Delta U(t) = \boldsymbol{f}(e(t))$$

$\mathrm{e(t)} = V_r - V(t)$; *V is the controlled variable*

6. *Manipulated variable* is a QoS parameter which has an impact on the performance of the system and the controlled variables. Its value must be adjusted dynamically to guarantee QoS specification and so system robustness. For instance, the data freshness has an impact on the miss ratio MR, so, it can be considered as a manipulated variable. In fact, decreasing the number of update transactions will degrade data freshness. Consequently, the number of completed user transaction will increase and so the average miss ratio MR will decrease.

Auto-adaptation consists on adjusting (decreasing or increasing); by a certain function; the value of the *manipulated variable* depending on the value of the *control signal* ΔU which is computed from *Controlled variables errors* (based on PID function or other control function).

The *QoS management algorithm* makes, at each sampling period, the *Auto-adaptation*. It is running on the *QoD Manager* which is considered as the *QoS regulator*. Regulation orders come from *QoS controller* which sends him the *control signal.*

We propose a metamodel to design feedback control loops for QoS management in RTBDMS presented in Figure 2. This metamodel establish relations between the different concepts explained in this section.

From this metamodel, different models can be generated for specific requirements.

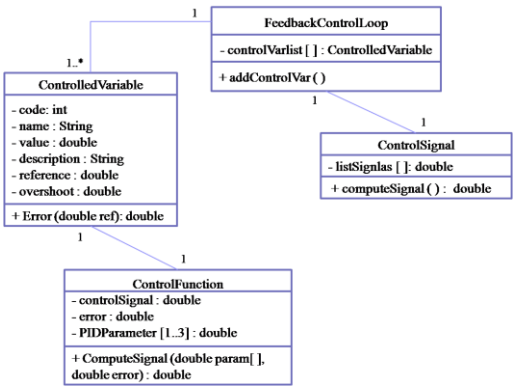


Figure 2 : Feedback Control Loops metamodel

V. QoS MANAGEMENT APPROACHES METAMODEL

All studied works [1, 2, 3, 8, 9, 10, 16] use the Feedback Control Scheduling Architecture. However, we notice many differences on their QoS management approaches.

They propose different QoS metrics. The specification of feedback control loops in the QoS controller varies from an approach to another. They are applying the PID control (Proportional Integral Derivative Control) [14, 15] as a control function, but there are some differences in their formula.

Each approach use only two kinds of transactions: user transactions and update transactions with firm deadlines (if they miss their deadlines, they will simply be rejected from the system).

Compared approaches use different transaction models. In [1, 7], they propose the milestone model where a transaction is decomposed into one mandatory sub-transaction which must obligatory meet its deadline and many optional sub-transactions which can be rejected in overload situations without affecting QoS specification.

However, in [8, 12] they use a service differentiation. They don't decompose transactions, they classify them in three service classes regarding to an importance factor.

All approaches consider only base data which hold the view of the outside environment, in opposition to derived data which are derived from other base or derived data.

Each approach uses a specific Data model and update policy [8, 9, 12].

Even transactions queues are configured differently. Queues configuration depends on transactions model.

In these approaches, are applied different *update policies* (adaptive policy, MDE policy) and different *scheduling*

 ISBN: 978-1-61208-184-7

algorithms (EDF: Earliest Deadline First, HEF: Highest Error First, HEDF: Highest Error Density First).

Many QoS management algorithms are proposed such as: FCS-IC1(Feedback Control Scheduling-Imprecise Computation-1) [1]; FCS-IC2 (Feedback Control Scheduling-Imprecise Computation-2) [1]; FCS-HEF (Feedback Control Scheduling-Highest Error First) [13]; FCS-HEDF (Feedback Control Scheduling- Highest Error Density First) [13], QMF1 (QoS-sensitive approach for Miss ratio and Freshness guarantees 1) [10] and QMF-Diff (QoS-sensitive approach for Miss ratio and Freshness guarantees with Differentiated Services) [10].

However, these algorithms are little similar. These algorithms try to balance the system load between user transactions and update transactions. For example, in overload situations, the QoD is decreased applying the corresponding QoS management algorithm until steady state reaching, where QoD will be increased without QoT violating. When the system is saturated, all arrived transactions are discarded by the admission controller [3, 9].

An evaluating study of these approaches and algorithms is detailed in [17].

We propose a metamodel of QoS managements approachs as shown in the following figure. Based on this metamodel we can derive any QoS management approach model for specific QoS requirements.

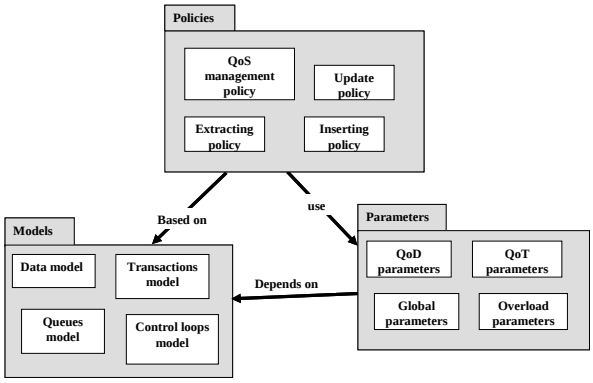


Figure 3 : QoS management Approachs metamodel

VI. PROCESSES FOR RTBDMS DEVELOPMENT

Existing QoS management approaches are very interesting. However, it is difficult to reuse them or a part of them. Furthermore, it is very difficult to develop a new RTDBMS architecture from scratch, to extend or to reconfigure an existing architecture, to modify the QoS management algorithm or to add other QoS parameters and QoS specification.

The proposed framework tries to answer these issues. We are interested only in RTDB management systems with feedback control scheduling, because they are complex

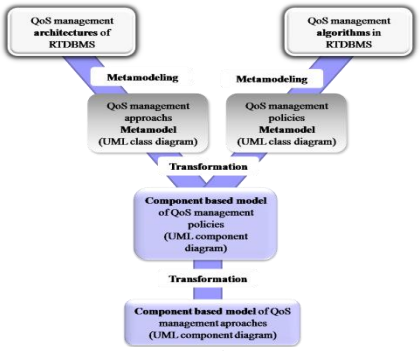


Figure 4 : Y-process for model generation

systems but at the same time, they are robust systems offering QoS constraints specification and management.

Our first aim was to design a metamodel for QoS management loops and then a metamodel for QoS management approaches.

Metamodels will be transformed to generate abstract component based models for QoS management in RTDBMS: Component based model of QoS management policy and Component based model of QoS management approach.

The component based models resulting from the Y-process are the entry to the second process (as shown in Figure 7) which allows the reuse of these models to build new ones and to generate the implementation into a specific language.

We built a three-layered database (Figure 7) for QoS models reuse and code generation. Component based models of QoS management approaches and policies are stored in the "Models level" of the database with platform models (J2EE…). Models are decomposed and components are stored separately in the "Component level". The data about metamodels, models, components and bindings are stored in the "Metadata level".

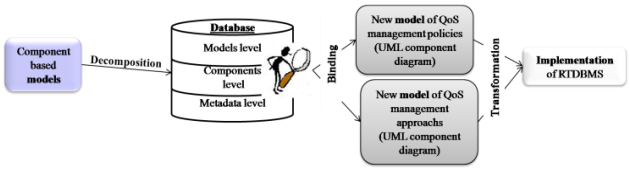


Figure 5: Model reuse process

VII. FRAMEWORK IMPLEMENTATION

Every real time database management system can be modeled using our metamodels. The QoS management layer can be added through models transformation as shown in the following figure. We used the Eclipse Modeling Framework tool to implement different metamodels conformant to the Ecore metametamodel and to generate models in the XMI (XML Metadata Interchange) format. The Kermeta langage is used to load and transform models.

 ISBN: 978-1-61208-184-7

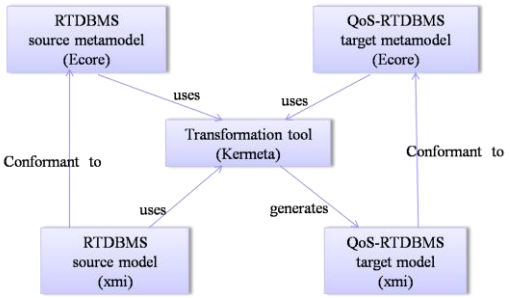


Figure 6 : Models transformation for QoS integation

A graphical user interface with the java language is under development to facilitate the RTDBMS model design and transformation for designer not expert at EMF and Kermeta.

VIII. CONCLUSION

This paper focused on the real-time database management systems (RTDBMS) using the Feedback Control Scheduling Architecture (FCSA) for QoS and time constraints management. Studied architectures are interesting but can't be easily reconfigured, extended or reused. To answer these issues, we proposed a model driven framework for QoS-aware RTDBMS development based on the Model Driven Engineering principles and the Model Driven Architecture standards. It provides processes, metamodels, component-based models, models transformation and models repository. It is possible to generate new approaches and QoS policies from stored ones or from scratch. Generated models are component based for two reasons: (1) make easy the reuse and reconfiguration of models (2) generate a code for component oriented platforms (J2EE). Object-oriented or aspect-oriented code may be generated through mapping between considered metamodels.

REFERENCES

[1] M. Amirijoo, "Algorithms for Managing Real-time Data Services Using Imprecise Computation", Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA), Taiwan, 2003.

[2] E. Bouazizi C. Duvallet and B. Sadeg, "Using Feedback Control Scheduling and Data Versions to enhance Quality of Data in RTDBSs". Proc. of IEEE International Computer System and Information Technology (IEEE ICSIT'2005), Alger, Algérie, 2005, pp. 322-327.

[3] E. Bouazizi, C. Duvallet and B. Sadeg, "Une nouvelle approche pour la gestion de la QdS dans les SGBD temps réel", Proc. of INFORSID'2006, Hammamet, Tunisie, 2006, pp. 547-559.

[4] L. Cingiser, H. Son and K. Ramamritham, "Real-Time Databases and Data Services", Journal of Real-Time Systems, 28, 179-215, 2004.

[5] B. Combemale, "Approche de métamodélisation pour la simulation et la vérification de modèle : Application à l'ingénierie des procédés", PhD Thesis, Institut National Polytechnique de Toulouse, 2008.

[6] B. Combemale, "Ingénierie Dirigée par les Modèles (IDM) : État de l'art", hal-00371565, 2008.

[7] J. Den Haan, "MDA and Model Transformation", 2008. http://www.theenterprisearchitect.eu/archive/2008/02/18/mda-and-model-transformation.

[8] C. Duvallet, E. Bouazizi and B. Sadeg, "Improvement of QoD and QoS in RTDBS". Proc. 14th International Conference on Real-Time and Network System (RTNS'2006), Poitiers, France, 2006, pp. 87-95.

[9] J. Hansson, M. Amirijoo and S. H. Son, "Specification and Management of QoS in Real-Time Databases Supporting Imprecise Computations". IEEE Transactions on Computers, 2006, V. 55, No. 3.

[10] K. Kang, S. Son A. Stankovic, "Managing Deadline Miss Ratio and Sensor Data Freshness in Real-Time Databases". IEEE Transactions on knowledge and data engineering, Vol. 16, No. 10, 2004, p. *1200-1216.*

[11] S. Kent, "Model Driven Engineering", IFM 2002, 2002, p. 286-298.

[12] C. Lu, A. Stankovic, G. Tao and H. Son, "Feedback control real-time scheduling: Framework, modeling and algorithms", Journal of Real-Time Systems, vol.23, n. 1, 2002, p.85-126.

[13] O. Patrascoiu, "Model Transformations in YATL". Studies and Experiments, 2004, Technical Report 3-04.

[14] J.D. Poole, "Model-Driven Architecture: Vision, Standards And Emerging Technologies", ECOOP 2001, 2001.

[15] K. Ramamritham, "Real-Time Databases", International Journal of Distributed and Parallel Databases, 1996.

[16] B. Sadeg, C. Duvallet and E. Bouazizi, "Prise en compte des données dérivées temps réel dans une architecture de contrôle par retroaction". Proc. of MAJECSTIC'2006, 2006.

[17] S. M'barek, L. Baccouche and H. Ben Ghezala, "An evaluation of QoS management approaches in Real-Time Databases". ICONS 2008, 2008, p. 41-46.

[18] S. H. Son, M. Amirijoo and J. Hansson, "Specification and Management of QoS in Imprecise Real-Time Databases", IEEE Database Engineering and Applications Symposium (IDEAS), Hong Kong, 2003.

[19] D. Xue, Y. Chen and D.P. Atherton, "PID Controller Design", Linear Feedback Control, Chapter 6, 2007. Society for Industrial and Applied Mathematics.

[20] M.J. Willis, "Proportional-Integral-Derivative Control", 1999.

 ISBN: 978-1-61208-184-7

Examining Challenges in IT Service Desk System and Processes: A Case Study

Marko Jäntti
University of Eastern Finland, School of Computing
Software Engineering Research Unit
P.O.B 1627, Kuopio, Finland
Email: marko.jantti@uef.fi

***Abstract*—The adoption of IT Infrastructure Library (ITIL) framework is a challenging task for many IT service provider organizations. Many government organizations in Finland have also started to use ITIL and need help in configuring tools and defining processes. The research problem of this study is: What types of challenges exist in the IT service provider's customer support? The main contribution of this paper is to present challenges on IT service support of Finnish Tax Administration.**

***Keywords*-incident management; service support; IT service.**

I. INTRODUCTION

Many IT service provider organizations consider the improvement of IT service management processes as a difficult and challenging task. Typically, the process improvement is based on the processes and methods of the IT Infrastructure Library (ITIL). ITIL is the most widely used IT service management framework consisting of a set of best practices for managing IT services.

The service management section of the ITIL version 2 consists of two parts [1]: 1) Service Delivery (Service level management, IT financial management, availability management, capacity management, IT service continuity management) and 2) Service Support (service desk function, incident management, problem management, change management, configuration management and release management). The ITIL version 3 emphasized the service lifecycle thinking and introduced five core books: Service Strategy [2], Service Design [3], Service Transition [4], Service Operation [5] and Continual Service Improvement [6]. The recent update 3.1 did not introduce very radical changes to Service Operation processes. However, the Service Strategy book was completely rewritten.

There are several factors that might prevent an effective process improvement. First, the companies have to use external ITIL consultants in providing training for their employees. These consultants know the ITIL framework and IT service management concepts very well but have limited knowledge on the existing business concepts, methods, tools, services, and the structure of service desk groups. Second, inadequate or too complex IT service management tools shall slow down any IT service management process improvement initiative. Third, lack of process culture and process thinking is very common phenomenom among IT companies. ITIL is a process oriented framework. Thus, the ITIL implementation team should be well-trained and have excellent process improvement and change management skills. Finally, lack of management support for ITSM project may cause that an organization do not allocate enough resources for the process/tool improvement. Besides allocating enough resources for improvement work, the management has to motivate and reward people who pass ITIL certificate exams and participate in IT service management work.

Sharifi et al. [7] have explored why ITIL implementations fail. They list at least the following factors: Spending too much time on complicated process diagrams, Not creating work instructions, Not assigning process owners, Concentrating too much on performance, being too ambitious, allowing departmental demarcation, ignoring or eviewing of the ITIL every time, and not memorizing ITIL books self. Mohamed et al. [8] have integrated knowledge management elements to the IT service management. Moreover, Lahtela et al. [9] have explored how to measure IT service management processes ín practices.

Peppard categorizes information system and technology (IST) services into four categories [10]: *Application services* are services that are delivered via software applications. *Operational services* maintain the IT environment including installation services for hardware and software, change management, and problem shooting services. *Value-enabling services* increase the value of of information assets (e.g. consulting, system design, and help desk). Finally, *infrastructure services* focus on creating an effective IT infrastructure including, for example, security and capacity issues. Bardhan et al. [11] state that IT services have aspects such as the high degree of involvement by people in delivery and that they are more or less intangible.

In order to improve IT service management processes, organizations can use various IT service management frameworks, such as the Control Objectives for IT and related Technology (COBIT) framework [12], Microsoft Operations Framework (MOF) [13], Kapella's Framework for Incident Management and Problem Management [14], IT Service Capability Maturity Model [15] or IT service management standard ISO/IEC 20 000 [16].

More and more academic studies are being published in the field of IT service amanagement, such as incident management problem management [17], success factors in

 ISBN: 978-1-61208-184-7

ITIL implementations [18]. The main contribution of this paper is to describe the challenges regarding IT service support and maintenance of Finnish Tax Administration.

The results of this study might be useful for service managers, service desk managers and IT service management process managers. The remainder of the paper is organized as follows. In Section II, the research methods of this study are described. In Section III, challenges in service desk tool and processes are presented. Section IV is the analysis of findings. The discussion and the conclusions are given in Section V.

II. Research Problem & Methodology

This case study is a part of the results of KISMET (Keys to IT Service Management and Effective Transition of Services) project. The research problem of this study is: What types of challenges exist in the IT service provider's customer support?

According to Yin [19], a case study is "an empirical inquiry that investigates a contemporary phenomenon within its real-life context". Eisenhardt [20] defines a case study research as "a research strategy focusing on understanding the dynamics present within single settings". The settings in this paper mean the customer support environment of Tax Administration. A case study research method with a single case was used to answer the research problem. Figure 1 describes the research settings of the case study. The study was carried out in Finnish Tax Administration's Kuopio unit.

Figure 1. The research settings of the case study

A. The Case Organization and Data Collection Methods

In Finland, taxation is carried out mainly by four organizations: Ministry of Finance, Tax Administration, Customs Finland and TraFi (Traffic safety agency). Our case organization is the Information System Management unit of Finnish Tax Administration that provides IT services (e.g. desktop services, service desk) to the tax administration staff. The organization had 5336 fulltime employees in 2010. The organization used a phased approach for implementing service management processes. In the first phase, the focus was on incident management and the ITIL-based service desk service was launched in Spring 2011.

The case study was carried out in August-October 2011. In order to increase the quality of the case study, case study researchers can use three important principles of data collection: 1) using multiple sources of evidence: three researchers participated in data collection from several sources 2) creating a case study datastore (a case study diary) 3) maintaining a chain of evidence (linking observations to data sources). The following sources of evidence were used:

- Documentation from the case organization (e.g. incident management process description, service support metrics, ITSM tool user guide, service catalogue, service area, event management material, error handling guide).
- Archives (service classification schema, incident and service request records)
- Interviews/discussions (discussions in work meetings between a research team and the case organization, informal coffee table discussions with service support workers, email conversations with process managers)
- Participative observation (proces improvement meetings and workshops (CSI workshop 27 September) and ITSM process trainings (45 minutes ITSM Introduction, 3 hour Basics of ITSM) organized by the KISMET research team)
- Fysical artefacts: Service desk tool, intranet

B. Data Analysis Method

A within case analysis technique [20] was used to analyze the collected case study data. Researcher triangulation was used in data analysis. Three case study researchers participated in the data collection and analysis. The within-case analysis focuses on examining cases carefully as stand-alone entities before making any generalizations.

III. Challenges in IT Service Desk

We used KISMET (Keys to IT Service Management Excellence Technique) model as a process improvement tool. The model consists of the following seven phases: Create a process improvement infrastructure, Perform a process assessment, Plan process improvement actions, Improve/Implement the process based on ITSM practices,

 ISBN: 978-1-61208-184-7

Deploy and introduce the process, Evaluate process improvement and Continuous process improvement.

In this paper, we focus on the 'Perform a process assessment' phase and present the customer support challenges that were identified during the phase:

- **Challenge:** Classification of support requests in the service desk requires clarification **Improvement suggestion:** Clarify the options in 'Reason for Contact Request' field of the incident record. Make the difference between service requests and incidents visible. Service area and the type of of support requests should be different fields. Collect concrete examples of both incidents and service request for training purposes.
- **Challenge:** Customers are not able classify support requests correctly **Improvement suggestion:** Remove the classification option from sucustomers and simplify the submission of support requests.
- **Challenge:** It is difficult to identify repeating incidents from the service desk system. **Improvement suggestion:** Mark the repeating incidents (for example, create an additional 'check box' type data field to an incident record: Repeating incident = x). Use the 'Relate Cases' function to establish relationships between similar cases. Create a problem record based on a repeating incident.
- **Challenge:** The interface between incident management and problem management does not work. People do not understand the difference between incidents and problems. **Improvement suggestion:** Train employees to open a problem record. Establish a simple-to-understand guidelines for problem management including triggers for problem management.
- **Challenge:** Service desk workers record several cases under one incident. **Improvement suggestion:** Train service desk workers to record cases in such a way that one incident record includes only one issue.
- **Challenge:** Improvement ideas are not recorded systematically into the service desk system. **Improvement suggestion:** Improvement ideas should be sent to a Continual Service Improvement team or Change Management team.
- **Challenge:** Lack of Configuration Management Database (CMDB). **Improvement suggestion:** Establish a Configuration Management process that is responsible for updating, maintaining and managing a Configuration Management Database (CMDB).

IV. Analysis

Regarding the service desk we observed during the study that the service desk tool supports the implementation of IT service management well. The organization had invested in automatizing the handling of service requests and electronic forms were well exploited in service request management. Regarding incident management roles, we observed that roles and responsibilities and incident management activities were defined in the process description and they followed the IT service management terminology. The organization had assigned a well functioning team (2-3 persons) for configuring the IT service management tool. Detailed process descriptions had been created for ITSM processes. Regarding the results, it has to be mentioned that the organization had focused on service desk and incident management in the first phase of process improvement cycle (starting from Spring 2011). Therefore, the remaining ITSM processes, such as change management and problem management, were inmature.

Many of the challenges seemed to be related to classification of support requests. Service desk workers indicated that users and customers have problems in classifying requests but nobody had measured the number of incorrectly classified requests. Our research team reported to the case organization that the service area requires changes and that the using the service desk system should be as simple as possible for customers.

It was not a surprise that the organization had problems with problem management. This challenge has been noted also in our previous studies. Some service desk workers had classified the case as a "problem" when the problem was one of the options in Reason for Contact field. Crucial for problem management would be recording information on which incidents are repeating incidents and thus sources of problems.

Another challenging area was the continual service improvement that consists of three main areas: measurement, reporting and management of improvement ideas. Organisation needed a model how to handle service improvement ideas in a more systematic way. One possibility would be to assign development ideas to change management team that would open a Request for Change. Interfaces between processes seemed not to work wery well in practice, for example, between incident management and configuration management, change management, and event management.

V. Discussion and Conclusion

This paper aimed to answer the research problem: What types of challenges exist in the IT service provider's customer support? The main contribution of this study was to explore how a service desk tool and the support process was improved in Finnish Tax Administration and describe the identified challenges in incident management.

The key challenges we identified in service support were related to classification of support requests both from service desk workers' viewpoint and customers' viewpoint, understanding the differences between incident and problem management processes, identifying the sources of problems and interfaces between service management processes.

This case study included certain limitations. First, data were collected by using solely qualitative research methods.

 ISBN: 978-1-61208-184-7

Quantitative methods could have provided a richer view on the organization. However, the qualitative case study method suits well to research business process related challenges in organizational context. Second, the case organization was a partner of the software engineering unit's research project and thus not randomly selected. Third, this study included only one case organization's one service area. Further research could explore the IT service management challenges in other service areas of the case organization.

ACKNOWLEDGMENT

This paper is based on research in KISMET project funded by the National Technology Agency TEKES, European Regional Development Fund (ERDF), and industrial partners.

REFERENCES

[1] OGCb, *ITIL Service Delivery*. The Stationary Office, UK, 2002.

[2] OGC, *ITIL Service Strategy*. The Stationary Office, UK, 2007.

[3] OGCb, *ITIL Service Design*. The Stationary Office, UK, 2007.

[4] OGCc, *ITIL Service Transition*. The Stationary Office, UK, 2007.

[5] OGCd, *ITIL Service Operation*. The Stationary Office, UK, 2007.

[6] OGCe, *ITIL Continual Service Improvement*. The Stationary Office, UK, 2007.

[7] M. Sharifi, M. Ayat, A. A. Rahman, and S. Sahibudin, "Lessons learned in itil implementation failure," in *Information Technology, 2008. ITSim 2008. International Symposium*, vol. 1, Aug. 2008, pp. 1–4.

[8] M. Mohamed, V. Ribiere, K. O'Sullivan, and M. Mohamed, "The re-structuring of the information technology infrastructure library (itil) implementation using knowledge management framework," *The Journal of Information and Knowledge Management Systems*, vol. 38, no. 3, pp. 315–333, 2008.

[9] A. Lahtela, M. Jäntti, and J. Kaukola, "Implementing an itil-based it service management measurement system," in *Proceedings of the 4th International Conference on Digital Society*. St. Maarten, Netherlands Antilles: IEEE Computer Society, February 2010, pp. 249–254.

[10] J. Peppard, "Managing it as a portfolio of services," *European Management Journal*, vol. 21, no. 4, pp. 467–483, August 2003.

[11] I. Bardhan, H. Demirkan, P. Kannan, R. Kauffman, and R. Sougstad, "An interdisciplinary perspective on it services management and service science," *Journal of Management Information Systems*, vol. 26, no. 4, pp. 13–64, 2010.

[12] COBIT 4.1, *Control Objectives for Information and related Technology: COBIT 4.1*. IT Governance Institute, 2007.

[13] Microsoft, "Microsoft operations framework," http://technet.microsoft.com/en-us/library/cc506049.aspx, September 2010.

[14] V. Kapella, "A framework for incident and problem management," International Network Services whitepaper, 2003.

[15] F. Niessinka, V. Clerca, T. Tijdinka, and H. van Vliet, "The it service capability maturity model version 1.0," CIBIT Consultants&Vrije Universiteit, 2005.

[16] ISO/IEC 20000, *IT Service Management, Part 1: Specification for service management*. ISO/IEC JTC1/SC7 Secretariat, 2005.

[17] M. Kajko-Mattsson, "Corrective maintenance maturity model: Problem management," in *ICSM '02: Proceedings of the International Conference on Software Maintenance (ICSM'02)*. Washington, DC, USA: IEEE Computer Society, 2002, p. 486.

[18] W.-G. Tan, A. Cater-Steel, and M. Toleman, "Implementing it service management: A case study focussing on critical success factors," *Journal of Computer Information Systems*, vol. 50, no. 2, 2009.

[19] R. Yin, *Case Study Research: Design and Methods*. Beverly Hills, CA: Sage Publishing, 1994.

[20] K. Eisenhardt, "Building theories from case study research," *Academy of Management Review*, vol. 14, pp. 532–550, 1989.

 ISBN: 978-1-61208-184-7

Branching Program-Based Programmable Logic for Embedded Systems

Vaclav Dvorak
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
dvorak@fit.vutbr.cz

***Abstract*—The paper considers realization of logic functions by branching programs running on special purpose Decision Diagram Machines (DDMs). It is not the fastest way to implement logic, but it enables different versions and frequent modifications, e.g., in embedded systems. First, this paper derives upper bounds on the cost of multi-terminal binary decision diagrams (MTBDDs); the cost is directly related to the size of branching programs derived from MTBDDs. Second, optimization of heterogeneous branching programs is undertaken that makes a space-time trade-off between the amount of memory required for a branching program and its execution time. As a case study, optimal configurations of branching programs are found for a set of benchmark tasks. Beside DDMs, the technique can also be used for micro-controllers with a support for multi-way branching running logic-intensive embedded firmware.**

Keywords- Boolean functions; multi-terminal binary decision diagrams MTBDDs; branching programs; MTBDD complexity; decision diagram machines DDMs

I. INTRODUCTION

The popularity of programmable architectures is due to the savings in hardware development time and cost. Various methods exist to realize multiple-output logic functions by programmable architectures. The FPGAs are widely used; however, they require layout and routing in addition to logic design. Look-up table (LUT) cascades, i.e., a series connection of memories, are more flexible, since the architecture is simple; various classes of functions can be realized by LUT cascades efficiently [1], [11]. Finally, Decision Diagram Machines (DDMs) are special purpose processors that evaluate decision diagrams. Branching programs that evaluate single- or multiple-output Boolean functions on DDMs can be directly constructed from decision diagrams (DDs).

A binary DD (BDD) represents a single Boolean function in a form of the directed acyclic graph with internal decision nodes controlled by input variables and with terminal nodes valued 0 or 1. Generalization to integer-valued terminal nodes leads to multi-terminal BDDs (MTBDDs) [1]. As the number of decision nodes in the ordered (MT)BDDs depends dramatically on the order of variables, we strive to find such variable ordering that reduces the node count as much as possible, and proportionally also the size of the branching program. As the optimal ordering belongs among NP-complete problems [1], heuristic methods have been suggested and used toward this goal. For example, the sub-optimal ordering of variables and (MT)BDD synthesis can be done simultaneously by the iterative decomposition of the original function, i.e., by repeatedly removing variables that minimize the node count at the current level of the diagram [2].

Mapping of optimal (MT)BDDs to branching programs is straightforward; non-terminal nodes are mapped to branch instructions, whereas terminal nodes to output instructions. Branching programs run faster on a special purpose processor (DDM) than on a general-purpose CPU [3]. Optimization criteria for branching programs are the execution time, memory size (area) or the area – time product. Some parameters subject to optimization are: testing more than 1 variable at a time, a number of instruction addresses, and a number of parallel DDMs. With the help of above optimizations, the execution speed of branching programs can be even adjusted to achieve very high performance [4]. Among applications of DDMs, let us mention micro-program sequencers, logic simulators, industrial programmable logic controllers and recently packet filters [5].

In this paper, we first analyze the MTBDD cost for general *R*-valued functions of Boolean variables. Then the class of sparse functions often used in real life is defined. The new results on upper bounds of MTBDD cost and profile of sparse functions are derived. This is generalization of results for single-output functions in [6]. In the second part, we show optimization of branching programs with respect to the area – time product. Heterogeneous MTBDDs for arbiters and controlled shift circuits serve to illustrate this optimization.

The paper is structured as follows. Section II introduces related works, whereas Section III gives the preliminaries. MTBDD profiles and costs for sparse logic functions are derived in Section IV. Mapping MTBDDs to branching programs is dealt with in Section V and branching program optimization in Section VI. The experimental results and future research directions are commented on in Conclusion.

II. RELATED WORKS

Various DDMs have been proposed in literature for evaluation various types of decision diagrams - ordered BDDs and Quaternary Decision Diagrams (QDDs), quasi-reduced, BDDs and QDDs (QRBDDs and QRQDDs) and ordered Heterogeneous Multi-valued Decision Diagrams (HMDDs) as well as Quasi-Reduced HMDDs (QRHMDDs). Six DDM architectures have been compared with respect to area-time complexity, throughput and compatibility to the existing memory [3].

Area-time complexity is important for embedded systems, because DDM with low area-time complexity dissipates low

 ISBN: 978-1-61208-184-7

power. Since the instruction memory occupies the most area for the DDM, we assume that the area is proportional to the memory size. The QDD Machine was found the best for area-time complexity [3]. Quasi-Reduced diagrams contain not only true decision nodes, but also degenerated nodes with one output edge only. This leads to higher memory consumption but enables pipelining and thus leads to the best throughput for QRQDD Machines [3].

The main problem with the above DDM architectures is that multiple-output functions are implemented by partitioning into single output functions. That is why we study the direct use of MTBDDs for branching programs. In Section IV we formulate hypothesis 4.1 suggesting that for a multiple-output sparse logic function the cost and the memory area to store the MTBDD are much lower than those for *r* BDDs of its *r* single-output component logic functions. Thus the architecture of the MTBDD Machine proposed in this paper should be superior.

Six DDM architectures mentioned above all use fixed number (1 or 2) of control inputs at decision nodes. The MTBDD machine is more flexible - the number of tested variables can be varied from one node to another, e.g., between 1 and 4. This lowers memory requirements and power consumption even further.

Code optimization for QDD Machines [10] has been achieved by means of 3 instead of 4 addresses in the instruction and by means of four types of branching instructions. In the other hand, in the MTBDD Machine we use only two instructions and only one base address that gets modified by the values of tested variables.

Applications for DDMs include industrial process controllers and logic simulators. Also a parallel DDM with 128 QDD Machines implemented on FPGA and running at 100 MHz has been proposed [4], that is about 100 times faster at the peak performance than Intel´s Core2 Duo microprocessor (@ 1.2 GHz) and requires a quarter of the memory.

III. BASIC DEFINITIONS AND NOTIONS

To begin our discussion, we define the following terminology. A system of *m* Boolean functions of *n* Boolean variables,

$$f_n^{(i)}: (Z_2)^n \rightarrow Z_2, \; i = 1, 2, ..., m \qquad (1)$$

will be described as a logic function F_n with output values from $Z_R = \{0, 1, 2, \ldots, R\text{-}1\}$,

$$F_n: (Z_2)^n \rightarrow Z_R, \qquad (2)$$

where R is the number of distinct combinations of m output binary values enumerated by values from Z_R.

Function F_n is incomplete if it is defined only on set $X \subset (Z_2)^n$; $(Z_2)^n \setminus X$ is the don't care set. (We assume that all component functions (1) have the same don't care set.)

Definition 3.1 Under the **sparse functions** $F_n: (Z_2)^n \rightarrow Z_R$ we will understand functions with the domain $(Z_2)^n$ divided into two subsets X and D, $(Z_2)^n = X \cup D$, $|X| << 2^n$, if one of the following conditions hold:

1) F_n is a fully specified function in $(Z_2)^n$,
$F_n: [X \rightarrow Z_R \setminus \{0\}, D \rightarrow \{0\}]$
(without loss of generality, value 0 is taken as the dominant value);

2) F_n is an incomplete function in $(Z_2)^n$, $F_n: X \rightarrow Z_R$ and $(Z_2)^n \setminus X = DC$ is the don't care set.

In this second case we can artificially define mapping $DC \rightarrow \{0\}$ and come back to the first case. Further on we therefore consider only the first case.

Definition 3.2 The **weight** of function F_n, denoted by u, is the cardinality of set X in Def. 3.1, $u = |X|$.

Definition 3.3 Let $F_n: (Z_2)^n \rightarrow Z_R$ be the function of binary variables $x_1, x_2, \ldots, x_n$. **Sub-function** $f(x_{n-k+1}, \ldots, x_{n-1}, x_n)$ of k variables is the function $f = F_n(v_1, v_2, \ldots, v_{n-k}, x_{n-k+1}, \ldots, x_{n-1}, x_n)$ for any given combination of binary constants $v_1, v_2, \ldots, v_{n-k}$.

Lemma 3.1 There are up to min $(2^{n-k}, R^{2^k})$ sub-functions of k variables, $k = 1, 2, \ldots, n$, but not all of them are necessarily distinct.

(Proof) According to Def. 3.3, each k-variable sub-function $(Z_2)^k \rightarrow Z_R$ is related to a particular binary vector $(v_1, v_2, \ldots, v_{n-k})$. There are 2^{n-k} such vectors and related sub-functions. On the other hand, the number of k-variable sub-functions is limited by the number of function values R. Maximum number of single variable (k=1) sub-functions is the same as the number of distinct pairs of function values, i.e., R^2. Two-variable sub-functions (k=2) are 4-tuples of function values and there are up to R^{2^2} of them. Continuing in the same way, we have up to R^{2^k} sub-functions of k variables (2^k-tuples of function values). A lower value of the two limits gives the bound, QED.

Definition 3.4 Let the order of variables in the MTBDD be $x_1, x_2, \ldots, x_n$ and the set of nodes controlled by x_j be the **level** j of the diagram. The **local width** w_j of the MTBDD at level j, $j = 1,2,\ldots, n$, is the number of all nodes at level j, i.e., the number of all distinct sub-functions of n-j+1 variables $x_{n-(n-j)}, \ldots, x_{n-j}, x_n$. The **width** w of the MTBDD is the maximum width of the MTBDD among the levels (w is referred to as the C-measure in [1]).

Note that k sub-function variables are counted from x_n backwards, whereas local widths $w_1, w_2, \ldots, w_j$ are indexed from x_1 onwards, i.e., the same way as are MTBDD levels. The relation between indices j and k is thus $j = n$-k+1.

Definition 3.5 Let $F_n: (Z_2)^n \rightarrow Z_R$ be the function of binary variables $x_1, x_2, \ldots, x_n$. The **profile** of the function F_n is the vector $(w_1, w_2, \ldots, w_n)$. Note that always $w_1 = 1$, $w_2 = 2$. The total sum of all non-terminal nodes is $W = w_1 + w_2 + \ldots + w_n$.

Definition 3.6 The **local cost** c_j of the MTBDD at level j is the number of true decision nodes (distinct non-constant sub-functions) in that level. The **cost** C of the MTBDD is the sum $C = c_1 + c_2 + \ldots + c_n$.

The local cost c_j is always less or equal the local width w_j because c_j includes only decision nodes with two output edges whereas w_j is the number of all nodes at level j including those with a single output edge (depicted by black dots in the sample MTBDD at Fig 3.1).

Example 3.1 A sample MTBDD is in Fig.1. The profile of the related function is {2, 3, 3, 4}. The number of true decision nodes is at the minimum (C =4); if the function is to depend on all its variables, at least one true node per variable is required. Note that decision nodes with a single output edge

 ISBN: 978-1-61208-184-7

do not decide anything. We can shift terminal nodes up to the root over a sequence of such nodes and branch to terminal nodes not only in the last level. For example, the terminal node 2 is reached after testing variable x_1 and x_2 only.

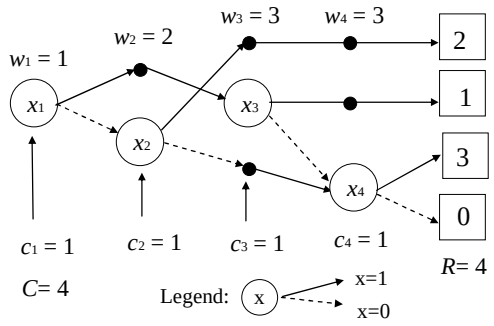


Figure 1 An example MTBDD for the 4-valued function of 4 Boolean variables.

Each of two characteristics, the profile and cost, is important in one of two different implementations of F_n. Whereas the profile, and especially the global width w, determine the LUT cascade configuration for F_n (hardware implementation, [11]), the size of the branching program is proportional to cost C; remaining W–C nodes just shrink to edges.

Most often two parameters of MTBDDs are optimized: cost C and width w. For branching programs based on MTBDDs, the cost optimization is of interest. Minimization of two parameters, cost C and width w, cannot be strictly separated. In the bottom-up synthesis of MTBDDs using heuristics [7], we select the variable so as to minimize the number of nodes in the next higher level of the MTBDD. If two variables produce the same number of nodes, we take the one with a lower number of true decision nodes. This goes on iteratively level by level, from leaves to the root. We expect that the total cost will be close to the minimum total cost. This has been confirmed for functions of up to 10 variables by an exhaustive search [11].

IV. Complexity Issues

Before analyzing complexity of sparse functions, we first review the complexity of general multiple-output Boolean functions. The knowledge of a function profile is essential for complexity analysis. By summing up local costs and by inspecting local widths of a MTBDD, we can arrive at global cost and width given in the following

Theorem 4.1 Cost C and width w of the MTBDD for function F_n: $(Z_2)^n \rightarrow Z_R$ are upper-bounded by

$$w \le \max_k \min(2^{n-k}, R^{2^k})$$
$$C \le \min_k (2^{n-k} + R^{2^k}) - R - 1, \quad (3)$$

where $k = 0, 1, \ldots, n-1$.

(Proof). The first relation follows directly from Lemma 3.1 and Def. 3.4, if we include sub-functions of k=0 variables (terminal values). In the case of cost C we must subtract constant sub-functions (nodes with a single output edge), see Lemma 3.1:

$$C = 1 + 2 + 4 + \ldots + 2^{n-(k+1)} + \\ + (R^{2^k} - R^{2^{k-1}}) + \ldots + (R^{2^2} - R^2) + (R^2 - R), \quad (4)$$

By computing the sum and taking the minimum we arrive at the total cost C:

$$C \le \min_k (\sum_{i=k+1}^{n} 2^{n-i} + \sum_{i=1}^{k} (R^{2^i} - R^{2^{i-1}}) = \min_k \left[(2^{n-k} - 1) + (R^{2^k} - R) \right],$$

QED.

Example 4.1 The profile of a general 4-valued function of 12 variables is according to Lemma 3.1 limited by
2, 4, 8, 16, 32, 64, 128, 256, 512, 256, 16, 4.
The MTBDD cost is

$C \le (1+2+4+\ldots +512) + (256-16)+(16-4) = 1275$ and width $w \le 512$. These bounds are too weak for real-life functions which have typically low values of w.

Random functions $(Z_2)^n \rightarrow Z_R$, $R = 2^r = 2^n$ are the most difficult functions to implement. Their MTBDDs have a form of the full binary tree and the number of all sub-functions is $W = C = 1+2+4+\ldots 2^{n-1} = 2^n-1$.

TABLE 1 Upper Bounds on MTBDD Cost

R ↓	n									
	1	2	3	4	5	6	7	8	9	10
2	1	3	5	7*	15*	29	45	77	141	269
4		3	7	15	27	43	75	139	267	507
8			7	15	31	63	119	183	311	567
16				15	31	63	127	255	495	751
32					31	63	127	255	511	1023
64						63	127	255	511	1023

Binary n-bit multipliers with $2n$ binary inputs and $2n$ outputs have R close to 2^{2n}. (End of Example)

The upper bounds for cost C for selected classes of multiple-output logic functions are summarized in Tab.1. They were calculated from (3), except for two items marked by asterisk which should have been 9 and 17, see Theorem 4.2 and Corollary 4.1 below. Separate regions in Tab.1 are interpreted as follows:
- the top region: minimum in (3) occurs at i = 2,
- the middle region: minimum in (3) occurs at i = 1,
- the bottom region: minimum in (3) occurs at i = 0.

Theorem 4.2 The cost of the BDD of the arbitrary logic function of 4 variables is $C \le 7$.

(Proof by construction) There are 65536 functions of 4 variables. Under the group of negations and permutations, we can reduce this count to only 222 equivalence classes. Now it is sufficient to prove the theorem for one representative out of each equivalence class. This was done by exhaustive search considering all 24 variable orderings. The upper bound 7 was reached in only 8 cases (the average node count was 4.6), QED.

 ISBN: 978-1-61208-184-7

Corollary 4.1 The cost of the BDD of the arbitrary logic function of 5 variables is $C \le 15$. (The first decision node can fork to two BDDs of two 4-variable sub-functions.)

The upper bounds on cost are too weak for most of functions in digital engineering practice. Very often the functions are defined only in a small fraction of all 2^n binary input vectors. Therefore, from now on, we will consider sparse logic functions and will attempt to obtain stronger upper bounds for them. Such bounds on local values of $w_{n\text{-}k}$, k = 2, 3,…,6 are known [6] for single output Boolean functions (R=2). Here we will analyze multiple-output Boolean functions and generalize the previous results.

Lemma 4.1 Let sparse function F_n: $(Z_2)^n \rightarrow Z_R$ attains non-zero values 1, 2, …, R-1 in $|X| = u << 2^n$ points, $X \subset (Z_2)^n$. Consider distinct k-variable sub-functions of F_n. Such sub-functions are specified by column vectors of $t = 2^k$ elements (rows). The maximum number of distinct column vectors is (5)

$$w_{n-k+1} = \lambda(t,u,R) = \sum_{i=0}^{\sigma} \binom{t}{i} (R-1)^i + q \text{ for } 0 < u < \sum_{i=1}^{t} i \binom{t}{i} (R-1)^i = u_m$$

where σ is the integer satisfying the relation

$$u_1 = \sum_{i=1}^{\sigma} i \binom{t}{i} (R-1)^i \le u \le \sum_{i=1}^{\sigma+1} i \binom{t}{i} (R-1)^i = u_2 \qquad (6)$$

and $q = \lfloor (u-u_1)/(\sigma+1) \rfloor$.

Note that $\lambda(u)$ is piece-wise linear, monotone increasing for $0 < u < u_m$. In the first interval $1 \le u \le t(R\text{-}1)$ the value of $\lambda(u) = u+1$. On the other hand, when $u \ge u_m$, $\lambda(u)$ takes up the constant value

$$\sum_{i=0}^{t} \binom{t}{i} (R-1)^i. \qquad (7)$$

Theorem 4.3 Let $\lambda(2^k, u, R)$ be the number of distinct k-variable sub-functions for an n-variable sparse function $(Z_2)^n \rightarrow Z_R$ and with weight u. Then

$c_{n\text{-}k+1} \le \lambda(2^k, u, R) - \varepsilon, \quad k = 1,..., n\text{-}1, \quad$ and $c_1 = 1$.

This theorem is immediately derived from Lemma 4.1: local cost $c_{n\text{-}k+1} \le w_{n\text{-}k+1}$, because of constant sub-functions. The upper limit on $c_{n\text{-}k+1}$ is the upper limit on $w_{n\text{-}k+1}$, i.e., $\lambda(2^k, u, R)$, decreased by ε, that is by the number of constant sub-functions of k variables for the given u and R. In general, if $u \ge u_m$, λ attains the saturated value $\lambda = R^{2^k}$ and we must subtract $\varepsilon = R^{2^{k-1}}$ constant sub-functions from λ to get the value of $c_{n\text{-}k}$, as seen from (4). However, if $u < u_m$ and $\lambda < R^{2^k}$, correction ε depends on u, = ε(u); we must always subtract 1 (all zeros pattern) and incidentally some other constant patterns if they appear in the range of u, QED.

Example 4.2 Let us have t = 2, R = 4, u = 10. All distinct sub-functions of a single variable are columns in the Table 2. Now, we can compare upper bounds on local costs c_n and values of $\lambda(2^1, u, 4)$ for some values of u.

For u = 10, we have λ= 9, but c_n = 8 only (all zero pattern does not count). For upper bound we have to consider the worst case when constant sub-functions are taken only if there is no other choice (see the last 3 columns in Table 2). So if u = 18 we get λ= 13 and c_n =12. This count does not grow any further, for u > 18 we get always c_n =12. (End of Example)

Local costs are functions of three parameters $t = 2^k$, u and R. With some computations according to Lemma 4.1, one can figure out the upper bound on the cost of any given sparse function.

Example 4.3 Cost profiles of sparse function of 13 variables, $(Z_2)^{13} \rightarrow Z_R$, R = 2, 4, 8, 16 and u = 100 are depicted in Fig. 2. For illustration, let us calculate $\lambda(t, u, R)$ for u = 100, R = 8 and t = 2, 4, 8, 16, …:

$t = 2: u_1 = 14 \le u \le 14 + 98 = 112 = u_2$,
$\lambda(2,100,8) = 1 + 14 + \lfloor (100-14)/2 \rfloor = 57,\ c_{13} = 56$

$t = 4: u_1 = 28 \le u \le 28 + 294 = 322 = u_2$,
$\lambda(4,100,8) = 1 + 28 + \lfloor (100-28)/2 \rfloor = 65,\ c_{12} = 63$

$t = 8: u_1 = 56 \le u \le 56 + 2744 = 2800 = u_2$,
$\lambda(8,100,8) = 1 + 56 + \lfloor (100-56)/2 \rfloor = 79,\ c_{11} = 78$

$t = 16,\ 32,\ 64: c_{10} = c_9 = c_8 = u = 100.$

The remaining local costs from c_7 to c_1 are 64, 32, 16, 8, 4, 2, 1. (End of Example)

TABLE 2 COMPUTATION OF $\lambda(t, u, R) = \lambda(2, 10, 4) = 9$

0	1	2	3	0	0	0	1	1	2	2	3	3	1	2	3
0	0	0	0	1	2	3	2	3	1	3	1	2	1	2	3
1			6								9				
←			u=10										→		

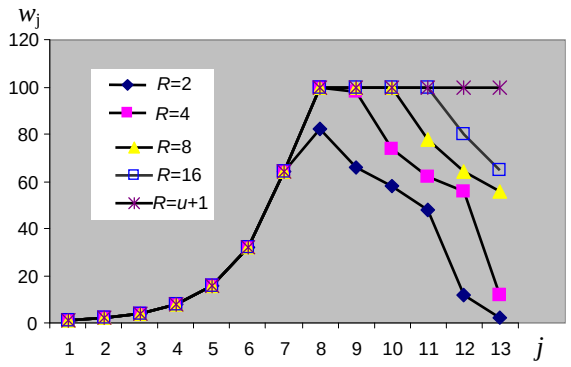


Figure 2 Profiles of sparse functions of 13 variables (u = 100)

From the inspection of Fig. 2, the following hypothesis can be formulated:

Hypothesis 4.1 The cost of the MTBDD for multiple-output sparse function F_n: $(Z_2)^n \rightarrow Z_R$, $R = 2^r$ and the memory area to store the MTBDD is much lower than the cost and memory area to store r BDDs for its r single-output component logic functions.

V. MAPPING MTBDDS TO BRANCHING PROGRAMS

Branching programs have typically two instructions: (multi-way) branch and output instruction. Their format depends on the architecture of a DDM as well as on the type of the DD. For a multi-output function, a partition into single output

functions (BDDs) or into groups of functions (multiple MTBDDs) has been used [3]. Due to our hypothesis we will use partition into groups of functions. Our starting point will be the MTBDD with sub-optimal ordering of variables obtained by heuristic [7], that can be easily converted to a 2^k-valued DD with generally non-uniform k (heterogeneous DDs). The architecture of a suitable DDM is in Fig. 3.

The code memory stores instructions that evaluate nodes of the DD. Each node is represented by a 2^k-way dispatch table that starts at a node base address and its items are indexed by k-bit offset. Each item contains a code for input multiplexers (group id) and the mask which together specify the offset for the next node. Fig. 4 shows two instruction formats. The multi-way branch instruction evaluates a non terminal 2^k-ary node, while the output instruction evaluates a terminal node:

1) the jump to an address in the PC modified by BCU;
   L_n: branch $L_m@x_1...x_k$;
2) output and the unconditional jump to the specified address
   L_n: output, go to L_m .

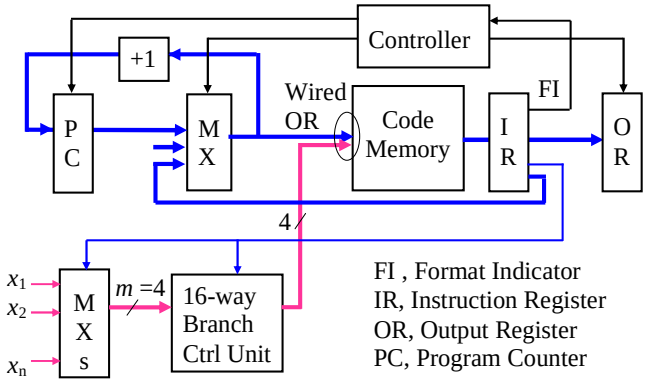


Fig. 3 Architecture of Decision Diagram Machine (DDM)

branch instruction

FI	group id	mask	node base address

output instruction

FI	output data	next address

Fig. 4 Instruction formats for a DDM with the support for multi-way branching

The multi-way indirect branch is executed in 1 clock cycle, the current base address in the PC gets modified by external variables (operator @), by up to 4 variables at a time, including 0 variable (no modification, the unconditional jump), by means of 16-way Branch Control Unit (BCU). Input variables are selected by multiplexers, so that instructions contain MXs control field and a BCU mask. The task of the 16-way BCU4 is to shift up to 4 active inputs, selected by a 4-bit BCU mask, to the lowest positions of the 4-bit output vector and reset the rest of outputs. The output vector then serves as an offset from the base address of a dispatch table. This way the dispatch tables can be stored in code memory in a compact form; the bits of the base address supplied by the BCU must be reset to 0 if wired-OR is used for modification. Tri-state outputs of the BCU are wire-ORed to the address inputs of the code memory.

The advantages of above architecture are:

- the word lengths for the multi-way branches are the same
- relatively short word lengths for instructions (single base address only)
- easy extension for other instructions and addressing (incrementing PC for longer output sequences, support for a return address stack, etc.)

Example 5.1 Let us implement the Round Robin Arbiter (RRA) with 4 input requests r_0, r_1, r_2, r_3. The priority register $[p_0, p_1, p_2, p_3]$ points to the requester i, currently with the highest priority (one-hot encoding). Priority decreases for subsequent inputs:

$$g_3 = p_3r_3 + p_0!r_0r_3 + p_1!r_1!r_0r_3 + p_2!r_2!r_1!r_0r_3$$
$$g_2 = p_2r_2 + p_3!r_3r_2 + p_0!r_0!r_3r_2 + p_1!r_1!r_0!r_3r_2$$
$$g_1 = p_1r_1 + p_2!r_2r_1 + p_3!r_3!r_2r_1 + p_0!r_0!r_3!r_2r_1$$
$$g_0 = p_0r_0 + p_1!r_1r_0 + p_2!r_2!r_1r_0 + p_3!r_3!r_2!r_1r_0.$$

The quaternary MTBDD for this RRA obtained by means of HIDET tool [2] is at Fig. 5. The sample of a branching program with inspection of two binary inputs at a time is shown at Fig. 6. The symbolic program is composed of 9 4-way and of 2 2-way dispatch tables. The base addresses of dispatch tables shown in Fig. 6 as L1 to L11 correspond to the same labels in the MTBDD in Fig. 5. The total number of instructions is

$9\times4 + 2\times2 = 40.$ (End of example)

VI. BRANCHING PROGRAM OPTIMIZATION

The most important parameters that are usually subject of optimization are memory size, execution time, and power consumption. Since code memory occupies the most area for the whole DDM, we assume that the area is proportional to the memory size. Area-time complexity, or the product of memory size and performance, is important for embedded systems. In this section, we will focus on optimizing the area-time product only. As a by-product, a processing with low area (time) means low dissipation of static (dynamic) power, too.

There are two possibilities for optimization, ordering of input variables (not discussed in this paper) and their grouping. Whereas variable ordering influences the size of a MTBDD and required code memory, grouping of input variables impacts the speed of processing. Testing several input variables simultaneously can also be visualized as converting the MTBDD into a multi-valued DD (MVDD), [8] – [9]. Very often the nodes of such MVDD are degenerated in a certain degree, i.e., not all the output edges are distinct. The DDM architecture at Fig. 3 can utilize this fact for minimization of memory requirements; it can vary the number of variables tested in each step according to the local structure of the MTBDD on the followed path. For example if we test

 ISBN: 978-1-61208-184-7

three variables at a time, the complete tree of 7 nodes could be converted to a single 8-valued node. However, there are

$$\sum_{i=0}^{7} C(7,i) = 2^7$$

possible configurations of i = 0, 1, …. 7 degenerate nodes and if they occur in the same level of the diagram, we can skip their testing and reduce the size of a dispatch table. Fig. 7 shows all possible sub-graphs rooted at a local node subject to such reduction. An extreme case is that there are no true decision nodes on the path and a dispatch table is eliminated completely. If we test k variables at a time, then there are

$$\sum_{i=0}^{k-1} C(k,i) = 2^k - 1$$

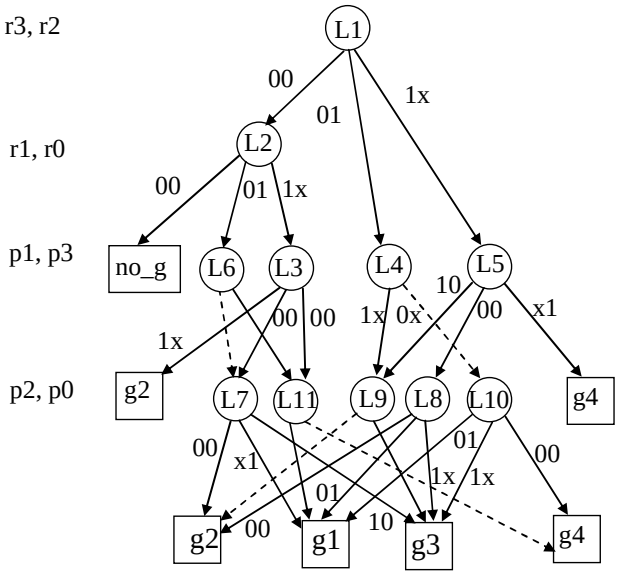


Fig. 5 MTBDD of the 4-input RR arbiter.

```
RRA:        exit L1@r3r2
L1@00:      exit L2@r1r0
L1@01:      exit L4@p1p3
L1@10:      exit L5@p1p3
L1@11:      exit L5@p1p3
L2@00:      no_g exit Next
L2@01:      exit L6@p3
L2@10:      exit L3@p1p3
L2@11:      exit L3@p1p3
        …..
L10@00:     g4 exit Next
L10@01:     g1 exit Next
L10@10:     g3 exit Next
L10@11:     g3 exit Next
L11@0:      g4 exit Next
L11@1:      g1 exit Next
Next:
```

Fig. 6 A symbolic microprogram for the RRA.

sub-graphs leading to dispatch tables of reduced size. Simplifying nodes with 2^k outputs wherever possible leads to a heterogeneous MVDD with nodes controlled by k or less variables.

Example 6.1 Continuing in Example 4.1, we can apply various grouping of input variables and then reduction of multi-valued nodes to the smaller ones. If we create groups of 2 input variables, we will do with 7 4-way nodes (7 dispatch tables of size 4) and 4 binary nodes (4 dispatch tables of size 2), altogether 36 instructions. Had we used only single variable tests (a binary program with 2-way branching), we would need 17 dispatch tables of size 2, i.e., 34 instructions in total. However, the performance would be 2- times lower due to execution of a chain of 8 instructions, one in each level of the MTBDD. Processing in three steps could test 2, 3, 3 or 2, 2, 4 decision variables, but the area-time product would get worse. The fastest execution tests 4 decision variables at a time (16-way branching). The features of various options are summarized in Table 3. The area × time product is a figure of merit of quality of the implementation. It gets its best (lowest) value for testing two and four variables at a time. (End of Example)

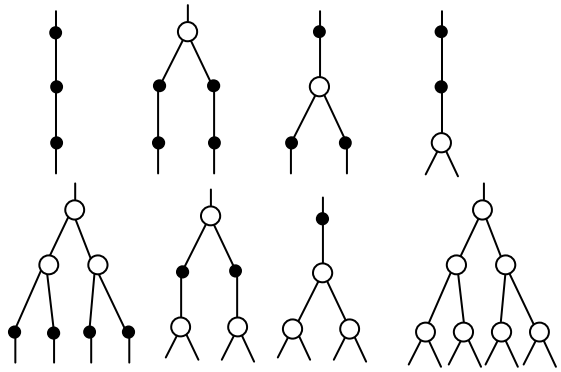


Fig. 7 MTBDD sub-graphs with 0, 1, 2 and 3 decisions on each path

TABLE 3 VARIOUS MICROPROGRAM OPTIONS

tested variables:	total micro-instructions	execution time	space x time
8 x 1	34	8	272
4 x 2	36	4	144
2, 3, 3	52	3	156
2, 2, 4	64	3	192
2 x 4	72	2	144
8	256	1	256

Similar optimization has been carried out by a home-made software tool for a number of logic modules such as branch control units (bcu), round robin arbiters (rra), least-recently-served arbiters (lrs) and priority encoders (pe) with a various number of inputs.

MTBDDs of these logic modules have been obtained and optimized by HIDET tool from cube specification [2].

 ISBN: 978-1-61208-184-7

MTBDD parameters (cost *c*, size *s*, and width *w*) for these modules are listed in Table 4, together with optimal grouping of input variables ordered by HIDET and resulting area × time (a×t) product. This product is calculated as the aggregate number of all instructions in dispatch tables (a dispatch table for a *k*-ary node has 2^k instructions) multiplied by the number of dispatch tables from the root to leaves.

The results show that the best area-time complexity is obtained for $k = 3$ or 4 variables tested simultaneously. This does not corresponds to the result in [6], where quaternary ($k = 2$) DDs were found best with respect to this figure of merit in a different set of benchmarks. The reason for this may be not only in benchmarks, but also in our different DDM architecture supporting variable multi-way branching.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a new MTBDD machine that could outperform known DDM architectures. The size of the branching programs and the maximum required code memory for this machine can be estimated using derived upper bounds on the cost of MTBDD for logic functions specified in *u* input vectors and attaining only a single value (0) for other input vectors. The bounds are not limited to sparse functions with $u << 2^n$, but are valid generally. The case of incomplete functions can be treated similarly, replacing don´t - cares by a dominant function value (0).

TABLE 4 OPTIMUM BRANCHING PROGRAM CONFIGURATIONS

	n	m	s	c	w	groups	a×t
bcu4	8	4	45	30	16	2x4	160
bcu6	12	6	189	126	64	3x4	1008
bcu7	14	7	381	254	128	4,4,4,2	2368
bcu8	16	8	765	510	256	4x4	5440
rra4	8	4	37	17	8	2x4	144
rra6	12	6	91	40	11	3x4	438
rra8	16	8	179	75	22	4x4	1248
rra12	24	12	489	189	44	8x3	4688
lrs4	10	4	39	17	6	3,3,3,1	168
lrs6	21	6	119	36	9	7x3	742
pe8	8	4	8	8	2	4x2, 2x4	64
pe12	12	5	12	12	2	4x3	128
pe16	16	5	16	16	2	8x2, 4x4	256

Firmware implementation of a MTBDD is usually a matter of trade-off between performance and the size of memory storing the code. The memory size can be derived as an aggregate size of all dispatch tables, and the performance is given by the number of dispatch tables on the path from the root to leaves of the MTBDD. The area-time complexity has been optimized for the MTBDD machine that supports multi-way branching in one clock cycle with variable number of ways (0 to 16). On the given set of benchmarks the optimum area-time product has been reached for DDM for *k*-ary nodes with $k = 3$ and 4, what is in contradiction to finding QDD ($k = 2$) as optimal in [3].

Future research should compare the MTBDD Machine to other published DDMs on the common set of benchmarks and thus verify hypothesis 4.1 and implied superiority of MTBDD Machines. The library of optimal MTBDDs for a such a benchmark suite should be created first, it is not available as yet. Another optimization problem is to pack dispatch tables of all MTBDD nodes into as small memory as possible. The final step would be a hardware implementation of the MTBDD machine and also of its parallel version in FPGA.

ACKNOWLEDGMENT

This research has been carried out under the financial support of the research grants GP103/10/1517 “Natural Computing on Unconventional Platforms”, and MSM 21630528 “Security-Oriented Research in Information Technology” and the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

REFERENCES

[1] T. Sasao, Memory-Based Logic Synthesis. Springer, New York, 2011, 189 pages.

[2] V. Dvořák and P. Mikušek, "Design of Arbiters and Allocators Based on Multi-Terminal BDDs". In: Journal of Universal Computer Science, Vol. 16, No. 14, 2010, AT, pp. 1826-1852.

[3] H. Nakahara, T. Sasao, and M. Matsuura, "A comparison of architectures for various decision diagram machines," International Symposium on Multiple-Valued Logic, Barcelona, Spain, May 26-28, 2010, pp. 229-234.

[4] H. Nakahara, T. Sasao, M. Matsuura, and Y. Kawamura, "A parallel branching program machine for sequential circuits: Implementation and evaluation," IEICE Transactions on Information and Systems, Vol. E93-D, No. 8, pp. 2048-2058, Aug. 2010.

[5] H. Nakahara, T. Sasao, and M. Matsuura, "Packet classifier using a parallel branching program machine," 13th EUROMICRO Conference on Digital System Design (DSD-2010) Lille, France, Sept. 1-3, 2010, pp. 745-752.

[6] T. Sasao, "On the number of LUTS to realize sparse logic functions," 18th International Workshop on Logic and Synthesis, (IWLS-2009), Berkeley, CA, U.S.A., July 31-Aug. 2, 2009, pp. 64-71.

[7] P. Mikušek and V. Dvořák, "On Lookup Table Cascade-Based Realizations of Arbiters". Proc. of the 11th EUROMICRO Conference on Digital System Design DSD 2008, Parma, IT, IEEE CS, pp. 795-802.

[8] S. Nagayama and T. Sasao, "On the optimization of heterogeneous MDDs," IEEE Transactions on CAD, Vol. 24, No.11, Nov. 2005, pp.1645-1659.

[9] H. Nakahara, T. Sasao, and M. Matsuura, "A Comparison of heterogeneous multi-valued decision diagram machines for multiple-output logic functions," International Symposium on Multiple-Valued Logic (ISMVL-2011), Tuusula, Finland, May 23-25, 2011.

[10] T. Sasao, H. Nakahara, K. Matsuura, Y. Kawamura, and J.T. Butler, "A quaternary decision diagram machine: Optimization of its code," IEICE Transactions on Information and Systems, Vol. E93-D, No. 8, pp. 2026-2035, Aug. 2010.

[11] V. Dvořák and P. Mikušek, "On the cascade realization of sparse logic functions". In: Euromicro Proceedings, Oulu, FI, IEEE CS, 2011, pp. 21-28.

 ISBN: 978-1-61208-184-7

Orchestration Driven by Formal Specification

Charif Mahmoudi
Logics, Algorithm, Complexity Laboratory
LACL, Paris 12 University
Creteil, France
charif.mahmoudi@sfr.fr

Fabrice Mourlin
Logics, Algorithm, Complexity Laboratory
LACL, Paris 12 University
Creteil, France
fabrice.mourlin@wanadoo.fr

***Abstract*—Mobile agent software provides a programming paradigm which allows reconfiguration during runtime. Because code migration is a basic concept, software architecture becomes more important. Classically, the lifecycle of distributed application starts with specification description. Several facets have to be specified: agent behavior, message exchange, service composition, but also architecture. This description has also two levels: software and hardware. We use formal specifications because our objective is to define properties about our application. Also, process algebra, like Pi-Calculus, is a formal language, which allows us to provide a formal description of architecture. We can then combine agent behavior and reason to define minimal constraint set of future runtime context. Our work provides a process from formal specification of distributed applications to a skeleton of BPEL script.**

Keywords-mobile agent; architecture specification; service composition.

I. INTRODUCTION

Mobile agent application is a kind of distributed application where software can react with its environment and react to external events. Also, this is particularly useful in case of unstable runtime context or when architecture changes during execution. For instance, grid computing needs large set of computing resources. But, if a resource is missing or fails, the whole computation has to continue until its end. In that context, the result is more important than the performance, also mobile agents are able to move computation to a node where computing resources is free [1]. Without mobile agent, it is not possible to adapt a distributed application to its runtime context because placement is defined at load time. This limit is suppressed with mobility.

Mobile agents are useful in other domain such as software administration or code instrumentation. Software administrator needs to deploy new distributed applications with adapted configuration for security, underlying services, etc. A first solution is to replicate a static image from one node of the network to the others, but the strategy becomes complex when the nodes are not similar. If location involves a specific behavior then mobile agent is a solution. It can adapt its mission to the precise location where it incomes. This can mean select specific permissions depending to a resource location, or choose between several persistence services, etc. [2].

Code instrumentation is another domain where context adaptation is essential. Software instruments can observe runtime properties such as time measure of methods, memory allocation of data structure or state of threads into a thread group. If the analysis is done after an execution, a classical approach can be applied, but if actions have to be done depending on features which are observed then only mobile agents can react and adapt their actions to a specific context [3] [4]. For instance, several threads are blocked because there is a gridlock. Also, a mobile agent can change state of one of the threads to force a specific execution.

We have presented the role of one agent into a distributed application but these examples are useful for understanding the concept of software adaptation based on code migration. Into a case study, there are a large number of mobile agents and all have a common objective, for instance data collection for a performance analysis. Coordination between agents is crucial to insure that all contributions will be used in a suitable manner. This means writing coordination specification. It plays the role of master description where each agent is a piece of software like a rugby player into his team. The whole objective is to win a match, but depending on his role into the group, his own behavior will be to adapt his actions to the context and his partners.

Our experience into software specification was about use of formal language like CCS [5] or Unity [6]. Agent migration needs a higher order language and Pi-Calculus possesses such kind of construction. Also, we used this formal language for writing our formal specifications. Pi-Calculus [7] has operational semantics, which allows us to evaluate terms and transform our specifications into other representations useful for reasoning. In this document, we present how we write coordination description of mobile agent group or agency. Then, we explain how this specification can be used to provide a more executable representation. Finally, we propose an approach to specify architecture and a way to exploit it by an agency. By the end, we sum up through an example that illustrates main concepts of agent migration with message definition.

II. COORDINATION SPECIFICATION

Coordination can be considered as a road book for an agency or group of mobile agents. It contains start state and a final state and between them a succession of steps. A step is realized by a mobile agent. Also, this action step is defined with a location where it has to be done, initial information

 ISBN: 978-1-61208-184-7

for the launch and eventually final information for observing results. This means that only one description is not enough but several descriptions are built concurrently.

Two specification approaches are observed. First, a top down specification approach needs to build all descriptions into a coherent manner. This starts with step definition; for instance, the objective to achieve and then the migration (from where to where); finally, the data format of the input message is defined. These descriptions can be completed by some more details about local resources and output message. This formal language has a syntax which allows designer to express mobility as term. Channels are used to exchange not only data but also agents which are specified as first order term. First, we present quickly higher order Pi-Calculus language.

A. Formal specification language

The descriptive ability that Pi-Calculus offers, emerges from the concept of naming, where communication links, known as channels, are referenced using a naming convention. Hence, mobility arises by having processes communicating the channel names. Some extensions are added by R. Milner himself to allow specification writers with higher order term [8]. Then, agent migration can be expressed through a communication of a first order term [9].

The Pi-Calculus notation (Fig. 1) models distributed agent into a system, which can perform input or output actions through channels, thus allowing the agents to communicate. The message which is sent from one agent to the other is a name, which gives a reference to a channel or a first order term which gives a reference to a local mobile agent.

P, Q	$::=$	0	nil
		$P \mid\mid Q$	parallel composition of P and Q
		$\overline{c}\langle v \rangle.P$	output v on channel c and resume as P
		$c(x).P$	input from channel c
		$(\boldsymbol{\nu} x)P$	new channel name creation
		$!P$	replication

Figure 1. Syntax of Pi-Calculus language.

When a term is received by an agent host, term unification algorithm is applied to propagate names into agent host definition. Operational semantics [9] is useful to build evaluation tree of the agent host term. As an example, we provide a specification of SLP protocol. The Service Location Protocol (SLP) is an Internet Engineering Task Force (IETF) standard track protocol [10] that provides a framework to allow networking applications to discover the existence, location, and configuration of networked services in enterprise networks.

SLP can eliminate the need for the user to know the technical features of network hosts. With the SLP, the user needs only to know the description of the service he is interested in. Based on this description, SLP is then able to return the URL of the desired service. SLP is a language independent protocol. Thus, the protocol specification can be implemented in any language. The SLP infrastructure consists of three types of agents:

1. *UserAgent (UA)* is a software entity that is looking for the location of one or more services; its role is client,
2. *ServiceAgent (SA)* is a software entity that provides the location of one or more services; its role is mobile agent,
3. *DirectoryAgent (DA)* is a software entity that acts as a centralized repository for service location information; its role is registry.

$$\mathbf{System} = \nu(SrvRqst, SrvRply, SrvReg, SrvUnReg, SrvAck)$$

$$UA(SrvRqst, SrvRply)$$

$$|SA(SrvReg, SrvUnReg, SrvAck)$$

$$|DA(SrvReg, SrvUnReg, SrvRqst, SrvAck)$$

Figure 2. Main term of SLP specification.

The subterms UA, SA and *DA* (Fig. 2) are detailed into annex. This grammar is useful for writing specification by hand but is quite complex to use into a workflow system also, we have translated this grammar into an XML schema. Also, this allows us to write specification in a more rigorous manner. Our XML schema stresses the structure of an agent based on the composition operators: sum, parallel, match, restriction, etc. A higher order Pi-Calculus specification becomes a well formatted XML description, which can be transformed into an object easily. It is the pilot of an activity of mobile agents.

B. Coordination of an agency

In the previous example, all components are independent and each has its own behavior. But, the problem is to describe relation between these behaviors. Coordination of software component is not a new challenge. Solutions have been already given by web project architects. Reo project forms a paradigm for composition of software components based on the notion of mobile channels [15]. This project defined its own coordination language which is a channel-based exogenous coordination model. The specification writer defines complex coordinators, called connectors, which are built out of simpler ones [16]. Of course, the Reo coordination language provides, pleasant features such that: loose coupling among components and services or support for distribution and mobility of heterogeneous components or compositional construction. But this language is not become a standard. Also, it is not easy to inter operate with other coordination model. But, Reo language stresses which are the key concepts into coordination. First, a composition of agents has two kinds of observation [14]. On one side, an external observer is not able to distinguish the structure of the composition. On the other side, an internal observer can follow the precise evaluation of the composition. Secondly, the better coupling is asynchronous and exchanges are considered as message passing [13].

We considered these requirements to select a language for defining coordination of agents. An obvious solution could be to declare a master agent which contains the scheduling of coordination. But this approach has drawbacks. If the description is inside an agent, a new

 ISBN: 978-1-61208-184-7

coordination cost becomes another development and there is no standardization of the approach. When an agent pilots coordination programmatically, the state of the evaluation is difficult to observe. Also, the definition of coordination should be external and then its interpretation can be done by an agent or another engine.

Because of our past experience on Web service design, we studied several existing coordination languages such as WSCI (Web Services Choreography Interface), BPML (Business Process Modeling Language), WSCL (Web Services Conversation Language), BPEL4WS (Business Process Execution Language for Web services) [12].

WSCI is a description language based on XML, which aims at describing the messages exchanged between agents into coordination. BPML is a high level language which is used to describe business process as a sequence of simple, complex activities including the interaction between participants in order to achieve a common objective. WSCL language is used to describe the business logic or public sub processes based on the definition of a web service. BPEL [11] language (Business Process Execution Language) replaces previous specifications of Microsoft XLANG and WSFL (Web Services Flow Language) from IBM. BPEL is used to model two types of processes

- Abstract process: specifies the messages exchanged between the partners, without specifying the internal behavior of each.
- Executable process: specifies the execution order of activities constituting the process, the partners involved in the process, the messages exchanged between the partners, and processing of errors and exceptions specifying the behavior in case of errors or of exceptions.

An external observer can consider a process as a mobile agent if this agent has a formal declaration. In the context of BPEL language, this description is provided as WSDL format.

BPEL is a language for describing orchestration of Web services. But inside an orchestration services are composed and often a transaction is created for the execution. We consider that BPEL specification can describe the execution order between a numbers of agents constituting the process definition, the partners involved in the process, the messages exchanged between these partners. Next, we need to define a mapping between higher order Pi-Calculus and BPEL language. It means a transformation from a formal language into a more operational language.

C. From HOPi calculus to BPEL

Some works already exist about mapping between Pi-Calculus and BPEL. Faisal Abouzaid uses a first version of Pi-Calculus based on monadic expressions and first order term definitions [18], [19]. We extend this work and adapt it to our framework of mobile agent system. Two main features are taken into account: polyadic expression and higher order term which are used for communication description. Because BPEL language is verbose and contains a lot of technical details, we have developed a strategy to generate BPEL skeleton. The choice of BPEL language involves that each component can be considered through its WSDL description. This one contains several parts such as types, messages or port type, etc. Also, we consider Pi-Calculus specification as an input source for filling not only BPEL skeleton but also WSDL declaration.

Because our input specifications are written into XML format, each step of our strategy is an elementary transformation belonging to a more global chain called BPEL generation. We use the structure of specifications to enrich all our artifacts (WSDL and BPEL).

III. TRANSFORMATION INTO BPEL SCRIPT

As we presented in Section 2.A, a Pi-Calculus specification contains a main term, called System into Figure 2. This pi-calculus process is composed by parallel and synchronizing actions. So, the underlying rules of the mapping are correspondences between Pi-Calculus terms and BPEL blocks. Identifiers are essential to propagate data and refactoring is necessary as a pre statement for preparing future generation. The first part of our transformation chain is described as follows into figure 3:

Figure 3. Pre statement from a specification to an enriched description.

A. Structural transformation

We use a top down approach; this means that we exploit the structure of a higher order Pi-Calculus specification. First, we consider the main term as a main BPEL sequence. The definitions of each sub term are considered first as partners of the script.

Process calls can contain typed arguments. Abstract data types can be specified with Pi-Calculus language as a process. Such a definition is converted into types in the WSDL description. Of course, this declaration is included first into the WSDL flow where agent is declared. But, data types are shared between several process declarations, also, it is useful to create XML schema which contains complex type. Then, XML schemas are imported into WSDL definitions of associated agent. We build a dependency graph of the definitions (data and behavior). The edges represent definition importation and communication relation (I/O). This relation is used to enrich first XML representations with annotations. These ones are about oriented actions such as input message, output message, call of agent, etc.

B. Annoted XML flow

Each transformation is built with XSL-T language. This means that we use a standard language dedicated to graph transformation. Because each XML flow can be considered as a graph, we can use a set of rules for the basic construction of Higher Order Pi-Calculus language and a rule engine to select closest rules. A rule is a template based on specific patterns of XML from the input source.

Then, an XML output is computed based on the input. Our schema allows us to check the structure of agents before and to map Pi-Calculus structure on to BPEL blocks. For instance, a sequence of actions is mapped as a BPEL sequence. More complex is the transformation of data

 ISBN: 978-1-61208-184-7

exchange. An input of data means a message output; this involves not only a type definition for the message, but also a call to an operation of another agent.

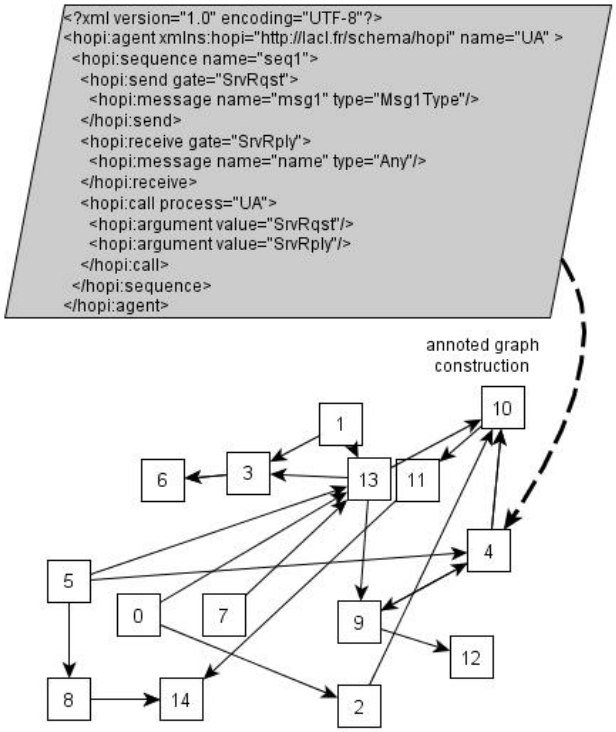


Figure 4. From a higher order Pi-Calculus into an annoted graph.

The previous figure (Fig. 4) shows an XML view of the Pi-Calculus specification of *UA* agent (SLP protocol). After enrichment, we obtained another graph where each node is an XML tag with new attributes and process annotations. More precisely, we save into annotations technical information useful for the construction of the final BPEL script and WSDL script.

C. BPEL and WSDL attributes

We have defined two main schemas; one is about useful details of WSDL language and another about useful details of BPEL language. So, we can label a node with attributes called: *partnerLink*, *variable*, *portType*, etc. For instance, if we consider a part of the specification of *UA* agent (complete definition is given in annex): an emission (1) on channel *SrvRqst*, then the corresponding labeled node is presented (2) below and the BPEL skeleton is displayed as (3):

$\overline{SrvRqst}\langle Prmt, SrvRply\rangle$ (1)

Previous parsing of input XML sources provides that *SrvRqst* channel is a link between *UA* agent and *DA* agent. Also send tag is transformed as follows,

```
<hopi:send gate="SrvRqst"  (2)
  bpel:partnerLink="DA" bpel:portType="ns1:DA">
  <hopi:message name="msg1" type="Msg1Type"/>
</hopi:send>
```

During the synthesis of all tagged graph, a part of the BPEL action is given as follows.

```
<bpws:reply name="Reply"  (3)
operation="SrvRqst" partnerLink="DA" portType="ns1:DA"
wpc:displayName="Reply" wpc:id="3">
  <wpc:input>
    <wpc:parameter name="msg1" variable="msg1"/>
  </wpc:input>
</bpws:reply>
```

Receive and reply activities go hands in hand in a request-response flow. After this output message, *UA* agent receives detail about *Print* service. We follow the same approach as before, first the Pi-Calculus term, then the tagged node and finally BPEL action.

$SrvRply(Name)$ (4)

As before, the dependency graph provides that the input channel is a link between *UA* agent et *IdleDA* agent.

```
<hopi:receive gate="SrvRply"  (5)
  bpel:partnerLink="IdleDA"
  bpel:portType="ns1:IdleDA">
  <hopi:message name="name" type="Any"/>
</hopi:receive>
```

Finally, a BPEL action is:

```
<bpws:receive createInstance="yes"  (6)
name="Receive" operation="SrvRply" partnerLink="IdleDA"
portType="ns1:IdleDA" wpc:displayName="Receive"
wpc:id="2">
  <wpc:output>
    <wpc:parameter name="Name" variable="Name"/>
  </wpc:output>
</bpws:receive>
```

Receive activity is known as blocking activity as in Pi-Calculus. This means it will wait till any message received. And it will create a new process instance. Inside the receive activity an output element is specified which refer to the request variable. The request variable data can be used in other activity in the business process.

Then the definition of the agent ends with a call to UA definition. We follow the same approach as before, first the Pi-Calculus term, then the tagged node and finally BPEL action.

$\nu\, Print\, UA\langle SrvRqst, SrvRply\rangle$ (7)

The dependency graph offers a lot of metrics such as scope and depth. Also we can label the XML tags as follows:

```
<hopi:call process="UA"  (8)
  bpel:partnerLink="UA"
  bpel:portType="ns1:UA"
  bpel:operation="process">
  <hopi:argument value="SrvRqst"/>
  <hopi:argument value="SrvRply"/>
</hopi:call>
```

The depth of this call corresponds to a distance computation in dependency graph. The closest definition is considered as a solution.

```
<invoke partnerLink="UA" portType="ns1:UA"  (9)
operation="process" inputVariable="UARequest"
outputVariable="UAResponse" />
```

UARequest and *UAResponse* are declared as local variable of the process definition. Their type is automatically computed from the input XML source. In our case, *UARequestType* is a couple of information and *UAResponseType* is a Boolean value as acknowledgment. We could detail all the primitive of the syntax presented before but the size of the document does not allow us to give more details about them. We have presented these three actions because they support higher order feature of the formal language. In definition (2), the Msg1Type can be the definition of another agent. This means that the message is linked to the port type of the mobile agent. So, the agent host

 ISBN: 978-1-61208-184-7

can then invoke all operations presented into the description of this port type.

D. Synthesis and control

After creating the upper part of the WSDL definition of agent, controls have to be applied to validate relations between the agents of the agency. This means that BPEL skeletons are used to check partner link definition and also their role into the main script. Type checking is also applied on variable used as parameters or as local data. The objective is to provide XML definitions as good as possible to specifiers.

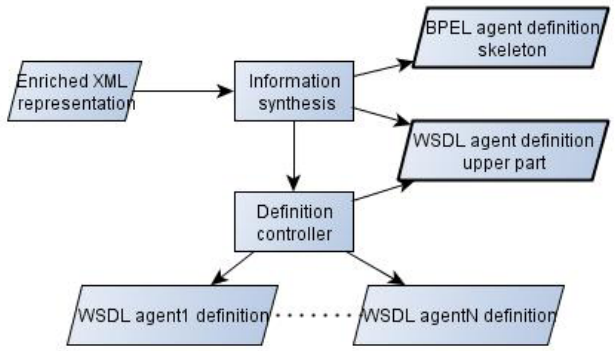


Figure 5. Information synthesis.

Previous diagram (Fig. 5) depicts the last part of our approach. As mentioned in the last section, a labeled graph is synthesized to obtain a couple of description (bold lines). Next controls are done by the use of type definition as XML schema and agent definition as WSDL definition.

IV. ARCHITECTURE DESCRIPTION

Because mobile agent system exploit network and is deployed over a set of computers, it is necessary to have a specification language that can model computers at a higher level of abstraction and enable analysis of description. The language should be powerful enough to capture high-level description of software architecture. On the other hand, the language should be simple enough to allow correlation of the information between the specification and the architecture manual.

Architecture Description Languages (ADL) enables design automation of embedded processors [21]. The ADL specification is used to generate various executable models including simulator, compiler and hardware implementation. This language is a reference in the architecture specification domain but it is not natural to compose such a specification with other process algebra specification.

A. Agent host and neighboring

We consider each node of our network as a future host for receiving mobile agents. Also, it is essential to describe which the local services available are for an incoming agent. More precisely, this can be viewed as a first security layer where local services are callable under condition on the role of the caller.

We need a description language for our software architecture which can be composed with HIGH ORDER (H.O.) Pi-Calculus. Matthew Hennessy proposes a process algebra called SafeDPi, which is based on Pi-Calculus. This is an extension used to type processes depending on their location [22]. Also, this is precisely what is important in our context of partner link and end point definition. If a unique resource location (URL) is used to call a Web service. We need to express this uniqueness into our specifications and define migration of agent based on this feature. Moreover, SafeDPi language is defined to embed higher order Pi-Calculus definition of agent host [23].

So, software architecture takes the form:

$$[\![host]\!]_1\,[\![UA]\!] \mid (new\ SrvReg{:}E)([\![host]\!]_2\,[\![SA]\!] \mid [\![host]\!]_1\,[\![DA]\!]) \quad (10)$$

Where there are two agents *UA* and *DA*, which are on the same location $host_1$, and the agent *SA* is running on another location called $host_2$. The agents *DA* and *SA* share information called *SrvReg* which is the gate to publish a service into the registry. The agents *UA*, *SA* and *DA* are defined with H.O. Pi-Calculus language.

We use this language to provide a formal description of all nodes which can host mobile agents. New location can also be taken into account as a new configuration of the network. So, our case study use 3 agents on 2 distinct nodes called $host_1$ and $host_2$.

$$location[host_1 : Host_1, host_2 : Host_2] \quad (11)$$

The type of the locations defines which kind of agent can be deployed on it. (12)

$$Host_1 = location[SrvRqst{:}w(DA), SrvRply{:}r(IdleDA)]$$

This definition stresses that a node of type $Host_1$ can support an agent which exploits a couple of resources called *SrvRqst* for sending a message to *DA* and *SrvRply* for receiving a message from *IdleDA*. The location definition should be completed to support also a *DA* agent. The $Host_2$ definition is defined with the same approach. We use such specification as a set of constraints for the deployment step. When an agent is deployed or installed initially on a node which is specified as previously; we can checked whether the communication services are compatible.

As in Section 2, we have defined an XML schema for SafeDPi-Calculus language. A deployment specification is an XML flow and we compare provided services of a node like $host_1$ with the required services used by *UA*. This is done importation of an XML flow into another one and the control of invoke, receive and reply actions.

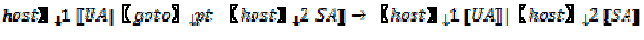
B. Local resources access and migration

Previously, we specified message types. Also, channel can also be types. For instance, a link between $host_1$ and $host_2$ can only support message of type *Document*. Also, we have added type on communication link. Now, we restricted the definition (12) into (13) to limited value on gate.

$$Host_1 = location\begin{bmatrix} SrvRqst{:}w(Msg1Type{:}DA), \\ SrvRply{:}r(Msg2Type{:}IdleDA) \end{bmatrix} \quad (13)$$

Now, we can check message type from agent specification and possible message type into deployment specification. But we need to have more control about the definition of node. Also want to express that from node $host_2$, it is possible to move to $host_1$ but not from another

node. Also, we place oriented links between nodes. So, location declaration can be enriched as follows:



This expresses that on $host_1$ the agent *UA* can be placed in parallel with a mobile agent with the ability to move towards $host_2$. Then, after the migration, each code can continue its evaluation on two separate nodes. A constraint is added on the migration statement: this will occur though the port type called *pt* on $host_2$. This allows us to add new deployment restrictions. Because these restrictions can be checked into agent specification, we can raise anomalies if a rule is not respected.

C. SLP case study

As presented before, this protocol is suitable for our presentation. It simulates the need of a print service by a client called *UA*. The service *print* is published by an agent called *SA* into a registry called *DA*. In nominal scenario, when *UA* agent asks the registry to know where the *print* service is, then it receives the service on the node where it is. Thus, we have to express code mobility and agent moves from $host_2$ to $host_1$. The data to be printed do not move on the network. This part of the description is just specified but the use of one specification level called software specification. Then, physical constraints are described through another level of specification called deployment specification. Because the formats of these specifications are compatible, we are able to combine them and check is software constraints are satisfied through physical constraints.

V. CONCLUSION

Through this paper, we have described a process to generate executable representation from formal specification. Of course, this work is currently prototyped through several examples and we need to complete our BPEL generator to help designer into his business process definition.

We think that design of distributed application can evolved by the use of mobility feature. Engineer has to separate the concerns: a level for software component and another for the deployment. With mobile agent, placement is not frozen from load time. But depending on runtime, mobile agents can move component and adapt initial placement as a new configuration for continuing the execution. Next direction is to provide our work to project partner for deeper validation. The extensions of Pi-Calculus language are as rich as extensions of BPEL language, also we are confident in our approach to assist business analyst in a more formal approach and check business property of his whole system.

REFERENCES

[1] C. Moemeng, V. Gorodetsky, Z. Zuo, Y. Yang, and C. Zhang, Agent-Based Distributed Data Mining: A Survey, L. Cao (ed.), Data Mining and Multiagent Integration, Springer Science + Business Media, May 2009, pp. 234-246.

[2] Takahashi, H. and Kavalan, V., A mobile agent for asynchronous administration of multiple DBMS servers, Systems Management, 1998. Proceedings of the IEEE Third International Workshop on 22-24 Apr 1998 pp. 32 - 33.

[3] Ponci, F. and Deshmukh, A.A., A Mobile Agent for measurements in distributed power electronic systems, Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE, 12-15 May 2008, pp. 870 - 875.

[4] Giacomo Cabri, Letizia Leonardi, and Franco Zambonelli, Mobile Agent Coordination for Distributed Network Management, Journal of Network and Systems Management Volume 9 Issue 4, Dec 2009.

[5] Verdejo, A. and N. Martí-Oliet, Implementing CCS in Maude 2, in: F. Gadducci and U. Montanari, editors, Proc. 4th. Intl. Workshop on Rewriting Logic and its Applications (2002)

[6] M. Große-Rhode. A Compositional Comparison of Specifications of the Alternating Bit Protocol in CCS and UNITY Based on Algebra Transformation Systems. In K. Araki, A. Galloway, and K. Taguchi, eds., Proceedings of the 1st International Conference on Integrated Formal Methods (IFM'99), pages 253–272, UK, 1999. Springer Verlag.

[7] B. C. Pierce, D. R#my, and D. N. Turner. A typed higher-order programming language based on the picalculus. In Workshop on Type Theory and its Apphcation to Computer Systems, Kyoto University, July 1993.

[8] R. Milner. Communicating and Mobile Systems: The Pi-Calculus. Cambridge University Press, Cambridge, UK, May 1999.

[9] Davide Sangiorgi. Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms. PhD thesis, LFCS, University of Edinburgh, Avr 1993.

[10] C.Perkins and E. Guttman. Service Location Protocol (SLP), Version 2. Sun Microsystems, http://www.ietf.org/rfc/rfc2608.txt.

[11] F. Curbera et al. Business process execution execution language for web services, version 1.0. Standards proposal, BEA Systems, International Business Machines Corporation, and Microsoft Corporation, http://www-106.ibm.com/developerworks/library/ws-bpel/, 2003.

[12] Siebel. Business process execution language for web services bpel4ws, version 1.1. http://www.siebel.com/bpel, 2003.

[13] Dave Clarke. A Basic Logic for Reasoning about Connector Reconfiguration. Fundamenta Informaticae 81(4):361-390, Jun 2008.

[14] Sascha Kluppelholz and Christel Baier. Symbolic model checking for channel-based component connectors. Science of Computer Programming 74(9):688-701, Sep 2009.

[15] Sun Meng, Farhad Arbab, and Christel Baier. Synthesis of Reo circuits from scenario-based interaction specifications. Science of Computer Programming 76(8):651-680, Avr 2011.

[16] Carolyn Talcott, Marjan Sirjani, and Shangping Ren. Comparing three coordination models: Reo, ARC, and PBRD. Science of Computer Programming 76(1):3-22, May 2011.

[17] Farhad Arbab. Elements of Interaction. In Marc Aiguier, Francis Bretaudeau, and Daniel Krob, editors, Complex Systems Design & Management, pages 1-28. Springer, 2010.

[18] Faisal Abouzaid. Toward a pi-calculus based verification tool for web services orchestrations. In Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS06), Paphos 2006.

[19] F. Abouzaid, "A Mapping from Pi-Calculus into BPEL", in Proc. ISPE CE, 2006, pp. 235-242.

[20] Uwe Nestmann and Frank Puhlmann: Business Process Specification and Analysis. In Process Algebra for Parallel and Distributed Computing. Boca Raton, Chapmann & Hall/CRC Press (2009) pp. 129-160

[21] ANSI/IEEE Std 1471™-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.

[22] Matthew Hennessy , Julian Rathke , and Nobuko Yoshida . SafeDPi: A language for controlling mobile code (2003). In Proc. FOSSACS, LNCS 2987

 ISBN: 978-1-61208-184-7

[23] C Lhoussaine. Type inference for a distributed pi-calculus (Jun 2003). Science of Computer Programming

ANNEX

$$\mathbf{DA(SrvReg, SrvUnReg, SrvRqst, SrvAck)} = \nu(input, reset, channel, inputIdle, resetIdle)$$
$$(SrvReg(S_{reg}).((DA_{Mem}(input, reset, channel, inputIdle)$$
$$|\, IdleDA_{Mem}(input, resetIdle, channel, inputIdle)\,|\,\overline{input}(S_{reg}).\overline{SrvAck})$$
$$+\, SrvUnReg(S_{reg}).\overline{reset}.\overline{resetIdle}.\overline{SrvAck})$$
$$|\, SrvRqst(S_{rqst}, ch).\overline{channel}(S_{rqst}, ch))$$
$$.\, DA(SrvReg, SrvUnReg, SrvRqst, SrvAck)$$

$$\mathbf{UA(SrvRqst,\ SrvRply)} = \nu(Print)$$
$$\overline{SrvRqst}(Print, SrvRply).SrvRply(Name).UA(SrvRqst, SrvRply)$$

$$\mathbf{IdleDA_{Mem}(input,\ resetIdle,\ channel,\ inputIdle)} =$$
$$inputIdle(S_{reg}).channel(S_{rqst}, ch).[S_{reg} = S_{rqst}]\,\overline{ch}(S_{reg}).$$
$$\overline{input}(S_{reg}).IdleDA_{Mem}(input, resetIdle, channel, inputIdle) + resetIdle.0$$

$$SA_1(SrvReg_{SA_1}, SrvDeReg_{SA_1}, SrvAck_{DA_1}) =$$
$$\overline{SrvReg}_{SA_1}(Service(print, f)).SrvAck_{DA_1}$$
$$.\,\overline{SrvDeReg}_{SA_1}(Service(print, f)).SrvAck_{DA_1}$$
$$.\,SA_1(SrvReg_{SA_1}, SrvDeReg_{SA_1}, SrvAck_{DA_1})$$

 ISBN: 978-1-61208-184-7

Optimized Testing Process in Vehicles Using an Augmented Data Logger

Karsten Hünlich
Distributed Systems Engineering GmbH
Esslingen, Germany
karsten.huenlich@distributed-systems.de

Ulrich Bröckl
University of Applied Sciences Karlsruhe
Karlsruhe, Germany
ulrich.broeckl@hs-karlsruhe.de

Daniel Ulmer
IT-Designers GmbH
Esslingen, Germany
daniel.ulmer@it-designers.de

Steffen Wittel
Distributed Systems Engineering GmbH
Esslingen, Germany
steffen.wittel@distributed-systems.de

***Abstract*—The growing amount of electronic components in vehicles requires an increasing communication load between these components and hence an increasing load on the vehicles communication buses. Both aspects entail an increasing work load for the test engineer developing and executing test cases to verify the required system behaviour in the vehicle. This paper considers a way to automate and reduce the workload for in-vehicle testing by augmenting the functionality of current data loggers. The idea is to use the data logger for supporting the testing process for test drivers. The introduced implementation shows a way to verify the test cases' execution on the fly in order to avoid finding erroneously executed test cases at a later point in time. Additionally, the presented implementation seamlessly includes the test environment for in–vehicle testing into the tool chain which is already used on lower integration levels. This allows the test engineer to reuse test cases from the lower integration levels in vehicle tests and to compare the results from test runs on different integration levels. The paper ends with a summary of the feedback collected in a case study with a prototypical implementation.**

Keywords-automotive; data logger; intelligent data logger; test case development; test case monitoring

I. INTRODUCTION

Many data loggers in the automotive industry are designed to record the communication between Electronic Control Units (ECUs) [1]. In more advanced systems, the data content of the RAM (Random Access Memory) of the ECUs is additionally recorded [10]. These data loggers become more and more important to the test engineers because the number of the networked ECUs and hence the testing efforts in a vehicle is continuously increasing. From each requirement on vehicle level the test engineers have to derive test cases to ensure that the ECUs in a vehicle are performing the correct action within correct time constraints. To check this in an in-vehicle test it is necessary to record the bus traffic and the data content of the ECUs' RAM while executing a test case manoeuvre with a car. The result of the test is determined by evaluating the recorded data.

The amount of collected data can turn the evaluation of the recorded test case data into a time consuming challenge. The result of the evaluation can be classified as "valid", "failed" or "not valid". In case of a "passed" classification the recorded data show that the System under Test (SuT), e.g., an ECU, exhibited the expected behaviour described by the requirements. The classification "failed" shows a deviation of the measured data from the expected values and hence from the expected test result. The "error" case indicates a significant mistake during the test case execution that makes an evaluation of the recorded data impossible with respect to the test case's definition.

To minimize the error cases, and therefore the time for the test case execution and evaluation, the data logger can be augmented with additional functionality to monitor the correct execution of the test case. The necessary conditions are to be defined by the test engineer. The data logger can be extended with instructions supervising relevant signals. For these signals boundaries may be defined. A test case can, e.g., be successfully accomplished if the signal stays within these boundaries. However, the goal is not to test the driver's behaviour as mentioned in [2]. The goals are to give instructions to the test case executor, which may be a driver, a robot or a test automation tool, and to additionally supervise the execution's correspondence to the conditions predefined by the test engineer. Especially for a human driver being in the light of our experiences, the biggest error source in a vehicle during a test case execution the augmented data logger can help to avoid unnecessary work by immediately indicating erroneous test runs. For example, the augmented data logger can supervise an Antilock Braking System (ABS) manoeuvre where the driver speeds up to 60km/h and does a full brake without turning the steering angle.

Figure 1 shows an example of a system development process according to the V-Model as shown in [3]. In this example, the test on vehicle level is the last level of testing

 ISBN: 978-1-61208-184-7

within the integration process. Before this stage many other tests have already taken place on lower integration levels. For efficiency reasons, it would be helpful if the test engineer could reuse test cases developed on lower integration levels, e.g., test cases from Hardware in the Loop (HiL) tests [4]. The reuse of these test cases minimizes the work for the test engineer to adopt the test cases to the desired test platform. The reuse also enables the comparability of the test results with the lower integration levels. For guaranteeing the reusability of the test cases it is essential to specify the test cases platform independently. This format is interpreted by certain testing platforms.

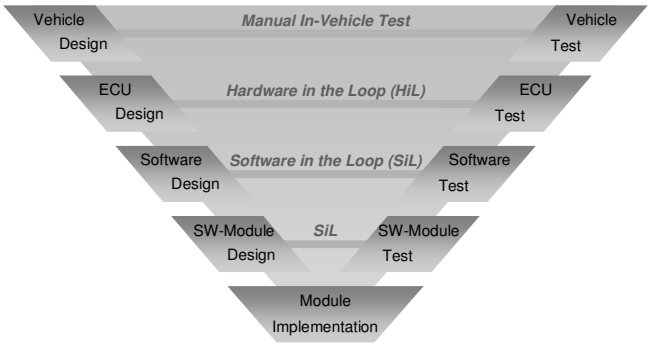


Figure 1. Commonly used application of the V-Model in the automotive industry

This paper describes a solution to reuse test cases from lower integration levels by adding information to guide the driver through the test case and to supervise the actions of the driver on the basis of the test case implementation of the test engineer. It starts with the description of the software and hardware components that extend the data logger to an augmented data logger. The paper ends with a summary of the feedback collected in a case study using a prototypical implementation.

II. STATE OF THE ART

Today in-vehicle tests are usually executed without the support of a software tool for giving feedback on the quality of the test execution or a tool that guides the driver through a test case. The test cases are often written in plain human readable text which describes what a tester has to do in the vehicle to fulfill the test case. These textual test cases are stored for example in a database. For taking a set of test cases to the car, they are either printed out or downloaded to a robust handheld computer. In both cases, they are read before or during a driving manoeuvre. The quality of the execution of the manoeuvre thus depends on the skills of the test driver. Details of the execution quality can be determined offline on a parking lot or by evaluating the information on the data logger. Especially if test driver and test engineer are not the same person, this process is error prone and time consuming. Since the test cases are in natural language there is enough room for misunderstandings between a test manager who writes the test cases and a test driver who has to execute the manoeuvre. This fact tends to result in multiple iterations of in vehicle tests of the same test case.

There are several solutions that have the aim to optimize in vehicle tests and to minimize the time overhead. A touch-display can be used in vehicles for check lists. A more advanced system is shown in [12] which comprises of a driver guidance system and a feature to immediately evaluate if the test is passed or failed.

For testing driver assistance functions, manoeuvres have to be executed very precisely by the test driver. That means in a significant number of tests the tests are failed not because the system is not working correctly but the test driver has made a mistake. To minimize this number of invalid tests this paper describes a way to detect deviations of the given test case during the execution. This avoids a usually time consuming evaluation of invalid test cases.

III. AUGMENTED DATA LOGGER

Current data loggers [10] are designed for recording data and neither for interpreting it nor for participating in the measurement process. This section describes a way of augmenting the functionality of the data logger in order to support the testing process and to seamlessly integrate the vehicle tests in the system integration and testing process.

A. Basic Data Logger System Design

A data logger to record digital information in vehicles might be designed in the way described in [5]: i.e., a host computer is connected via a network interface, e.g., Ethernet, to the data logger. Over this connection, the data logger can be controlled and configured. The configuration defines which signals are stored in the data storage and on which bus interface the signals can be received. The host computer is mainly used to start and stop the data logger and to visualize an excerpt of the recorded data on the fly. The data logger hardware is responsible for the real time processing of the data. A commonly found feature might be a trigger that starts a measurement when a predefined condition becomes true as it is described in [6].

For evaluating the trigger conditions the data logger needs information about the connected data buses and the data that is transferred over a particular data bus. Usually, this information is available in form of configuration and signal files that are interpreted by the host computer and transferred to the data logger.

In some parts of a data logger execution in real-time is mandatory. This is necessary because the test engineer needs to know exactly when some data have been transmitted on a particular bus. A common solution is that the communication on a bus system is recorded together with timestamps which indicate the time instance when a message is transferred over a bus [9]. Figure 2 shows the procedure of recording a message from a bus. If the data logger receives a message a timestamp is taken. For the evaluation of the recorded data it is possible to correlate in time the different recordings with the help of the timestamps, which means that the more precisely the timestamp is taken the more precisely the situation can be reproduced and evaluated.

 ISBN: 978-1-61208-184-7

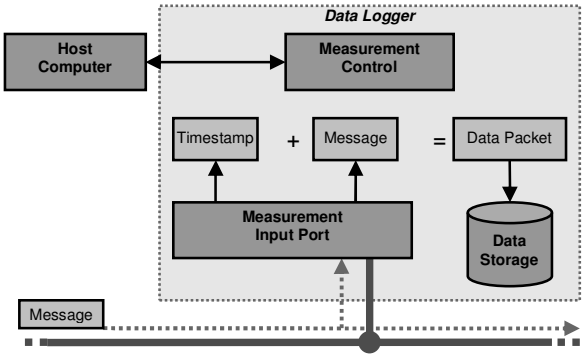


Figure 2. Schematic procedure of measuring a message on the bus

The example in Figure 2 shows a host computer which is connected over a communication interface with the measurement control unit within the data logger. The host computer is commonly a PC or a notebook with an operating system that does not support real time tasks. Via the host computer the engineer has access to and control over the data logger. Additionally, the host computer can access measurement data and visualize them to the user. Evaluating this data while conducting a manoeuvre is almost impossible since, in this case, the driver would have to fully concentrate on the monitor instead on his driving task.

B. Current Testing Process on Vehicle-Level

In the common testing process the test engineer starts looking at the requirements for the SuT. Based on these requirements the test engineer creates the corresponding test cases. How the test engineer writes down these test cases for in-vehicle tests is normally not defined. In some way, the test cases have to be readable by the driver while he is executing the manoeuvre in the vehicle. After finishing writing a test case, the test engineer has to hand over the test case to the driver who executes the manoeuvre specified in the test case in the vehicle. The role of the test engineer and of the driver might be taken by the same person or by different ones. If the test engineer and the driver are different persons who write and execute a test case the test case must be well defined to prevent misunderstanding. If the test case specification is not complete and therefore the driver does not execute the test case as intended by the test engineer, the following work might be unavailing.

After having recorded the data of the manoeuvre that is specified in the test case the driver hands over the recordings to the test engineer. Afterwards the test engineer evaluates the data. Usually this is done manually. The test engineer has to search through a database of signals with probably more than 10,000 entries. If the result of the test case is “passed”, the test case will be documented and closed. In case the result is “failed”, the test engineer has to find the exact reason. The SuT can either have a bug or the test case has not been executed accurately which means that the test is “not valid”. If the test case was executed within all defined constraints by the test engineer the test case is “valid”. Both cases generate lots of work of analyzing and documentation for the test engineer. Especially, the work for the second case can be minimized by finding out the validity of the test case in an earlier stage of the process.

Generally, the biggest drawback of finding invalid test runs late in the process is the time that the test engineer spends on one test case. It must be considered that the number of test cases that must be performed for each major release can be up to several hundred test cases. As a conclusion two main issues can be identified that can be possibly optimized:

- The time for evaluating the test results by avoiding invalid test cases
- The numbers of times moving from the office to the vehicle and to the test track for repeating invalid test cases

The introduced testing process on vehicle level is very different from the test processes on lower integration levels of the development process shown in Figure 1. In the lower levels, i.e., HiL or SiL, a test case is written in a defined way. The test case can be reused and usually returns a reproducible result. Another point is that the test result is directly available after the test has been finished. It can be said that the processes on different levels have mainly five important parts [7]:

- Environment simulation that simulates the environment of the SuT
- Test case execution system
- The System under Test itself
- Measurement and data logging system
- Evaluation system

The evaluation system compares the measured values with the ones that are specified in the test case for the SuT. The test case execution system reads the test case and controls the environment simulation that affects the SuT. In a vehicle, the parts for the test process are different. The test case execution system in a vehicle is the test driver. The test driver has control over the environment of the SuT. The evaluation system in a vehicle test is the test engineer who evaluates the measurements.

The measurement and data logging system might be the same as the one used in the vehicle. For the in-vehicle test, an environment simulation is not necessary because the vehicle is used in a real environment. Sometimes both environments are mixed for the vehicle tests, e.g., foot passengers are simulated with synthetic dolls or imaginary sensor information.

C. New Testing Process Supported by the Augmented Data Logger

To reduce the time for testing and evaluating of in-vehicle testing a new approach for the testing work flow should be considered. The first aspect is the form how the test case is written. A uniform platform independent language (see Section III C. for more detailed information) is used to define the test cases. With this uniform language, the test engineer can precisely describe the test case. The test case is now not only human readable but also machine-readable and can be interpreted by a programme. The

 ISBN: 978-1-61208-184-7

additional code extends the abilities of the data logger. The system now knows about the manoeuvre which has to be executed for a particular test case. With the knowledge of how a test case must be performed, driving errors can be detected directly and time can be saved.

The new work flow has a strict separation between the office work and the work in the vehicle. Right after performing a test case, the driver gets a result if the test case was executed accurately. The feedback also includes the information why the test failed. This information depends on the test case description from the test engineer. If the test engineer describes the test case in many details more driving errors and failures can be detected without looking at the whole measured data back in the office. The advantage of this new approach is that the driver:

- Is guided through the test case execution process through a unified notification
- Gets a response directly after the manoeuvre if the test is executed correctly
- Gets the reason why a test case was classified as "not valid"

This reduces the evaluation work and the test case execution work. Since the data logger instructs and checks the manoeuvre it makes the execution more precise.

For this new approach parts of the evaluation system and the test case execution system are moved into the data logger. The schematic of a data logger shown in Figure 2 can be extended to execute additional code given by the test engineer that controls the data logger and guides the test driver through the manoeuvre. Figure 3 shows a simplified version of the extension of a measuring system. The CPU has to fetch the messages from the bus, add a timestamp to each message and extract the relevant signals with its values. The values of the signals are internally updated and provided for the test case code.

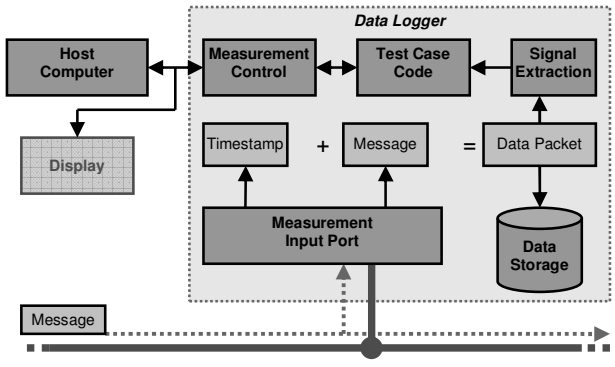


Figure 3. Schematic measuring system extended with the test case code

To control and configure the data logger the test case code needs a connection to the measurement control module. It starts or stops the test case and gets commands from the test case code to control, e.g., the data logging. All information that is known in the data logger can be used inside the test case code. The measurement control module has also the task to route the instructions for the test driver to the host computer or a connected display.

IV. TEST CASE IMPLEMENTATION AND EXECUTION IN A VEHICLE

In this section, the test case implementation and execution is shown using the following example: The driver starts the engine and accelerates the car to 60km/h. When 60km/h are reached, the driver performs a full braking. In this manoeuvre it is important that the driver keeps the steering wheel straight.

Such a manoeuvre is used, e.g., to measure data of an ABS and to evaluate if it has performed accurately during its intervention. A possible criterion for a "not valid" ABS-test execution is defined by looking at the steering angle. If the data show that the car did not drive straight, the test case has not been executed accurately. The manoeuvre can be described in a state chart manner represented by an XML file [8].

The example in Figure 4 shows the ABS-manoeuvre in XML code. The definition of the XML code is described by Ruf [11] for Hardware in the Loop tests. The test case is composed of states, actions, events and one rule. The rule checks the steering wheel angle for the whole test case. The states are following in chronological order. Each state has one or more actions that have to be performed by the test driver. If the condition of an event is fulfilled the state machine enters the next state.

```
<?xml version="1.0" encoding="UTF-8"?>
<Testcase xmlns="http://www.it-designers.de/adl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.it-designers.de/adl
http://www.it-designers.de/adl">

  <Rule SteeringAngle_deg_eqal="0" Tolerance_deg="5"/>

  <State num="1">
    <Action text="Get ready to start the manoeuvre"/>
    <Event wait_seconds="5"/>
  </State>

  <State num="2">
    <Action text="Start the engine"/>
    <Event wait_seconds="5"/>
  </State>

  <State num="3">
    <Action text="Accelerate to 60km/h"/>
    <Event velocity_kmh_equal ="60"/>
  </State>

  <State num="4">
    <Action text="Full braking"/>
    <Event velocity_kmh_equal ="0"/>
  </State>

  <State num="5">
    <Action text="Turn-off engine"/>
    <Event wait_seconds="5"/>
  </State>

  <State num="6">
    <Action text="Manoeuvre finished"/>
    <Event wait_seconds="3"/>
  </State>

</Testcase>
```

Figure 4. Listing o f a test case in XML

The "*wait_seconds*" events are needed to give the test driver time to perform the actions. The data logger is able to interpret the XML code and guide the test driver through the

 ISBN: 978-1-61208-184-7

described manoeuvre. To start the manoeuvre in the vehicle the test driver has to start the data logger.

In summary, the test cases that are written for lower integration levels using this XML language can be reused for in vehicle testing.

V. CASE STUDY

A case study has been performed to determine the benefits of the augmented measurement system for test drivers. The case study was conducted with a group of eleven candidates. The group consisted of team leaders, developers and testers. In the first step the content of the executed XML test case was explained to the candidates. With this knowledge the candidates were guided by the augmented measurement system to execute the test case in the role of a test driver.

The vehicle for the case study was equipped with an extra display that is attached to the windscreen. The setup in the vehicle looks similar to an external navigation system attached to the windscreen. In this setup the display shows the instructions and the current state of the running test case. The execution of the test case was done on a locked test track. This assures a save environment and that the candidates are not disturbed by surrounding vehicles.

After executing the test cases multiple times the candidate was interviewed about the experience with the augmented measurement system. The collected feedback is summarized in Figure 5.

Most candidates are confused and distracted by the display driving the test case for the first time. The reason might be that the candidates do not yet intuitively follow the instructions on the display. As soon as the instructions are known to the candidate he can concentrate less on the display and more on his driving task. After a short learning curve the confidence and sureness working with the augmented data logger raised. In summary, 7 candidates are seeing a benefit of such a system to speed up and assist them in their daily work.

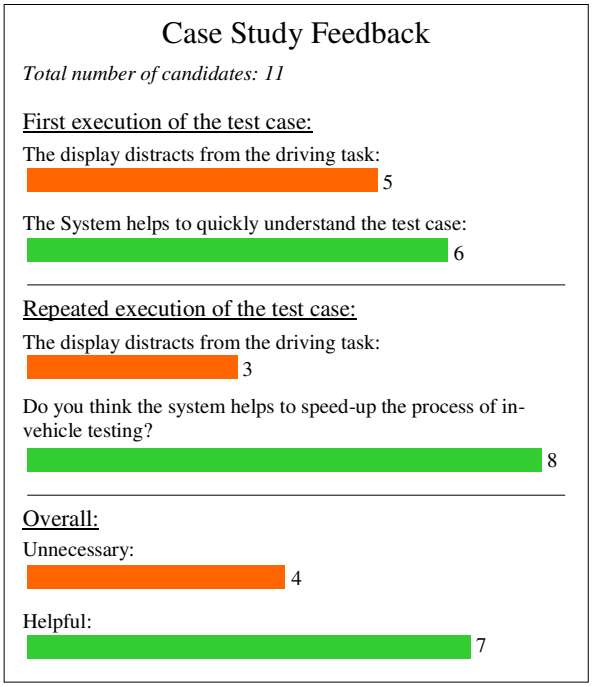


Figure 5. Case Study feedback results

Furthermore, the feedback also includes suggestions for improvements. The four mostly mentioned suggestions were:

- Additional speech output for instructions
- Direct connection to quality and lifecycle management tools
- More detailed information in case of a “not valid” result
- Using LCD glasses instead of a display attached to the windscreen

The feedback of the case study indicates that the augmented data logger helps to speedup the testing process for in-vehicle testing.

VI. CONCLUSION AND FUTURE WORKS

This work shows an approach how the process of in-vehicle testing can be improved. The introduced approach shows a way to reduce the costs for the testing process by reusing test cases from other testing platforms and by optimizing the workflow of in-vehicle testing. A major part in the optimized workflow is the possibility for declaring a test case “valid”.

The extended classification of a test case enables an early feedback about the quality of the executed test case and hence makes sure that only valid test cases are evaluated. In the introduced approach a test case can be classified as "passed", "failed", "valid" and "not valid". The first two classifications are based on the requirements for the SuT while the other two classifications reveal if the test case was executed within defined constraints that are based on additional testing requirements. The test engineer has only to look at the measurements of the test cases that are classified

 ISBN: 978-1-61208-184-7

as "valid". This helps to reduce the evaluation time especially if the test case manoeuvre is very complex or time critical.

A tool chain supports the test engineer during the test case development process. He can use test case descriptions from lower integration levels and use them as a basis for the in-vehicle test. The test engineer needs no knowledge in programming languages for implementing and running a test case on the introduced augmented data logger. Several test cases can be downloaded to the data logger and are automatically executed.

While driving a test case the test driver has precise instructions on his current tasks and is guided through the test case manoeuvre. The test driver has immediate feedback if the constraints of the test case added by the test engineer are fulfilled. The augmented data logger observes the execution and the driver gets a response if the manoeuvre is "valid" or if the test driver has made a mistake during the execution. It is then up to the test driver to decide if he wants to immediately repeat the manoeuvre or continue with the next test case.

To further improve the system it is planned to work on the interface between the augmented data logger and the test driver. The visualization of the manoeuvre can be optimized by using intuitive icons, by using speech output or even by an augmented reality display.

Another benefit of the introduced augmented measurement system is to being able to reuse identical test case descriptions in XML on several test platforms. It is left for future work to investigate on comparing the results of e.g. a Hardware-in-the-Loop platform with the results from an in-vehicle test which is guided by the augmented measurement system.

REFERENCES

[1] K. Athanasas. Fast Prototyping Methodology for the Verification of Complex Vehicle Systems. Dissertation, Brunel University, London, UK, March 2005

[2] Petersson, L., Fletcher, L., and Zelinsky, A. 2005, 'A framework for driver-in-the-loop driver assistance systems', Intelligent Transportation System Conference 2005: Proceeding of an IEEE International conference, 13. – 15 September 2005, Vienna, Austria, pp. 771 – 776.

[3] Meier, E., February 2008, 'V-Modelle in Automotive-Projekten', AUTOMOBIL-ELEKTRONIK Journal, pp. 36 – 37

[4] Schlager, M. 2008, 'Hardware-in-the-Loop Simulation', VDM Verlag Dr. Mueller e.K., ISBN-13: 978-3836462167.

[5] Park, J. and Mackay, S. 2003, 'Practical Data Acquisition for Instrumentation and Control Systems', An imprint of Elvester, ISBN-10: 075-0657-960.

[6] Koch, M., Theissler, A., September 2007, 'Mit Tedradis dem Fehler auf der Spur', Automotive Journal, Carl Hanser Verlag, pp. 28 – 30.

[7] S. Dangel, H. Keller, and D. Ulmer. Wie sag' ich's meinem Prüfstand? RD Inside, April/Mai:7, 2010.

[8] B. Ruf, H. Keller, D. Ulmer, and M. Dausmann. Ereignisbasierte Testfallbedatung - ein MINT-Projekt der Daimler AG und der Fakultät Informationstechnik. spektrum 33/2011, pp. 68–70,.

[9] D. Ulmer, A. Theissler, K. Hünlich. PC-Based Measuring and Test System for High-Precision Recording and In-The-Loop-Simulation of Driver Assistance Functions. Embedded World 2010.

[10] Simon McBeath (2002). Competition Car Data Logging: A Practical Handbook. J. H. Haynes & Co.. ISBN 1-85960-653-9.

[11] B. Ruf, H. Keller, D. Ulmer, M. Dausmann. Ereignisbasierte Testfallbedatung, Spektrum 33/2011 pp. 67 – 68.

[12] mm-lab, Driver guidance system, Automotive Testing Technology International, September 2009, page 89.

 ISBN: 978-1-61208-184-7

Providing In-house Support to Disabled People Through Interactive Television

Begoña Fuentes-Merino, Miguel Ángel Gómez-Carballa, Carlos Rivas-Costa, José Ramón Fernández-Bernárdez, Rubén Míguez-Pérez, Manuel José Fernandez-Iglesias and Luis Eulogio Anido-Rifón
Instituto AtlantTIC, EE Telecomunicación
Universidade de Vigo
Vigo, Spain
{miguelgomez, carlosrivas, rmiguez, manolo, lanido}@det.uvigo.es, {bfuentes, jramon.fernandez}@uvigo.es

***Abstract* — We introduce a low-cost open platform to provide services targeted to disabled people and their families. This system is deployed using mainstream consumer electronics equipment and is specifically designed to provide services at home. The proposal goes beyond state-of-the-art tele-assistance to implement the broader concept of socio-sanitary care. Thus, the final objective of this platform is to assist disabled people having specific care necessities, but considering, at the same time, their social integration and their quality of life. This is achieved through a wide range of services that are dynamically adapted to users' specific conditions and environments. The proposed solution significantly improves the quality of life at home of disabled people.**

Keywords – Teleassistance; socio-sanitary care; eHealth; disabled people; interactive television.

I. INTRODUCTION

In the past years, improvements in quality of life and life expectancy [1] and advances in Information Technologies have fostered the deployment of solutions targeted to disabled and elder people. Indeed, society urgently demands tele-assistance systems to address the needs of this population sector, to facilitate their integration and, eventually, to develop an efficient socio-sanitary system adapted to the present-day demographical and social situation. The term *tele-assistance* [2] refers to a wide range of systems used to provide remote care to old and/or physically challenged people to facilitate living in their own homes.

This paper discusses the development of a service platform that deals with a new way of understanding tele-assistance that also considers solutions to provide challenged and elder people better ways to become more integrated in society, to communicate with their families and caretakers, and to attend all their specific needs. Potential users of this platform are elderly and/or disabled people who are in a care-dependence situation. Because of the characteristics of its potential users, the system has been designed to be simple to use and adaptable to specific physiological user needs.

A standard TV set was selected as the presentation and users' interaction device. This decision was based on several reasons, including the existence of a TV set in most if not all potential users' homes, which contributes to stress the low-cost design principle and to limit users' reluctance due to the introduction of a new technological system into their daily routines. Other relevant reason for this choice is that elderly and handicapped people are usually quite familiarized with a TV set's mode of operation, which contributes to reduce adaptation and learning time.

A further aspect taken into account when designing the system was the present-day economic scene and the average financial capabilities of this population sector. For this, we tried to keep implementation and deployment costs as reduced as possible.

In order to design the most suitable architecture and software platform, we made an exhaustive analysis of several technologies. After this analysis, we selected a Linux-based Home Theater PC (HTPC) [3] as the main component at home connected to a standard TV set, complemented with an open-source media center solution,. This architecture was chosen because of its availability, its limited cost, and its support to the seamless integration of interactive services and the Internet, fulfilling all initial system requirements, namely low-cost, acceptance by potential users, modular structure enabling service personalization, reusability and extensibility.

Regarding communications, the platform captures all relevant data from users and sends it through the Internet to a central server, which is also accessible through a web-based management interface. Since users utilize the TV set and a common interface to access the provided platform services, they are configured and deployed using a common remote management platform. To increase the range of users, the system integrates additional interfaces to facilitate service access to a wide range of physically challenged users.

To sum up, the potential users of this innovative platform are elderly people or people with some kind of disability, who in most of the cases are in dependence situation. Users are provided with access to a wide range of socio-sanitary services through a low-cost, open, adaptable and extensible platform.

Section II introduces the proposed system's architecture, that is, the technical framework adopted when designing and integrating the various hardware and software components. Section III is devoted to discuss the selected service provision mechanisms and standards. Section IV discusses additional details on interface design, as this topic is instrumental to provide an adequate solution to our target audience. Finally, Section V collects the most relevant conclusions of this work.

 ISBN: 978-1-61208-184-7

II. SYSTEM ARCHITECTURE

In the next paragraph, we discuss the components that configure our service platform and their interactions. Figure 1 outlines the basic structure of the platform.

A. User Hardware Platform

To provide the TV set with the desired interactivity, several hardware devices were considered during the design phase. Among them, those that could be easily integrated in a living-room atmosphere were taken into account. Initially, game platforms were considered due to their low price, ease of use, wide acceptation, and the availability of next-generation interfaces like Nintendo WiiMote or Sony Kinect [4][5]. However, they were discarded because of the difficulties faced when trying to integrate off-the-shelf and open components or specifically designed interface devices for physically challenged people.

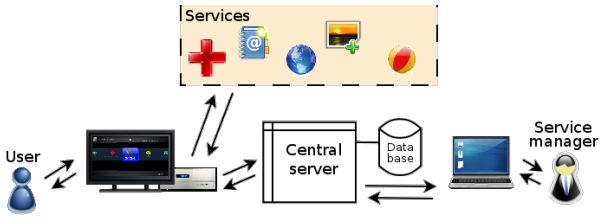


Figure 1. System Architecture outline

Other devices considered were intelligent set-top boxes (STB) [6], that is, devices whose primary purpose is the reception and decoding of multimedia signals to be displayed in a TV set. Every state-of-the-art STB includes some kind of middleware that supports the development of hardware- and software- independent applications. However, this class of systems was also discarded due to the complexity of the software development tools available and their limited support for general-purpose software applications.

The device eventually selected to be part of the final system is the Home Theater PC or HTPC, which is a personal computer specifically designed to be used as a multimedia reproduction device at home, using a standard TV set as the display unit. These devices can be defined as small full-fledged personal computers that give up processing power in favor of a noiseless, small and attractive system design. The main advantage of HTPC regarding our goals is their PC architecture, which supports system's extensibility through widely available standard interfaces. Besides, any conventional operating system may be installed in an HTPC, which contributes to facilitate its extensibility through new applications, and they provide native support for remote control devices. When compared to other solutions, HTPC provide more features at the same cost, higher connectivity, and a more complete and accessible software development environment. In addition, advanced interfacing devices like Microsoft Kinect for Xbox or Nintendo WiiMote could also be supported by the final system taking advantage of the HTPC's personal computer architecture.

B. User Software Platform

The software platform is based on an open source Linux distribution. This choice was made due to its usability, and the free, low-cost and open source orientation of the final system.

Since final users will operate the HTPC using a typical TV remote, we also introduced a media center to enable full interactivity between the remote interfaces and HTPC content. Four media center solutions were examined to choose the most suitable one fulfilling the requirements of the final system, namely MythTV, Moovida, LinuxMCE and Xbox Media Center (XBMC).

Moovida was dismissed due to its lack of maturity and the absence of a TV watching functionality or a back-end, and LinuxMCE was ruled out because of its complexity. Thus, the final choice was made between XBMC and MythTV. These are very similar systems, MythTV having outstanding TV functionalities and XBMC having greater extensibility. Taking this into account, and due to the fact that one of the main features of the final system should be its extensibility, XBMC was the final selection. XBMC supports extensibility through two mechanisms, namely plugins and scripts. Plugins are used exclusively to add new multimedia sources to the media center, while scripts allow the addition additional functionalities. Both MythTV and XBMC are supported by a large community of developers, and provide a comprehensive and clearly organized documentation, but XBMC provides an easy to use programming interface for developing new services written in Python [7], and has an intuitive and user-friendly user interface. Unlike MythTV, XBMC doesn't provide any Internet browser or a native feature to watch TV. However, watching TV was easily solved thanks to a MythTV script that can be easily integrated in XBMC, and thus used as a front-end for the MythTV back-end.

Once the software and hardware components at home were selected, the next step consisted on their integration to provide a functional system supporting interactivity with the final user. Interconnection of the HTPC with the display device, i.e., the TV set, was easily supported through high-definition multimedia interfaces (HDMI), already integrated in both devices. Then, we had to select and implement the services that will turn this platform into a socio-sanitary services platform. Before going further on this topic, it is important to remark some relevant aspects of the platform itself. As mentioned above, the TV set is used as the final presentation and interaction device. Thus, the platform is not a system itself, but a module of a larger distributed system extending the classical TV set with a portfolio of services. From the hardware point of view, the platform is a modular system extending a TV set with additional devices required by the services provided (e.g., web-cams, microphones, movement sensors, blood pressure monitors, etc.) using existing communication solutions like Bluetooth adapters, USB ports, or infrared sensors. This modular approach is a

 ISBN: 978-1-61208-184-7

key feature of the system, and contributes to limit the final cost of the system, as the only peripherals needed will be off-the-shelf components in most cases. It also enables the possibility of adapting the final platform to any specific user's needs. This is a very important aspect taking into account the fact that potential users of the system range over a wide range of situations insofar dependence is concerned. Another key feature of the final system is its adaptation capabilities to the specific needs of potential users.

C. Network Structure

In order to support user interaction both with the services provided and the outside world, to collect and retrieve content, and to manage user data, a typical three-tier service architecture was implemented and deployed. Figure 1 above outlines the main components of this architecture, which are outlined below:

- **Business logic:** It is composed of a Login Server and an Extensible Messaging and Presence Protocol-compliant server (XMPP server) [8]. The Login Server provides user authentication and system adaptation. Users contact the Login Server to sign in the system. According to users' credentials and profile information, access is granted to the appropriate services while user interfaces are adapted to the specific needs of each user. Other actions supported by this server are new users' registration or service availability assessment. The XMPP server works as an adaptation layer to provide access to the actual user-oriented services available at the platform, both internal and external. The behavior of this element depends on the type of service accessed, as discussed in Section III below.
- **Client:** Users access the socio-sanitary services offered by the system using the hardware/software platform discussed above. Client equipment provides an adapted user interface that includes the required interaction devices according to each service and user profile (e.g., TV remotes, sensors, actuators, biomedical equipment, etc.). Note that the platform is designed for disabled or elder people that may present any kind of disability and typically spend a lot of time at home watching TV. Users may access any service provided to them according to their profile. The system also provides a Web management interface to administrators, that is, an interface enabling service and system managers to perform their tasks and to access to all the relevant information about the system.
- **Back-end:** This is an information server that manages data about users and service profiles. Relevant data may be distributed among several physical servers. For example, a central server linked to the login and XMPP servers may collect al relevant information needed by these servers, and additional servers may host the data needed by final services (e.g., content server in a streaming service, or communications management server in a videoconference service).

III. SERVICE PROVISION

Once the main structure of the system has been introduced, we will discuss along the next paragraphs how different users get access to the different services provided, and how they interact with each other. Different service provision schemes will require different communication methods. This is due to the fact that each service has an associated server, which may present specific characteristics that differ from the others. In order to facilitate to the reader the comprehension of all the communication procedures, the three basic service provision methods are illustrated below.

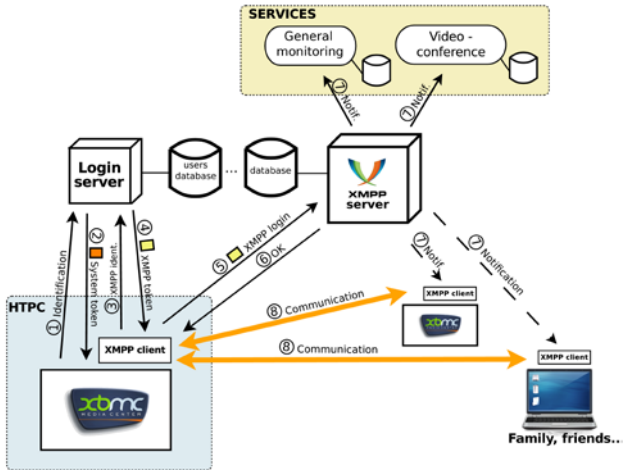


Figure 2. XMPP server communication

The platform was designed according to a quite simple but functional structure. On top of this structure all kind of services are inserted, including monitoring, communication, social integration, entertaining, rehabilitation, and educational services. After being developed, services may be added to the platform in a pretty simple way. The only requirements for service integration are that they have to be written in a Python language version supported by XBMC, and use libraries that are compatible with XBMC. These services may interact with a wide range of user devices, as long as they communicate through a standard protocol implemented in the PC platform (e.g., infrared, Bluetooth, USB, etc.).

Once the user has been signed in, an Extensible Messaging and Presence Protocol-compliant client (XMPP client) is automatically launched at the XMBC. The XMPP client requests an XMPP specific token from the login server, which contains the needed user information to login into the corresponding XMPP server.

The *User Session Management Service (USMS)* is a basic service that is included in every platform regardless of the actual user needs. This service handles authentication and personalization, that is, it has the responsibility of checking

 ISBN: 978-1-61208-184-7

the user's identity as well as requesting the central server each user's initial configuration. This information is used, for example, to provide a personalized graphic user interface, or to identify specific interaction devices needed by specific users. Upon users' logout, USMS notifies every running service that the current user session has ended.

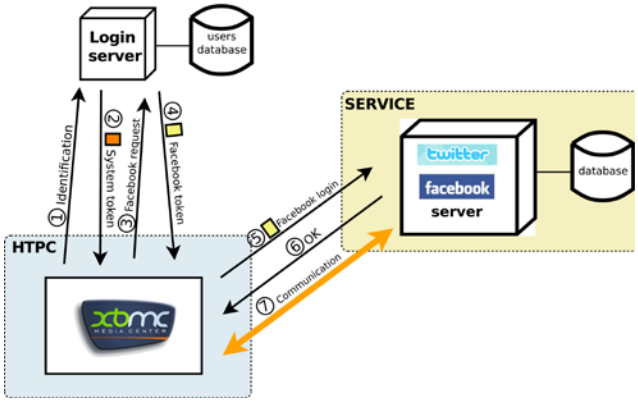


Figure 3. External server communication. Interactive services.

Once the user is logged into the XMPP server, service providers (e.g., videoconference server, medical monitoring equipment) and XMPP clients in other designated users' premises (e.g., friends, family members, caretakers) are notified. Other examples of services that require XMPP server notification reporting about the user's presence are presence control or every communication service that relies on the XMPP protocol to remotely communicate with other users. Figure 2 above outlines this scenario.

In case the user requires access to a third-party service provider (e.g., Facebook or Twitter), the system provides direct access to the corresponding external authentication facilities once initial sign-in has been completed as discussed above (c.f. Figure 3). Once the user has completed the access requirements of the external service, the corresponding access tokens are handled over to XBMC, and further communication takes place directly between XBMC and the external server, according to the provided Application Programmers' Interfaces (e.g., Facebook API).

In case the service accessed involves the provision of multimedia content (e.g., a broadcast or streaming service), the access tokens obtained after user authentication are handled over both to XBMC and to the multimedia content server (e.g., streaming server). Thus, when the user requests a multimedia content, XBMC will send the previously collected system token to the external server to authenticate itself and start the exchange of information. Then, the multimedia server will check the validity of the token and initiate multimedia content transfer. Figure 4 summarizes this process.

IV. INTERFACE DESIGN

Due to the profile of potential users, there were some relevant aspects related to interface design that were taken into account:

- People with visual deficiencies. Most elderly people have some kind of sight impairment. To overcome this situation, special care has been taken when designing the look and feel of the visual interface, and more specifically subtitles and other textual information [9]. Apart from that, eye-catching colors were used for active elements (e.g., clickable buttons) as well as easily identifiable images.
- People with lack of experience in the use of Information and Communication Technologies (ICT). Other common characteristic of the target population is that they are not accustomed to deal with new technologies. In most of the cases the use of this platform means their first contact with ICT. To address this issue, simple buttons rather than complex menus with many options have been used.
- People with cognitive difficulties. This platform is oriented to provide services both to elderly people and people with any kind of disability, including cognitive ones. Thus, it is possible that some users have difficulties to understand and reason about what is shown in the interface. Text content in interfaces has been carefully designed using simple straightforward expressions, avoiding for example complex or very long words.

The presently implemented interfaces are:

- A general common interface, where buttons are displayed following a 4x4 grid pattern, which are accessed using the four arrow keys of a standard TV remote. Apart from these four elements, OK/confirm and cancel/return buttons have also been included and mapped to the corresponding standard remote keys. In this interface, buttons are statically displayed and is the user the one that navigates using them.
- A scanning interface, which is specifically targeted to people with movement disabilities that make precise hand/arm movements difficult or impossible. In this case, every action and service can be accessed using a single button in the remote. This is achieved by moving buttons that rotate around the screen, being only one selected button at any time. The user waits for the button he or she wants to activate to have the focus. When this interface is operated through a movement-sensing device (e.g., Nintendo WiiMote), movements of the remote are mapped to instructions to move the focus of the active selection at the screen.
- Pointing interface: This interface is used in combination with the two interfaces discussed above, and is supported by the Nintendo WiiMote. It takes advantage of the pointer included in the WiiMote, which allows the selection of elements on the screen just by pointing at them.

Other options are scheduled to be implemented in the near future, as color-intuitive interfaces based on colored physical switches, eye tracking interfaces, or scanning-aided interfaces.

 ISBN: 978-1-61208-184-7

V. DISCUSSION

Elder and disabled people's functional limitations often entail, among others, limited mobility, poverty, or inadequate medical treatment related to over- or under-subscription. The solution discussed above contributes to address this situation in a cost-effective way. First, it helps to overcome limited mobility by providing communication tools with both family and caretakers, and users' social environment. Besides, providing tools to supervise treatment or even to monitor users' health status facilitates the health control of this population sector. Note that the service architecture described above enables the integration of existing remote monitoring solutions or online services.

However, there are some challenges to overcome to guarantee the deployment of this solution. These challenges are technological, organizational and social. Connected products and services targeted to disabled or elder people proliferate but, apart from basic industry communication standards (e.g. Internet protocols, USB, Bluetooth, etc.), there are not standard interconnection higher-level procedures to enable the interaction among these systems and products. The system described above proposes a basic communication model based on application-layer standards, but the integration of specific services or product may require the development of the corresponding wrapper or adaptation layer.

On the other side, to guarantee service availability and a reasonable level of functional quality, supporting personnel with different profiles is needed. Part of the roles intended for system administrators or administrators of specific products and services may be provided by technical personnel in service providers or even the social environment of users, but in some cases specific personnel may be required (e.g. a health professional to monitor the intake of prescription medicines for users of a remote medication control system).

Finally, the introduction of these service platforms should not imply the further isolation at home or at nursing homes of users. These solutions may be seen as convenient technical approaches to replace the care of humans, but its ultimate function is to complement existing human-based care solutions by contributing to minimize the drawbacks of this kind of care (e.g. limited availability, human errors, fatigue, etc.)

VI. CONCLUSION

We have outlined the basic structure and organization of a socio-sanitary service platform that uses the TV set as the basic interaction element. We have introduced its architecture, and discussed its basic operation.

Tele-assistance is a very active market. In a scenario where new projects and products proliferate, the platform discussed in this paper presents relevant innovative features. First, it successfully integrates the traditional TV experience with interactive services and Internet. Besides, it is a product that can be classified as real low-cost requiring under 300 € investment per home. On the other hand, the platform has been designed to facilitate its market introduction, since the only requirements needed to operate at home are a standard TV set and an Internet connection. These aspects contribute to its acceptance by potential users. An original aspect that differences the developed system from other similar ones is its architecture. The hardware configuration has a clear modular structure, which favors its simple extension using a broad range of external devices. This structure also enables the system to be completely customizable and re-usable. Indeed, both services and interfaces are automatically configured according to the profile of the user accessing the platform. Another relevant characteristic of this platform is its original approach to service provision, as the solution has been conceived as a base system on top of which diverse types of services can be easily deployed. The base system may be extended with new services and new interfaces specifically designed to fulfill specific user needs.

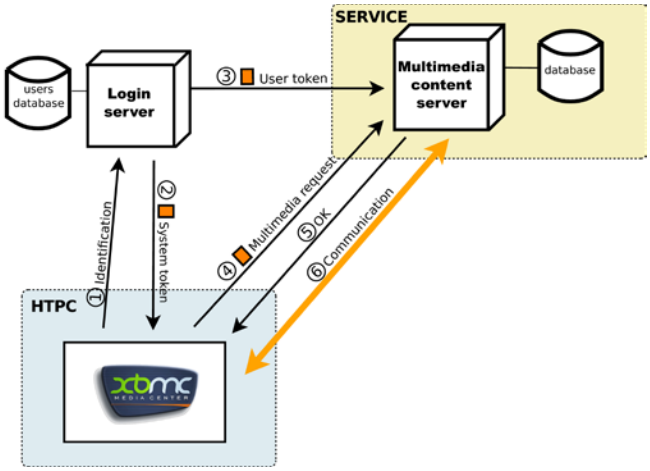


Figure 4. External server communication. Multimedia.

Apart from this, the developed platform uses open standards and widely available devices and software components. To sum up, with this system we show that it is possible to bring new technologies closer to people that are not used to them, and also to utilize new technologies to improve the quality of life of people that is usually disconnected from the rest of the society.

ACKNOWLEDGMENT

The authors wish to thank the support of the Spanish Ministry of Science and Innovation for its partial support to this work under grant "Methodologies, Architectures and Standards for adaptive and accessible e-learning (Adapt2Learn)" (TIN2010-21735-C02-01), and Xunta de Galicia for its partial support through the grant "ACETIC : ACio en Enxeñería de Tecnoloxías da Información e as Comunicacións" included in the "Programa de Consolidación e Estructuración de Unidades de Investigación Competitivas do SUG 2009".

REFERENCES

[1] United Nations, World Population Ageing: 1950–2050. Department of Economic and Social Affairs. Population

 ISBN: 978-1-61208-184-7

Division. United Nations Publications, ST/ESA/SER.A/207, 2001.

[2] JM. Aguilar, J. Cantos, G. Expósito, and PJ. Gómez, “Tele-assistance services to improve the quality of life for lederly patients and their relatives: the Tele-CARE approach”, in Journal on Information Technology in Healthcare, vol 2(2), pp. 109–117, 2004.

[3] D. Spinellis, “The information furnace: consolidated home control”, in Personal Ubiquitous Comput. vol 7(1) pp. 53–69, May 2003).

[4] C. Wingrave, B. Williamson, P. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. LaViola Jr, “The Wiimote and beyond: spatially convenient devices for 3D user interfaces”, in IEEE Computer Graphics vol 30(2), pp. 71–85, March, 2010.

[5] P. Lukowicz, O. Amft, D. Roggen, and J. Cheng, “On body sensing: from gesture-based input to activity-driven interaction”, in IEEE Computer, vol 43(10), pp. 92–96, October 2010.

[6] R. Peng, HY. Yu, and P. Zhang, “Software design of high definition set top box based on embedded linux”, in Dianshi Jishu (Video Engineering), vol. 34(9), pp. 52–56. Sep 2010.

[7] M. Ludz, Programming Python, 4th edition, Sebastopol, CA: O’Reilly Media Inc, 2010.

[8] P. Saint-Andre, Extensible Messaging and Presence Protocol (XMPP): Core, RFC 6120, Internet Engineering Task Force, 2011.

[9] F. Karamitroglou, “A proposed set of subtitling standards in Europe”, in Translation Journal, vol. 2(2), pp. 1–10, 1998.

 ISBN: 978-1-61208-184-7

The Pervasive Fridge
A Smart Computer System Against Uneaten Food Loss

José Rouillard
LIFL Laboratory – University of Lille1
59655 Villeneuve d'Ascq Cedex - France
jose.rouillard@univ-lille1.fr

***Abstract* — Food waste or food loss is food that is discarded or lost uneaten. The work presented in this paper is related to our researches in the field of pervasive and ubiquitous computing. Our "Pervasive Fridge" prototype allows users to be notified proactively, when a food arrives to its expiration date. Speech and image recognition are also integrated in our prototype. This system combines various resources in order to scan barcode, identify and store data related to products, with a smartphone. Later, notifications are sent freely to consumers by mail, SMS (with no charge) and pop-up, to avoid uneaten food loss.**

***Keywords*-Application-oriented system; pervasive computing; ubiquitous computing; ambient intelligence; fridge; barcode scanner; voice interaction; SMS; android.**

I. INTRODUCTION

Food waste or food loss is food that is discarded or lost uneaten. Currently, in the world, according to the Food and Agriculture Organization of the United Nations (FAO), consumers waste about 1.3 billion tons of food annually. Consumers in rich countries waste about 222 million tons of food products [1]. People buy foods that are kept in fridge or cupboard and when products arrive at their deadline, they are thrown.

The work presented in this paper is related to our researches in the field of pervasive and ubiquitous computing. We are particularly trying to find some ways to help consumers avoid wasting food. Nowadays, the consumers, generally less attentive than by the past, often do not interpret correctly the dates of consumption, and/or do not care about the organization of the refrigerator and so, do not have the time to manage their products. Obviously, this leads to important food losses. Our goal is to reduce this waste of food. As this is mainly due to problems of "memory", we decided to work on solutions that provide easy and usable ways of doing this reminder's task, thanks to new technologies. We have decided to tackle two main parts of this problem by providing: (a) multiple ways to enter into the system data concerning the products, and (b) various notifications in order to remind important deadlines to users.

We believe that smartphones can be efficiently used in this fight against food waste. Indeed, they are natively equipped with hardware (camera, microphone) and software (barcode reader, automatic speech recognition, etc.) that can be combined and easily employed in order to collect data and also remind important deadlines to consumers.

The document is structured as follows: background and motivation of this project are explained in section two. Section three presents the related work on that domain. Section four gives an overview of our approach in order to tackle the emerging problems encountered. Section five describes a case study and our results using our developed prototype. Then, we conclude this paper and give our roadmap for future work.

II. BACKGROUND AND MOTIVATION

One quarter of the food produced on an international scale is discarded without being consumed, while more than 800 million people suffer from hunger in the world. To the question "*Why does so much food that could have been eaten get thrown away?*", the "Love food, Hate Waste" web site answers : "The main reasons for throwing away food can be grouped in to "cooking or preparing too much" (for example cooking too much rice or pasta and it gets left in the saucepan or on the plate) or "not using food in time" - for example having to throw out fruit and vegetables because they have gone off in the fruit bowl or in the fridge, or not eating food before it goes past its use-by date. We know that there are lots of potential reasons why food might not get eaten in time – our plans change, we forget what food we have in the cupboards, we forget to freeze or chill something to use at a later date, we lack the confidence or knowledge on how to use up our leftovers – which is where our website can help!" [2].

Loss and wastage occurs on all steps in the food supply chain. In low-income countries most loss occurs during production, while in developed countries much food – about 100 kilograms per person and year – is wasted at the consumption stage.

Our research interests include human-computer interaction in the context of ambient intelligence. The main goal of our research is to find a way to model, design and implement computer systems to facilitate human activities.

 ISBN: 978-1-61208-184-7

Currently, our purpose is to cross these aspects with the notion of "green IT", allowing the user/consumer to obtain relevant information in order to take adapted decisions.

In this context, our motivation for this work is to look for solutions that can help consumers in avoiding the loss of uneaten food thanks to new technologies and smart devices. Thus, our research area is here clearly related to the field of application-oriented systems.

III. RELATED WORK

Trying to reduce the food waste is not a new idea. In the world, there are many applications that allow users to manage their food. In this section, we give some information about existing systems oriented toward this objective. We analyze the capabilities of each system, encountered during our study of the literature. We have listed below some characteristics of those applications.

"**Consume Within**" (Iphone, 2.99$) is an application that helps households to reduce the amount of food they waste. Food items can be added to three different locations – fridge, freezer or cupboard – and their expiration dates set individually. Each entry can be accompanied by a photograph to identify when removed from storage. The application alerts users daily of the foods that are about to expire within the next 3 days and displays them by location or as a single list [3]. In the same way, "**Food Reminder"** (Iphone, free) keeps tracks of food expirations and provides reminders [4]. **"FoodScanner"** (iPhone, 4.99 $) allows using the camera to scan UPC barcodes on foods in order to track how many calories are eaten throughout the day by users [5]. By using "**Fridge Police**" (iPhone, 2.99 $), it is possible to obtain, after a barcode scan, some information about the products (i.e., manufacturer, brand, size, etc.) that will be logged along with the "opened on" date. The user is also prompted to take a quick photo of the item. Then, the item is placed in an alarm calendar, which can be accessed at anytime. As the items "use by" dates approach, users will get an automatic reminder [6]. "**Rz Fridge Reminder**" (Windows Phone 7, free) is an application available in English and Italian and could manage expire date of items stored in the fridge. Alerts are sent upon users choices (expire tomorrow, next week, next month, etc.) [7]. "**Fridge Manager**" is a Korean application (Android, free), that helps users in managing goods into their fridge with an alarm service [8]. With "**Fridge Friend**" (Android, 0.99 $), consumers can organize the groceries in their refrigerator and see which ones need to be used first. It is also possible to export data in a CSV (Comma Separated Value) file, which can be emailed to a particular destination [9]. "**Fridgician**" (Android, 0.99 $) is the same kind of inventory tracking system for home refrigerator. It provides features such as multiple accounts on the same phone, or view food sorted by date, expiration, and name [10]. "**LG smart Refrigerator**" (Android, free) is an application that works with LG DIOS Smart Refrigerators. The application connected to the world's first Smart Refrigerator using wifi provides convenience food management. Some particular features related to this refrigerator are available, such as management food list, location management and marking the expiration date, remote checking system of food list, recommended recipes, shopping list checking and monitoring conditions of the refrigerator [11]. This monitoring feature is often used to qualify intelligent refrigerators: "*An intelligent refrigerator is one which possesses self monitoring capability of food item or material with minimal or no human intervention.*" [12].

Other smart systems are now embedded into refrigerators. For example, the "**MMS Fridge**" by Electrolux, the "**Samsung's RF4289 WiFi smart fridge**" (3499 $), which can be connected to Twitter, Pandora, Gmail, Epicurious, etc. and the South Korea's **"LG smart fridge**" (4124 $), that suggest recipes, are available for consumers interested in those technologies.

Numerous studies were conducted around the notion of smart fridge. Some researchers are focusing on the ability to enhance health and enable better nutrition [13] while some others prefer focusing on the communication matters between the user and the device [14]. Among the main shortages of current smart fridges systems, listed by [13], we particularly agree with the following: "*The technology is too complex for most household users, needing more user-friendly interface for general users who have little or no experiences of using computer.*" and "*There is no uniformed bar code to record information such as expire date of the food.*"

Moreover, concerning the notification about expired goods, we noticed that all the systems that we have studied are conceived on a reactive mode. When the user asks the system for expired dates, it replies with a list of products. Obviously, if the user does not ask, he/she will never be notified. We are studying other ways to inform users about important events, and we are deeply convinced that smart systems have to be modelized on a proactive mode.

It is the system itself that has to push the relevant information to the users, by various ways, and not the opposite. But according to Mike Kuniavsky, "*Reviewing the history of the commercial fridge computer demonstrates both the idea's tenacity and its lack of commercial success.*" [15]. Indeed, it is not necessarily a good idea to propose always more capabilities if they do not match the user's needs. For instance, "*The intelligent fridge ordering food (M = 3.63, SD = 1.19) and the intelligent TV ordering products (M = 3.63, SD = 1.07) were found to be the least attractive.*" in a study presented in [16].

 ISBN: 978-1-61208-184-7

Figure 1: Scanning the barcode and obtain information about the product with Pervasive Fridge

IV. OUR APPROACH

The main subject of this work is the modeling and implementation of a timely reminder of deadlines and validity of food stored in the fridge or cupboards. To avoid waste, we want to have a ("green") computer system to notify a smart home of the input and output of products into the fridge, via barcode, voice recognition or RFID tag, for instance. The system will also manage other aspects such as suggestion of recipes, depending on the ingredients available, allergic alerts, dietary advices, assistance to improve the conservation of food, etc.

One important difference between our system and the ones discussed in the related work part of this paper is that the database used by our system is managed externally (see Prixing) and takes advantage of a large community of consumers, that can add and comment a large set of products.

Our primary objective is to provide a useful, usable and low-cost system. We think that it is a good idea to use the capabilities of smartphones already daily manipulated by the consumers. Indeed, modern mobile phones and smartphones are equipped with barcode scanner, automatic speech recognition systems, SMS and instant message capabilities and could access the Internet network. So, we argue that the proactive mode of interaction that is predominant in our approach could be reached by the use of a pervasive system, in which the smartphone will be the core, for entering information and receiving notifications.

A. *Entering data about the product*

Our project name code is "Pervasive Fridge". It refers to the paradigm of pervasive computing and ambient intelligence, but does not mean that we are working on an intelligent fridge. The fridge of the users remains the same (i.e., classic, and not connected to Internet, without barcode reader, etc.). It is the application that we provide for smartphones that leads to the notion of pervasive and ubiquitous computing. Our system can be used in the supermarket, during the shopping or, at home, during the storage of the products in cupboards and fridge. The entering data task consists in giving to the system, for each food, relevant information for their future management. Thus, name, location, category, number of items, expiration date, and some optional data are asked to the user for each product and saved in a database.

This manipulation can be done, as traditionally, with the classic text/keyboard manner, but also with some more convenient ways with barcode/scanner, and voice/microphone of the mobile device. Figure 1 presents some screen captures of our prototype: (a) the user actives the camera of the device by a simple touch on the "scan" button, (b) the barcode of a product is scanned and detected, (c) the corresponding product is searched on an external database, available on Internet. Once the product is identified, the related fields of the form are automatically filled and the user enters the number of items, the category and the deadline (i.e., the expiration date). This could be done textually or vocally. On Android, the Google voice tools are used to do so.

 ISBN: 978-1-61208-184-7

One of the key problems in such project is to choose between internal or external tools, in order to retrieve information based upon barcode data. EAN-Search [17] and UPCDatabase [18] are available on Internet, but they do not provide API (Application Programming Interface). So we decided to choose Prixing [19] as an external database. Prixing is a powerful system that retrieves the price of a product (not only food) according to its EAN code/barcode.

The API of Prixing is freely usable for developers and currently limited to 500 requests per day, per account. The database contains around 180.000 food references among 2.000.000 products. More than 700.000 products have been added in seven months by the people of the Prixing community; it represents almost 4000 products added each day.

In Pervasive Fridge, the deadline of the product can be entered by different means: directly by text or voice, or indirectly, with an estimated date, depending on the category of the product.

Figure 2: Identifying product by a photo

As illustrated on Figure 2, with Pervasive Fridge, it is possible to identify a product by taking a photo of it. This feature is helpful in situations where foods are not marked with barcodes (fruits & vegetables, some bread, etc.).

B. Notifying users about deadlines

As we can see classically on others applications, with Pervasive Fridge it is possible to display, on request, the list of the stored products and to mention the remaining days before expiration.

For example, we can see on Figure 3 that the Coca-Cola is still consumable within the next 9 days. The red color is used to indicated the expired foods (“original Collection” – Haagen Dasz – and “Ice cream smoothie” in our example) while green or orange colors indicate respectively products that are still eatable and products that have to be eaten very soon (see the colors used in Figure 4) .

An internal reminder is also activated each day at a parameterized hour, while a vibration is activated on the device, and a pop-up invites the user to see what products are expired.

Figure 3: List of managed products

Moreover, we decided to propose a more proactive notification that can be pushed to the user, even if his/her mobile device is not active. This is done thanks to reminders programmable via Google Calendar Agenda, as illustrated on Figure 4. Therefore, pop-up, Email, but also SMS notifications can be used with Pervasive Fridge, in order to push to the user important food deadline messages, every time it is necessary, everywhere on the planet, on a multichannel manner.

Figure 4: Google Agenda notification used with pervasive fridge

 ISBN: 978-1-61208-184-7

V. RESULTS

In order to test and evaluate the capabilities of our system, we used Pervasive Fridge within the following scenario. Thirteen different products were bought in a supermarket.

These products were composed of foods and beverages. We tried to use barcode, voice, photo and text to identify products and to enter expirations dates. Thirteen products have been bought the 21th of September 2011. Here is the list of these goods, sorted by expiration deadline: a piece of pork (23/09/2011), a Pack choy vegetable (25/09/2011), a pack of Kiwis (27/09/2011), a Chinese cabbage, some tomatoes and bananas (28/09/2011), 4 chocolate creams (29/09/2011), 6 eggs (13/10/2011), a bottle of milk (17/12/2011), a large bottle of Coca-Cola (30/12/2011), a pack of beer (25/10/2012), a bottle of mineral water (01/03/2013) and a rice pack (06/08/2013). Five products have been successfully identified by scanning their barcode and thanks to the Prixing database: the bottles of water and milk, the eggs, the beers and the rice. One product has been successfully identified by pronouncing its barcode: the Coca-Cola. This operation was a little bit longer because it worked after a few tries and by pronouncing the barcode numbers, three by three such as, for instance: 317 – 478 – 000 – 043 – 1.

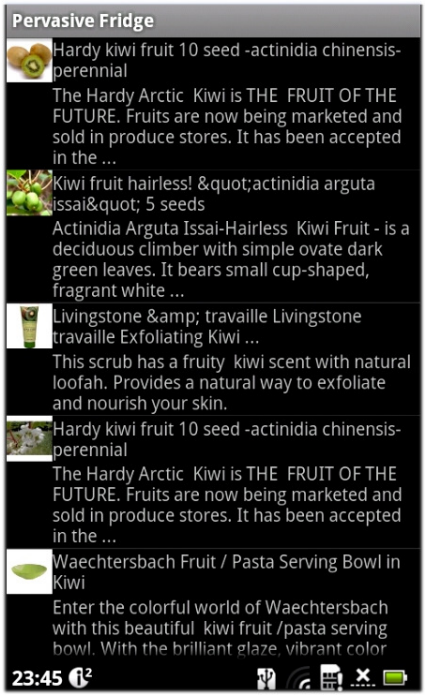


Figure 5: Kiwis, identified by a photo of the product

One product has been successfully identified by taking a picture of it: the pack of kiwis, as illustrated in Figure 5. The image recognition took a long time (around 40 seconds) compared to the barcode scanner method (around 2 seconds). After this image recognition phase, the system proposes a list of possible matching products and the user chooses among them the more representative, compared to the targeted one. The field “barcode of the product” is not filled, but the other fields of the form such as the name of the product, the category or the estimated relative deadline are automatically filled, if available. The six others products have been entered in the system by typing their barcode with the virtual keyboard of the Android device.

In the following lines, we give some information about the reactive and proactive received notifications in this experiment. The reactive notification way is traditional in this kind of applications. It means that the user ask voluntarily the system about the expirations date of the food stored in the system. In reply, the application indicates to the user, with a graphical internal reminder message (see Figure 6) and a vibration of the device.

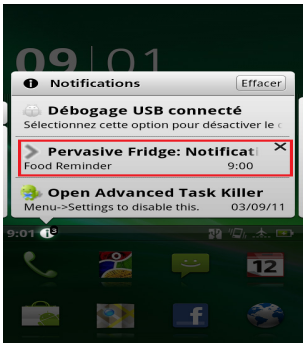


Figure 6: Internal reminder of Pervasive Fridge

For instance, in our tests, the 22th September of 2011, the Pervasive Fridge indicated two products to be eaten soon: the piece of pork (23/09/2011) and the Pack choy vegetable (25/09/2011). We see with this example that the system uses a notification parameter of 3 days for the vegetable category. The user can freely change all the notification parameters, and this will indicate to the system how many days before the expiration date he/she like to be notified.

Figure 7 is an example of proactive notification event created automatically by our system in the consumer's Google Calendar. We used the Google Calendar Agenda API to indicate the channel (window pop-up, Email and/or SMS reminder) and the timing chosen by the user.

 ISBN: 978-1-61208-184-7

Figure 7: A Google Agenda reminder event created by Pervasive Fridge

Thus, we provided the possibility to push some important information to the user, even if the Pervasive Fridge application is not activated at the appropriate moment. Finally, Figure 8 presents a screen capture of a SMS notification received freely by the user on his/her phone. We strongly believe that these kind of proactive messages, emanating from our Pervasive Fridge system, are very useful in the fight against uneaten food loss.

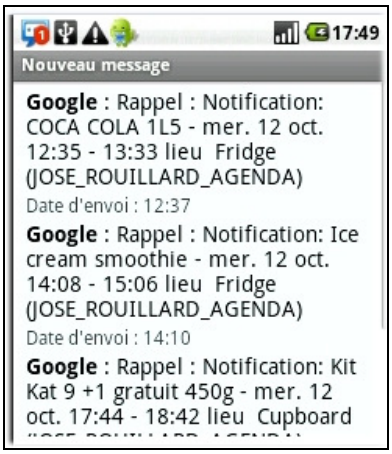


Figure 8: SMS Notification for Coca-Cola, Ice Cream and Kit Kat

VI. CONCLUSION AND FUTURE WORK

The goal of this paper was to describe the Pervasive Fridge system, and to explain how we can facilitate the fight against uneaten food loss by using ambient computing strategies and technologies. Our results shown that a proactive system is technically feasible, and that it can freely and easily push appropriate reminders to the consumers. Pervasive Fridge is an application-oriented system that can be used in a multimodal way (keyboard or voice) and with different technologies (camera barcode scanner, voice and image recognition). We tested our system with an experiment. It showed that Pervasive Fridge helped in (a) entering products information into the database, and (b) receiving relevant reminders across multiple channels of interaction. Our future work will be oriented toward the possibility to propose recipes according to the foods managed by our system. We are envisaging to propose relevant manners (with augmented reality for instance) to measure the necessary quantities of food to buy and cook, in order to avoid loss and wastes of uneaten food.

ACKNOWLEDGEMENT

The authors would like to thank Truong Vu Duy for his help in the prototype development, Big5media and CPER CIA (Contrat Plan Etat Région, Campus Intelligence Ambiante) for the financial support of this project and Prixing and IQ Engines Image Recognition for the API provided.

REFERENCES

[1] Global food losses and food waste, Düsseldorf, Interpack2011, retrieved on 12/23/11, http://www.fao.org/fileadmin/user_upload/ags/publications/GFL_web.pdf

[2] Love food hate waste, retrieved on 12/23/11, http://www.lovefoodhatewaste.com/about_food_waste

[3] Cusume Within, retrieved on 12/23/11, http://www.consumewithin.com

[4] Food Reminder, retrieved on 12/23/11, http://itunes.apple.com/us/app/food-reminder/id430227307

[5] FoodScanner, retrieved on 12/23/11, http://dailyburn.com/foodscanner

[6] Fridge Police, retrieved on 12/23/11, http://fridgepolice.com

[7] RZ fridge reminder for wp7, retrieved on 12/23/11, http://www.windowsphoneapplist.com/rz_fridge_reminder-a62.html

[8] Fridge Manager, retrieved on 12/23/11, http://www.appbrain.com/app/fridgemanager/com.tacademy.fridgemanager

[9] Fridge Friend, retrieved on 12/23/11, http://www.androidzoom.com/android_applications/shopping/fridgefriend_xrzf.html

[10] Fridgician, retrieved on 12/23/11, http://www.androidzoom.com/android_applications/productivity/fridgician_vkoz.html

[11] LG Smart Refrigerator, retrieved on 12/23/11, http://www.androidzoom.com/android_applications/tools/lg-smartrefrigerator_xltb.html

[12] Gangadhar, G., Subramanya N., and Puttamadappa C., Intelligent Refrigerator with Monitoring Capability through Internet. IJCA Special Issue on Wireless Information Networks & Business Information System (2):65–68, 2011. Published by Foundation of Computer Science.

[13] Suhuai Luo, Jesse S. Jin, and Jiaming Li "A Smart Fridge with an Ability to Enhance Health and Enable Better Nutrition" The University of Newcastle, Australia, CSIRO ICT Centre, Australia, International Journal of Multimedia and Ubiquitous Engineering, Vol. 4, No. 2, Avril, 2009.

[14] Samuli Pekkola, Yu You, and Mike Robinson "Telephone mobiles, Refrigerators, Bar Code Readers, Cameras, The Web and People" 23rd Conference on Information Systems Research In Scandinavia (IRIS'23), Uddevalla, Sweden, 2000.

[15] Kuniavsky Mike, Smart Things: Ubiquitous Computing User Experience Design, Elsevier, 318 pages, 2010.

[16] Ben Allouch, S., The design and anticipated adoption of ambient intelligence in the home. Doctoral dissertation, University of Twente, The Netherlands, 169 pages, 2008.

[17] EAN Search, retrieved on 12/23/11, http://www.ean-search.org

[18] UPCDatabase, retrieved on 12/23/11, http://www.upcdatabase.com

[19] Prixing, retrieved on 12/23/11, http://www.prixing.fr

 ISBN: 978-1-61208-184-7

A Graphical Development Tool for Earth System Model Using Component Description Language

Chao Tan, Sujun Cheng, Zhongzhi Luan, Si Ye, Wenjun Li, Depei Qian
Sino-German Joint Software Institute, Beijing Key Laboratory of Network Technology
Beihang University
Beijing, China
E-mail: tanchao128 @126.com

***Abstract*—Historically, researchers have developed large-scale Earth System Model (ESM) applications under the monolithic software development practices that seriously hamper further innovation in complex, highly integrated simulations. Earth System Modeling Framework (ESMF) which organizes applications as discrete components is built to achieve the loose coupling of model and component reuse. Yet it is a big challenge for earth researchers to develop, maintain and share the component code due to the absence of the system software development training and Integrated Development Environments for ESM. To overcome these disadvantages, in this paper we propose an Earth System Model Component Description Language (ESMCDL) which describes not only the interface of component but also the inner behavior of the interface. At the same time, based on this language a graphical development tool is designed to help researchers encapsulate, release components and build templates which consist of these components. Results show that the tool based on the ESMCDL significantly reduces the time required to develop models.**

Keywords-component description language; Earth System Model(ESM); ESM Framework.

I. INTRODUCTION

With the development of Earth System Science, the scale of the software systems for Earth System Model (ESM) becomes increasingly huge. In addition, under monolithic software-development practices, the structure is also becoming more and more complex and there exist a large number of reusable modules as well as their combinations. For example, Community Earth System Model [1], which has nearly 1 million lines of source code, is a coupled climate model for simulating Earth's climate system and composed of one central coupler component and five separate models that simultaneously simulate the Earth's atmosphere, ocean, land, land-ice, and sea-ice, what's more, each model is composed of a serial of physical processes and calculation processes.

NASA with other research institutions has built standards-based open-source software -- Earth System Modeling Framework (ESMF) [2], which defines a component architecture and aims to reduce the coupling between each module and increase Earth System Modeling software reusability.

However, on one hand, researchers who generally lack the system software development training have to consider about plenty of the essential framework code not related with the business logic; on the other hand, there is no unified platform which helps researchers from different institutions share component code. What's more, at present, no matter from general or professional perspective, there is still no widely adopted Integrated Development Environments (IDE) [3] [4] for ESM. Therefore, developing, maintaining and sharing the component code put extra pressure on researchers, which limit the widespread use of ESMF.

In order to solve these problems, this paper firstly introduces the Earth System Model Component Description Language (ESMCDL). The research on component description language can be traced back to the 1980's. The most representative works includes the OBJ [5] and LIL [6] developed by Gougen. In the 1990's, most efforts were spent on how to enable CDLs to describe component, as well as component sub-system. Main works include CIDER [7], RESOLVE [8], and etc. [9] presents a visual Coupling Description Language, but it only focus on hydrology domain. Our ESMCDL, as a kind of metadata describing language, describes not only the interface of the component defined in the ESMF but also the inner behavior of the interface. Moreover, based on this language we develop a graphical development tool which helps researchers not only encapsulate existing modules to form the general earth system component library through the ESMF and parallel technology, but also analyze and summarize the common combinations of components to form the general template library.

This ESMCDL helps the tool with the following functions to achieve the support of rapid development of software.

1) Interface Reuse: ESMCDL separates component code from the abstract definition layer and the specific implementation layer, like the concept of generality. Based on the identical abstract definition, different researchers can adopt different logic realizations.

2) Code Generating: one *.esmcdl file, the context of which follows the ESMCDL format, can be automatically mapped to FORTRAN source code by the tool we provide.

 ISBN: 978-1-61208-184-7

3) Specification Definition: ESMCDL defines one unified programming standard, based on which researchers will realize the encapsulation and release of all components. It can favor code's maintenance and sharing.

4) Template verification: templates could be verified by validating the links between components at the beginning of the design.

The remainder of the paper is organized as follows. Section II gives a short introduction to the ESMF. The syntax structure of the ESMCDL is presented in Section III. Section IV introduces an ESMCDL-based graphical development tool. In Section V, we demonstrate how the tool assist the researchers in developing ESM components as well as templates with a case study of Parallel Ocean Program (POP) [10] and give the preliminary result of it. Finally, we conclude with Section VI.

II. ESMF OVERVIEW

As illustrated in Figure 1, the ESMF comprises a superstructure and an infrastructure. The role of the superstructure layer is to provide a shell that encompasses user code and a context for interconnecting input and output data streams between components. Classes called gridded components, coupler components, and states are used in the superstructure layer to achieve this objective [11]. The infrastructure layer provides a standard support library that researchers can use to speed up construction of components and ensure consistent, guaranteed component behavior. It contains a set of data classes such as Array, Field, and utility classes (including Time, Config). This paper focuses on description for all superstructure classes and data classes.

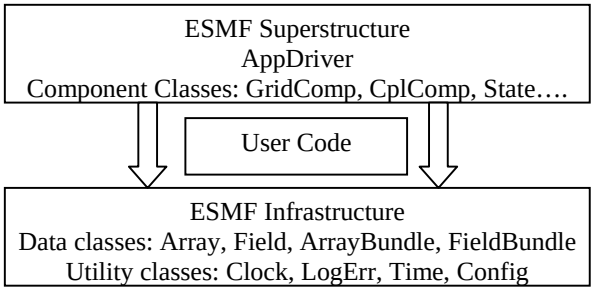


Figure 1. Architecture of the ESMF

Following the principles of loose coupling, we divide the code for each component into Registration, Init, Run, and Final methods as follows:

1) Registration: components register all INF (Init, Run, and Final) methods which can be multi-phase and create all sub-components and States.

2) Init: components allocate the key data structures used to pass persistent states in and out. Coupler components allocate ESMF_RouteHandl Objects.

3) Run: during the run phase, data should be accepted at the beginning of the method and updated after computing in the gridded components. Complex type data transfer occurs by ESMF_RouteHandl Objects in the Coupler components.

4) Final: applications shut down components cleanly in the Final method. For example, Gridded Components destroy sub-components, States and Complex type data, Coupler components destroy ESMF_RouteHandl Object.

Particular emphasis is given to the Sub-component's INF methods, which will be called recursively in each INF method of Gridded Component.

III. THE ESMCDL

In this section we will make rules for the coding behavior of users before introducing the ESMCDL, which simplify the complexity and is also beneficial to the realization of language engine.

1) All the Complex data such as Array, Field, ArrayBundle and FieldBundle should be created in the Init method of which the phase is 1 and destroyed in the Final method.

2) Each State object created in the parent Gridded Component is associated with only one child Gridded Component. It means if one Import State object is passed to any method of one child Gridded Component as parameter, another child Gridded Component couldn't take in it as input.

3) The name of Registration method adopts uniform format "*_setServices", which can be identified by tool.

4) State object (import or export) can only be added into the same type of State object.

5) A Gridded Component contains one Coupler Component at most.

A. Gridded Component

We adopt XML instead of other textual, graphical or binary formats to define ESMCDL as meta–language, since it has many advantages such as scalability, platform-independence and readability. Figure 2 shows the structure of the Gridded Component Description Language.

- **GridComp** denotes an ESMF_GridComp Object. Each GridComp element has *name*, *location* and *variable* property.
- **CplComp** denotes an ESMF_CplComp Object. Each CplComp element has *name*, *location* and *variable* attributes.
- **State** denotes an ESMF_State Object. Parent Gridded Component needs to assign Import State and Export State to its child components. Each State element has a *name*, *variable* and *type* attributes. Type is ESMF_STATE_IMPORT or ESMF_STATE_EXPORT.
- **Init** denotes Gridded Component registers an ESMF_SETINIT type of method in the *_SetSevice subroutine. It consists of four subelements: InitializeComp, StateAdd and AttributeGet, and AttributeSet, whose order of appearance is the same as the order of being called

 ISBN: 978-1-61208-184-7

in the source code. InitializeComp indicates that this Gridded Component will initialize its subcomponents. StateAdd and AttributeSet indicate that the Gridded Component creates data and adds them into state object.

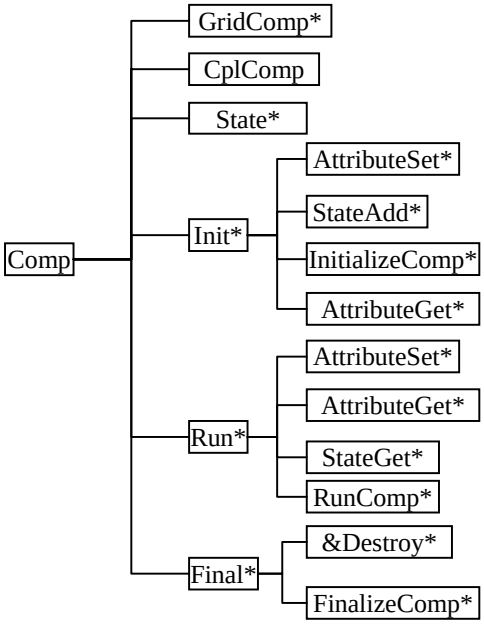


Figure 2. Core Structure of the Gridded Component Description Language

- **Run** denotes Gridded Component registers an ESMF_SETRUN type of method in the *_SetSevice subroutine. It consists of four subelements: RunComp, StateGet, AttributeGet and AttributeSet. RunComp. RunComp indicates that this Gridded Component will run its subcomponent. StateGet and AttributeGet indicate that the Gridded Component needs the data which is necessary to execute.
- **Final** denotes Gridded Component registers an ESMF_SETFINAL type of method in the *_SetSevice subroutine. &∈ {ESMF_GridComp, ESMF_State, ESMF_Array, ESMF_ArrayBundle, ESMF_Field, ESMF_FieldBundle}.

B. Coupler Component

A Coupler Component (or ESMF_CplComp) arranges and executes the data transformation between the Gridded Components. It takes in one or more import ESMF states as input and maps them through spatial and temporal transformation onto one or more output export ESMF states. The structure of the Coupler Component Description Language is given in Figure 3.

- **AttributeCopy** describes the mapping relationship of metadata between the Gridded Components. Scope property (may be one or all) denotes the scope of exchanging. The sub-elements from and to describe where the attributes are from and which to be copied to.
- **&**: There are ESMF_Array, ESMF_Field, ESMF_ArrayBundle, ESMF_FieldBundle on complex data types that have all versions of the data communication methods: Halo, Redist, Regrid, SMMS. Therefore, & can be any of all the 16 transformation methods
- **RouteHandle** denotes an ESMF_RouteHandle Object which identifies those stored communication patterns mentioned above which can be precomputed during an initialization phase and then later executed repeatedly.

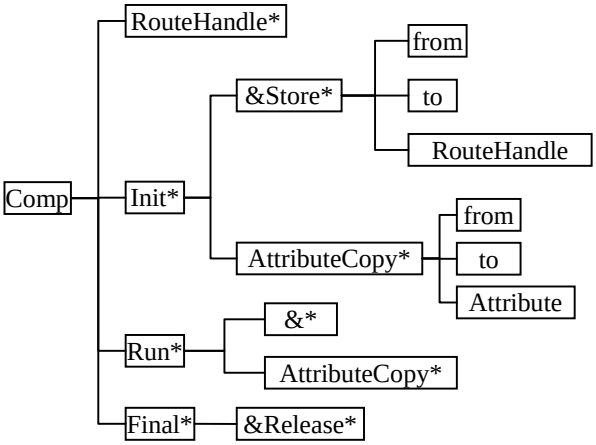


Figure 3. Core Structure of the Coupler Component Description Language

IV. THE GRAPHICAL DEVELOPMENT TOOL

To uniformly build, share components and templates, we created a graphical development tool. This tool is based on ESMCDL which consists of graphical modeling, components publishing and sharing, code generating and data validation. It is built on the Eclipse platform and adopts Photran [12], GMF [13] plug-in technology to strengthen the extensibility. The technical details are beyond the scope of this paper. As shown in Figure 4, the system is composed of four parts.

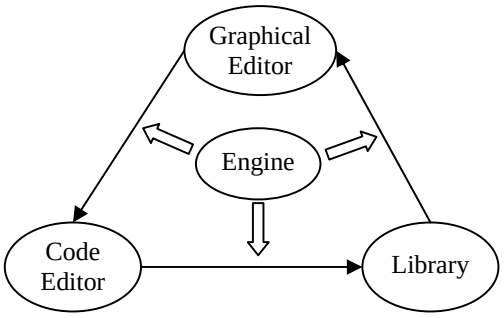


Figure 4. Overview of the graphical tool

 ISBN: 978-1-61208-184-7

1) Graphical editor: application based on the ESMF is executed in a top-down, recursive method mentioned in the section 2. In contrast, with the help of the graphical editor, the application can be built in a bottom-up, iterative method. In the Graphical editor, researchers can do the following things.

*a) Building a Gridded Component (including leaf component and parent component or from a *.esmcdl file directly).*

b) Building a Coupler Component.

c) Building a template: Researchers can drag and drop components which have been in the components library into the graphical editor and then organize them to build the whole template.

2) Code editor: It contains some IDE features such as syntax-highlighting, content assist, code version control and so on.

3) Component and template library: similar to database. It is used to manage ESM templates and components as well as corresponding ESMCDL interface files published by different researchers, including registering, deleting, searching and other functions. The principle that issued once and used many times is followed to provide component providers and consumers with a uniform platform, which make their sharing model code convenient and flexible.

4) ESMCDL engine: ESMCDL needs one corresponding language engine which involves converting ESMCDL documents to Component objects when dragged into the graphical editor or standard FORTRAN code as well as parsing source code of Components to generate ESMCDL files if rule validation is passed.

We present a case study next that ties everything together.

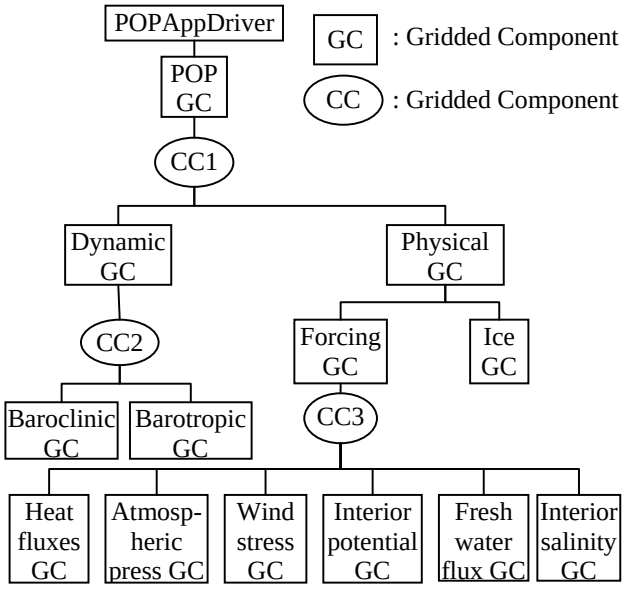


Figure 5. Structure of the components-based POP

V. CASE STUDY

To illustrate the power of our ESMCDL, we now give a complete example, Parallel Ocean Program (POP), one of the most representative models in ESM field. Figure 5 displays the structure of the components-based POP bringing in superstructure and infrastructure for ESMF and following the partitioning strategy mentioned in section 2 as well as programming rules referred in section 3. The Components-based POP is divided into two major parts: Dynamic GC processing data and Physical GC providing date. POPAppDriver is the entry point of the application.

As illustrated in Figure 6 we demonstrate how to create the POP in the bottom-up approach via our graphical software development tool.

First of all, we build those six leaf Gridded Components one by one, including heat fluxes GC, atmospheric pressure GC, wind stress GC, Interior potential GC, fresh water flux GC, interior salinity GC. It includes a series of steps: 1) drag a new component into the **graphical editor** and configure its information such as the name and interface functions. 2) Based on the configuration information, the standard framework code is generated with the help of the **EMCDL engine**. 3) After that, researchers can fill the INF's internal logic code of each child component In the **Code editor**. 4) **EMCDL engine** checks the component code according to the rules to determine whether the code is correct. 5) If no problem, one *.esmcdl file is generated with which every child component will be released to the **component library**. In addition, our tool also supports interface reuse, for instance, based on the same interface different researchers can realize the different calculation processes.

Now those six leaf components have been in our component library. Researchers can drag them into the graphical editor to form the Forcing Gridded Component and then release it into our component library in the same way stated in the previous paragraph. It is necessary to specially emphasize that ESMCDL engine will parse the code of the Component when generating the corresponding *.esmcdl file. We take the forcing Gridded Component and CC1 Coupler Component as an example to explain what the engine analyzes. Figure 7 and 8 illustrate that code is divided into many parts by ESMCDL engine. The code on the left will be mapping to the xml on the right.

Lastly, when all the type components (there may exist different ones based on the same interface) have been prepared in the component library, we can choose some components and drag them into the graphical editor and then link them to build a new POP template or modify an existing POP template saved in template library. Before generating model application code, ESMCDL engine will validate the legitimacy of the link between components recursively from the root node POPAppDriver.

 ISBN: 978-1-61208-184-7

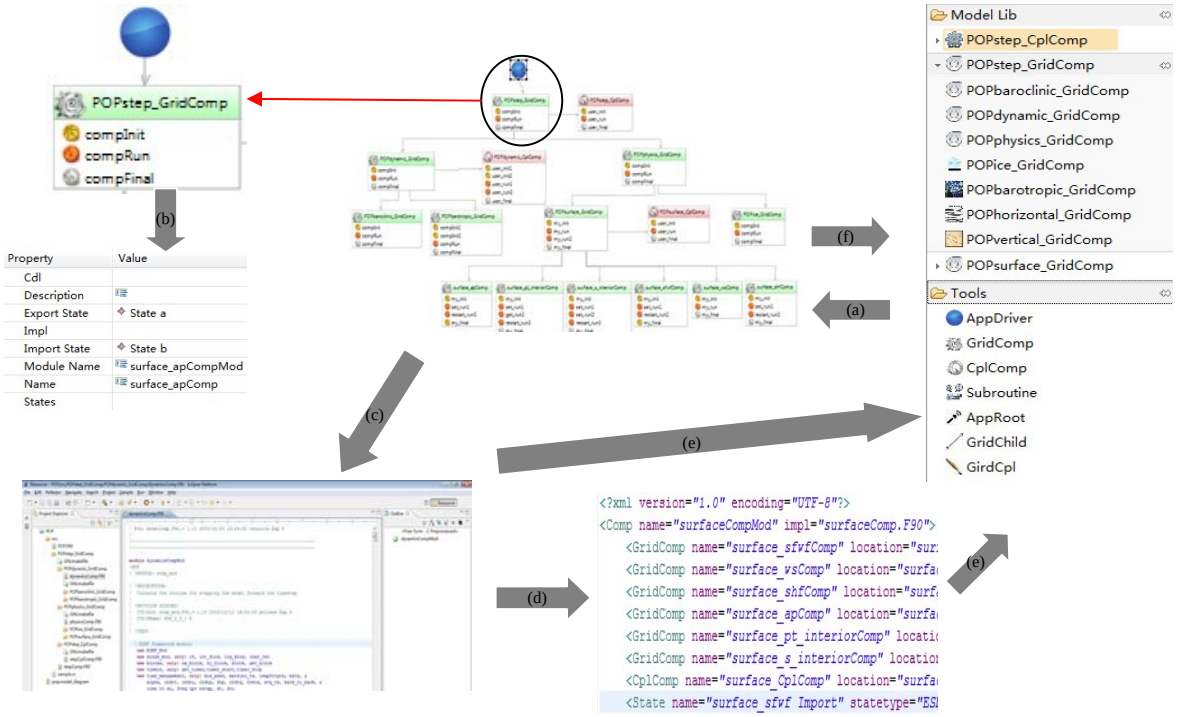


Figure 6. Process of creating one Earth System Model application. (a)Drag and drop components. (b)Configure components' information. (c)Validate the template and generate its code (d)Check the code of the new component and generate its *.esmcdl file. (e)Release the components into the component library (f) Release the template into the templateraray

```
<Comp name="surfaceCompMod"  imp="surfaceComp.F90">
  <GridComp name="sfwf_Comp" location="surface_sfwfCompMod"varaible="sfwf_Comp" />
  ……
  <CplComp name="surfaceCplComp" location="surfaceCplCompMod"variable="surfaceCplComp"/>

  <State name="surface_ws Import" statetype="ESMF_STATE_IMPORT" variable=" wsimp "/>
  <Init name="my_init" phase="1" importState="importState" exportState="exportState">
      <InitializeComp type="Grid "comp="ws_Comp" importState="wsimp" exportState="wsexp" phase="1"/>
      ……
      <StateAdd>
          <state>exportState</state>
          <item type="ESMF_Array" name="FW">FW_array</item>
      </StateAdd>
  </Init>
  <Run name="my_run" phase="1" importState="importState" exportState="exportState">
        <AttributeGet>
          <object>
                exportState
          </object>
          <name>lsmft_avail</name>
          <value type="log_kind">
                lsmft_avail
          </value>
        </AttributeGet>
  <RunComp type="Grid "comp="ws_Comp" importState="wsimp" exportState="wsexp" phase="1"/>
      ……..
  </Run>
  <Final name="my_final" phase="1" importState="importState" exportState="exportState">
      <ArrayDestroy>
          <state>exportState</state>
            <Array name="FW">
            FW_array
          </Array>
      </ArrayDestroy>
      ……
  .</Final>
</Comp>
```

```
//create all child components in register phase
sfwf_Comp=ESMF_GridCompCreate(name="sfwf_name",
grid=parentgrid, rc=rc)

//create all the State objects in register phase
Sfwfimp=ESMF_StateCreate(statetype=
ESMF_STATE_IMPORT, name="surface_sfwf Import")

//add date in to state object in the Init method
call ESMF_StateAdd(exportState, FW_array, rc=rc)

// Register INF method
call ESMF_GridCompSetEntryPoint (gcomp, ESMF_SETRUN,
my  run, rc=rc)

//get date from sfwfexp object
call ESMF_AttributeGet(exportState, name=" lsmft_avail ",
value= lsmft_avail, rc=rc)

//call child component'INF procedures
call ESMF_GridCompRun (ws_Comp, importState=wsimp,
exportState=wsexp, clock=parentclock, rc=rc)

//destroy all date created in Init method
callESMF_StateGet(exportState,"FW",FW_array,rc=rc)
call ESMF_ArrayDestroy(FW_array, rc=rc)
```

Figure 7. Mapping between ESMCDL and code for forcing Gridded Component

 ISBN: 978-1-61208-184-7

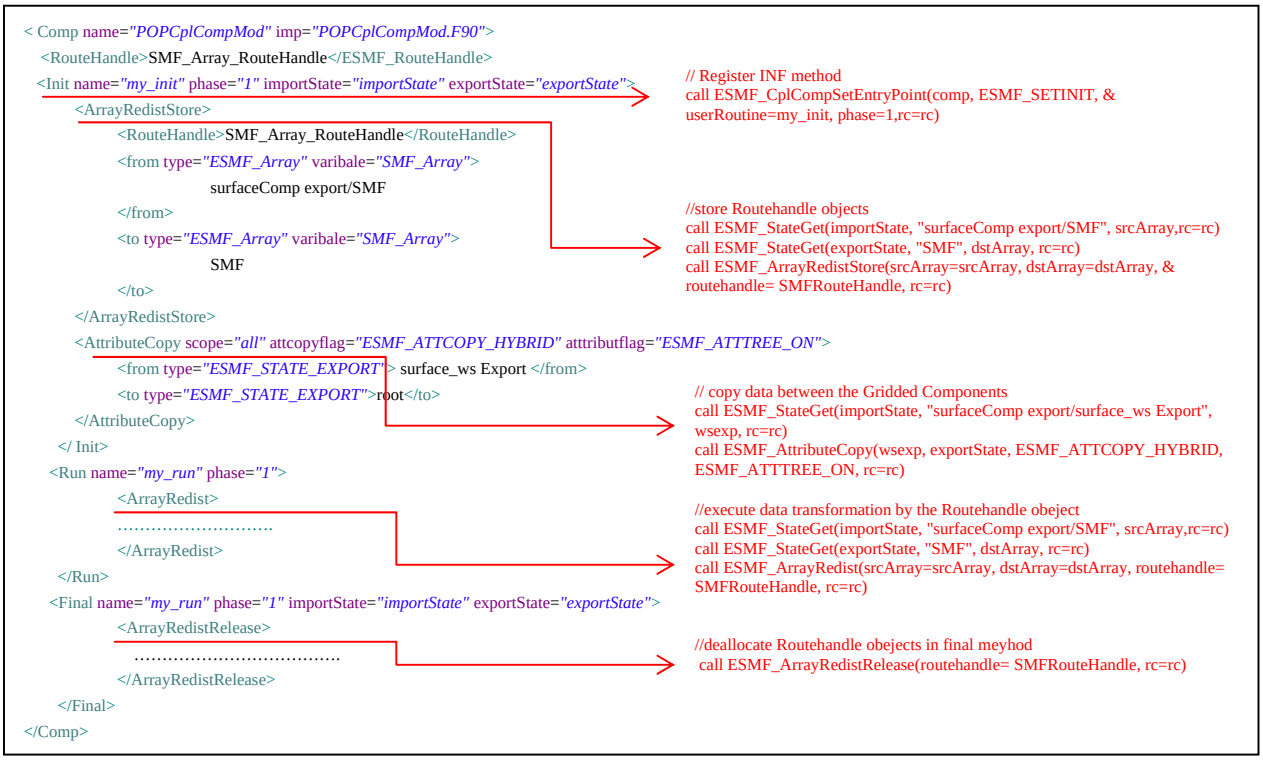


Figure 8. Mapping between ESMCDL and code for CC1 Coupler Componen

We simulated the results of the original POP and the component-based POP. Tests were run on the quad-core Intel(R) Xeon(R) CPU/2.40GHz. The environment of the platform is: RedHat 5.4, MPI: OpenMPI-1.4.1, FORTRAN Compiler: ifort-10.1.022.

We assume that if the K.E. diagnostic and tracer diagnostic in the output log file specified by the name list flag “log _filename” are congruent to both types of POPs; the reconstructed result for the component-based POP is true. The result of the experiments is shown in Table 1.

TABLE I. COMPARISON FOR THE COMPONENT-BASED POP AND ORIGINAL POP AT THE SAME PARALLEL DEGREE.

Parallel Degree	Original POP	Component-based POP	K.E	Tracer
1	7482s	7856s	true	true
2	4042s	4243s	true	true
4	1981s	2083s	true	true

VI. CONCLUSIONS AND FUTURE WORKS

With the development of Earth System Science, Software scale becomes increasingly large. In order to improve the earth system researchers' development efficiency, this paper proposes an Earth System Model Component Description Language and then based on this language introduces a graphical development tool which has been widely used by the earth system researchers from China. We have presented a case study that demonstrates the process of creating one earth system model application by using our graphical development tool. We conclude that the ESMCDL–based graphical development tool not only improves development efficiency for ESM application but also accumulates plenty of reused components and templates which will offer help to other researchers.

In the future, we will continue our research in the following directions: 1) improving the performance of the component-based POP by introducing the parallel technology; 2) expanding the ESMCDL to describe more complex behaviors of a component and logical relationships between components; 3) applying the ESMCDL and the tool to other earth system models to cumulate much more reusable components into our component library.

ACKNOWLEDGMENT

This paper is supported by the 863 project of China under the grant No. 2010AA012404, the China International Science and Technology Cooperation Program from the Ministry of Science and Technology of China under the grant No. 2009DFA12110.

REFERENCES

[1] Kauffman, B.G. and W.G. Large, “The CCSM Coupler Version 5101: Combined User’s Guide, Source Code Reference and Scientific Description,” National Center for Atmospheric Research, 2002, pp. 1-46.

 ISBN: 978-1-61208-184-7

[2] Hill, C., C. DeLuca, V. Balaji, M. Suarez, and A. DaSilva, "The architecture of the earth system modeling framework," Computing in Science and Engineering, 2004, Vol. 6, No.1 pp. 18-28, doi:10.1109/MCISE.2004.1255817.

[3] Eclipse, http://www.eclipse.org, June 2011.

[4] Microsoft Visual Studio, http://www.microsoft.com/visualstudio, July 2011.

[5] Gougen, "Parameterized programming," IEEE Transactions on Software Engineering, 1983, Vol.10, No. 5, pp. 528-543, doi:10.1109/TSE.1984.5010277.

[6] Gougen, "Reusing and interconnecting software component," IEEE Computer, Vol.19, No.2, February 1986, pp. 16-27, doi:10.1109/MC.1986.1663146.

[7] Paolo Bucci and Stephen H. Edwards, "Special Feature: Component-Based Software Using RESOLVE", Software Engineering Notes, ACM SIGSOFT, Vol. 19, No. 4, October 1994, pp.21-67.

[8] Whittle and M. Ratcliffe, "Software Component Interface Description for Reuse," IEEE BCS Software Engineering Journal, Vol.8, No.6. November 1993,pp. 307-318

[9] Tom Bulatewicz and Janice Cuny, "A domain-specific language model coupling," Proceedings of the 2006 Winter Simulation Conference, December 2006. pp. 1091-1100, doi:10.1109/WSC.2006.323199.

[10] Smith R.D. and Gent P., "Reference manual for the Parallel Ocean Program (POP)," Los Alamos Unclassified Report LA-UR-02-2484, 2002.

[11] Earth System Modeling Framework Reference Manual for Fortran Version 5.2, http://www.earthsystemmodeling.org, May 2011

[12] Photran, http://www.eclipse.org/photran/, September 2011.

[13] GMF, http://www.eclipse.org/modeling/gmp/, February 2011.

 ISBN: 978-1-61208-184-7

Magnetic Resonance Signal Processing in Medical Applications

Jan Mikulka, Eva Gescheidtová
Department of Theoretical
and Experimental Electrical Engineering
Brno University of Technology
Brno, Czech Republic
e-mail: mikulka@feec.vutbr.cz, gescha@feec.vutbr.cz

Karel Bartušek
Institute of Scientific Instruments
Academy of Sciences of the Czech Republic
Brno, Czech Republic
e-mail: bar@isibrno.cz

***Abstract*—Image processing in biomedical applications is an important developing issue. Many methods and approaches for image preprocessing, segmentation and visualization were described. It is necessary to choose a suitable segmentation method to create a correct three-dimensional model. The accuracy of reconstruction depends on precision of regions boundary determining in magnetic resonance slices. A frequent application is detection of soft tissues. To obtain images of the soft tissues mentioned, tomography based on magnetic resonance is usually used. Ideally, several tissue slices in three orthogonal planes (sagittal, coronal, transverse) are acquired. Following reconstruction of shape of examined tissues is the most accurate. In case of acquired slices only in one plane, the high spatial information lost occurs by image acquisition. Then it is necessary to reconstruct the shape of tissue appropriately. At first the images are segmented and with use of particular segments the three dimensional model is composed. This article compares several segmentation approaches of magnetic resonance images and their results. The results of segmentation by active contour, thresholding, edge analysis by Sobel mask, watershed and region-based level set segmentation methods are compared. The results for different values of parameters of segmentation methods are compared. As the test image, slice of human liver tumour was chosen.**

Keywords-magnetic resonance; biomedical image processing; image segmentation; level set; active countour; edge analysis; noise suppression; volumetry

I. INTRODUCTION

Image processing in biomedical applications is an important developing issue. Many methods and approaches for signal/image preprocessing, segmentation and visualizing were described. On the basis of segmented images, it is simple to describe the boundaries of the objects sought; these boundaries serve further processing aimed at calculating the perimeter, area, surface, volumetry or even 3D reconstruction of the object being imaged. To obtain images of the soft tissues mentioned, tomography based on magnetic resonance (MR) is usually used. It is necessary to reconstruct the shape of tissue appropriately. The shape is reconstructed via its segmentation in several slices. The reconstructed model has staggered shape. There are several methods for smoothing the shape. In this article the methodology for shape smoothing is discussed. The results of volumetry with use of several smoothing levels are compared. Impact of shape smoothing to quality of reconstruction is discussed. It is shown that high level of smoothing suppresses the staggered shape but the edge information is lost. With increased smoothing level the staggered shape of the model comes to be suppressed, with the transitions between individual model segments suppressed.

For a comparison of the segmentation properties of the methods, two sets of real medical images were chosen: MR images of the human liver with a tumour visible in several slices, MR images of the human head for the processing of the region of temporomandibular joint (TMJ). The segmentation of the mandibular disc is made difficult not only by the low contrast and the presence of noise but also by the fact that in most of the images obtained the region of the mandibular disc is represented by a very small number of pixels. The liver region exhibits homogeneous distribution of intensity, which is lower than in the surrounding area of the liver. The topicality of the selected topic of liver tumour segmentation is attested by the diverse conferences held and papers published [1] [2] [3] [4] [5] [6].

In the paper, an image segmentation method is described, which reduces the requirements for image pre-processing (elimination of noise) and yields good results also when segmenting a noisy image [7][8][9]. This is of much advantage when processing images exactly by the MR method. State of the art is mentioned in the next chapter. Some publications in the area of liver tumour segmentation are mentioned. It is followed by describing used mathematical models in the next chapter. Implementation problems, results and their comparison with other segmentation methods are shown in rest of the paper.

II. STATE OF THE ART

Extensive research was conducted in the area of processing MR images of tumours in the human liver in the past. The liver region exhibits homogeneous distribution of intensity, which is lower than in the surrounding area of the liver.

 ISBN: 978-1-61208-184-7

The topicality of the selected topic of liver tumour segmentation is attested by the diverse conferences held and papers published. Evidently, the greatest impulse to investigate methods for processing liver images was the workshop "3D Segmentation in the Clinic: A Grand Challenge II" [1], which was part of the conference "Medical Image Computing and Computer Assisted Intervention 2008". The aim of holding the Conference was a) to define the input data (64 or 40 slices in each set of images obtained by computer tomography were available, slice thickness 1 – 1.5 mm), and b) to define the criteria for the evaluation of the methods In [4], a chain of processing CT images of the human liver is described which, in brief, consists of image pre-processing – histogram-based segmentation of the region, multimodal thresholding, maximum a posteriori decision-making, and of the segmentation of the liver region, which is given by the basic local properties. The image processing chain gives very good results. A disadvantage can certainly be seen in the many degrees of freedom of this chain. A simpler approach is described in [3]. In the proposed methodology, the pre-processing of images is eliminated. Prior to the segmentation using the level set method the images are pre-segmented by fuzzy cluster algorithms. The author demonstrates the segmentation function on several selected CT images. But, the results are not discussed and there is no mention of important parameters such as segmentation speed or differences in comparison with the actual/real division of tissues, for example by tracing manually the tumour edges. A similar approach, segmentation of liver tumour area by the level set method, is described in [4]. The level set segmentation is preceded by the initialization method, which pre-processes the image prior to its segmentation proper. This initialization method is the so-called spiral-scanning method with supervised fuzzy pixel classification. The paper describes the segmentation of the liver tumour area in images produced by computer tomography. We can also come across methods that are based on segmentation approaches that are today considered traditional, e.g. the watershed method [5], thresholding method [6] and region growing method [7]. The development of the method for the segmentation of liver tumours follows from the previously published work [8], in which region-based level set segmentation was used.

The greatest disadvantage of the methods described above can be seen in that either a greater interaction of the physician in the segmentation of the liver tumour in tomographic images is required or it is necessary to choose a large number of segmentation parameters or to pre-process the images.

III. MATHEMATICAL MODELS

The selected segmentation methods are based on the solution of partial differential equations. These are iterative algorithms with initial conditions whose solution is used to shape the curve placed in the image. The steady-state solution is a curve delineating the image regions, which satisfies the sought minimum of the energy function of the mathematical model of a given method. The problem is to delineate by a smooth closed curve only the tumour region such that the curve does not delineate any other tissues. This can happen in the comparatively frequent case when the tumour is at the liver periphery; it is then important not to exceed the liver boundary. Better results were obtained using the edge-based analysis, which satisfied the above condition of delineating the tumour region alone.

The edge-based segmentation is described by this energy functional [9]:

$$F(\phi) = \lambda \int_{\Omega} g\delta(\phi)|\nabla\phi|\,\mathrm{d}x\mathrm{d}y + \nu \int_{\Omega} gH(-\phi)\,\mathrm{d}x\mathrm{d}y, \qquad (1)$$

where the first term means the length of the zero level curve of Φ (level set distance function) and the second term is called weighted area of Ω_{Φ}^{-}. λ and ν are the weighted coefficients of the mentioned terms, $\delta(\phi)$ is the Dirac function and H is the Heaviside function. The g function is the edge indicator defined by [10]

$$g = \frac{1}{1+\left|\nabla G_{\sigma} * I\right|^{2}}, \qquad (2)$$

where I is the original image and G_{σ} is the Gaussian kernel with standard deviation σ. By calculus of variation, the first variation of the functional in (2) can be written as [9]

$$\begin{aligned}\frac{\mathrm{d}\phi}{\mathrm{d}t} = \mu\left[\Delta\phi - \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] + \\ +\lambda\delta(\phi)\,\mathrm{div}\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) + \upsilon g\delta(\phi).\end{aligned} \qquad (3)$$

This gradient flow is the evolution equation of the level set function Φ. The second and third term in the equation (3) correspond to the length and area energy functional. The first term penalizes the deviation of the level set function from a signed distance function during its evolution.

The region-based segmentation method is of greater advantage when segmenting an image in which there are no sharp transitions or in the case when the extraction of an object in the image is required when the statistical properties of intensities at the site of the object sought are known. With this approach the principle is that no edges are sought in the image – regions in the image are viewed according to the local intensity statistics and, according to the given properties, the image is subdivided into two or more regions. The model of which is given by the energy functional [10]:

$$\begin{aligned}F_n(\mathbf{c},\boldsymbol{\Phi}) = \sum_{1\le I\le n=2^m} \int_{\Omega} \left(u_0(x,y) - c_I\right)^2 \chi_I\,\mathrm{d}x\mathrm{d}y \\ + \sum_{1\le i\le m} \nu \int_{\Omega} \left|\nabla H(\Phi_i)\right|\end{aligned} \qquad (4)$$

and for the two-phases result the general energy functional (4) is in shape [10]:

$$F(c_1,c_2,\phi)=\int_\Omega (u_0(x,y)-c_1)^2 H(\phi)\,\mathrm{d}x\mathrm{d}y + \int_\Omega (u_0(x,y)-c_2)^2 (1-H(\phi))\,\mathrm{d}x\mathrm{d}y + \nu\int_\Omega |\nabla H(\phi)|, \quad (5)$$

where the first two terms divide the area Ω of the original image u_0 to two subareas with the mean values of intensity c_1 and c_2. The third term minimizes the length of the resultant contour. It can be used for suppression of noise in the image and the final contour is smoother. This term is weighted by the coefficient ν. *H* is the Heaviside function. This function recognizes where the level set function ϕ is positive, respectively negative.

$$\frac{\partial \Phi}{\partial t}=\delta_\varepsilon(\Phi)\left[\nu\,\mathrm{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)-(u_0-c_1)^2+(u_0-c_2)^2\right]. \quad (6)$$

This gradient flow [10] is the evolution equation of the level set function *Φ*. The first term in the brackets (6) corresponds to the length functional.

Thresholding - the simplest segmentation method is defined as:

$$g(i,j)=\begin{cases}1 & \text{pro } f(i,j)\geq \mathrm{T}\\ 0 & \text{pro } f(i,j)<\mathrm{T}\end{cases}. \quad (7)$$

The biggest disadvantage of this method is that it is very sensitive to noise, but it is very often used in medical practice in manual processing. It is very simple and fast.

Image segmentation based on Sobel mask convolution is defined by convolution kernel:

$$S_x=\begin{pmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{pmatrix}, S_y=\begin{pmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{pmatrix}. \quad (8)$$

IV. IMPLEMENTATION

The algorithm was implemented in Matlab 7.0. The equation (3) was approximated by central and forward difference schemes and solved by iterative process. The Dirac function was approximated by [10]:

$$\delta_\varepsilon(x)=\begin{cases}0, & |x|>\varepsilon\\ \frac{1}{2\varepsilon}\left[1+\cos\left(\frac{\pi x}{\varepsilon}\right)\right], & |x|\leq\varepsilon\end{cases}. \quad (9)$$

V. RESULTS, LIVER TUMOUR SEGMENTATION

The aim of processing is to segment the tumour in all slices, reconstruct the obtained segments back to a 3D image and calculate the tumour volume so that the tumour evolution in time can be monitored (Fig. 1) In the case of liver tumour, a correct diagnosis is very important as it is decisive in determining the treatment procedure. The main drawbacks of traditional segmentation methods include, in the first place, the necessary individual pre-processing of images in order to suppress their unfavourable properties (presence of noise in images, blurred edges, low contrast), the necessity of individually adapting the segmentation method according to the number of image subregions sought (segmentation into two (background/object) or more subregions), etc. The aim was to simplify as much as possible the image processing chain, i.e., to find a method by means of which it would be possible in ideal case to segment the regions sought, without the necessity of image pre- and post-processing. For the segmentation of liver tumours the edge-based segmentation method was selected [9][10]. Good results were obtained using the edge-based analysis, which satisfied the above condition of delineating the tumour region alone.

Figure 1. Example of MR slice through human liver; the tumour sought is in the delineated region.

Fig. 2 shows the results of segmenting a liver tumour by the active contour method based on the edge-based analysis of image. The contour smoothness is given by the filtering properties of the method itself, which thus does not require any pre-processing of the image (smoothing, filtering, focussing) and which delineates only the region of the tumour proper and, in spite of the very similar mean value of brightness does not evolve in time towards delineating some narrow regions connected with the tumour.

 ISBN: 978-1-61208-184-7

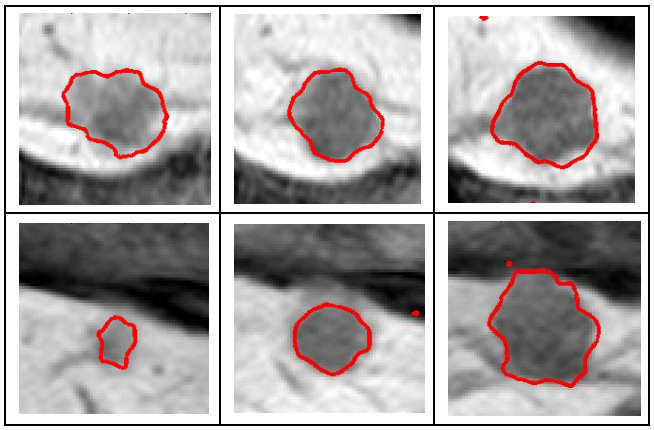


Figure 2. Results of segmenting a liver tumour by the active contour method based on edge-based analysis; 6 chosen slices.

VI. RESULTS, MANDIBULAR DISC SEGMENTATION

The aim of processing is to segment the mandibular disc in all slices and reconstruct the obtained segments back to a 3D image (Fig. 3). In the case of TMJ disorder in the disc area (rupture, dislocation) a correct diagnosis is very important as it is decisive in determining the treatment procedure. . The main drawbacks of traditional segmentation methods are similar to the previous one. The aim was to simplify as much as possible the image processing chain, i.e. to find a method by means of which it would be possible in ideal case to segment the regions sought, without the necessity of image pre- and post-processing. The MR slices with visible mandibular discs were segmented by edge-based level set segmentation method. This level set approach gives very good results in segmentation of noised MR images with low contrast and smooth edges so that it is not necessary to preprocess the image before the segmentation of image.

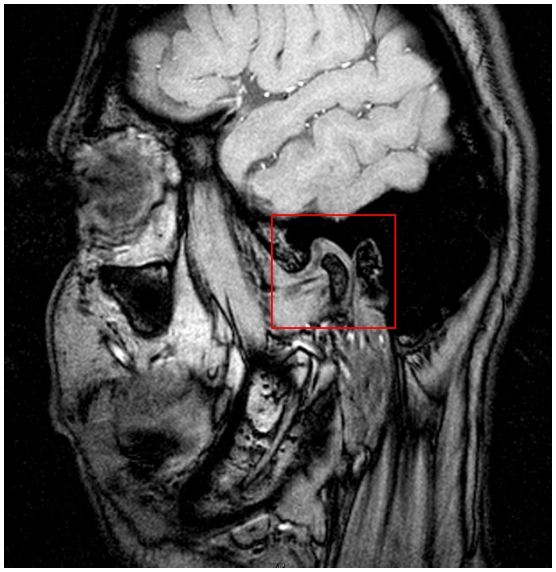


Figure 3. Example of MR image of human head for diagnosing TMJ, with delineated region of temporomandibular disc.

The results of segmentation without any kind of preprocessing are given in Fig. 4.

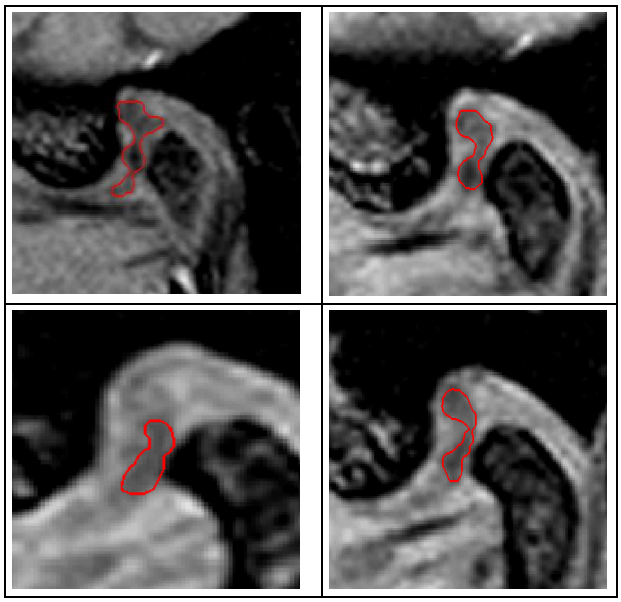


Figure 4. Result of segmentation of TMJ in four selected slices by edge-based segmentation active contur method.

VII. COMPARISON OF RESULTS WITH OTHER METHODS

In this section, a comparison is shown of the segmentation of a selected image (MR, human liver) and other traditional segmentation approaches. The first to be chosen was the simplest segmentation method, which is used very often in medical practice and is supported by the majority of professional software applications designed for MR images that are processed by physicians. This method is thresholding. Fig. 5 gives the result of segmenting the image of a slice through liver tumour by the thresholding segmentation method with two different thresholds, which were established empirically (100, 150). A mere subjective assessment is enough to conclude that in spite of its simplicity and speed, this method cannot be used to process such an image. With a low threshold level the darker regions inside the tumour were segmented while with a higher threshold level a contour was found that delineates the tumour region and goes through the tumour “edge” but this curve is not closed and penetrates farther into the liver region, out of the tumour. It is obvious that in the case of processing a large set of images the search for a suitable threshold would be demanding and the segmented images would have to be further processed in order to obtain a complete segmentation result.

Fig. 6 gives the result of the Sobel mask edge-based analysis [1, 2] with two different levels of thresholding the edge-based analysis image (0.05, 0.15).The edge-based analysis yields results similar to those of the segmentation thresholding method. Choosing a low threshold value gives an oversegmented image while a higher threshold will yield information only on very pronounced edges in the image.

 ISBN: 978-1-61208-184-7

The other edge analyzers give similar results; the result of edge-based analysis must be additionally processed and can be used, for example, as additional information for another segmentation method. Practically, never does this method give a closed curve representing the edges of the region of interest sought.

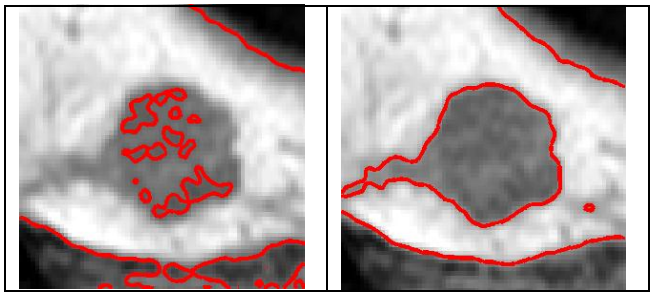


Figure 5. Result of segmenting the image of a slice through liver tumour via thresholding with thresholding levels a) 100, c) 150, within a brightness intensity range of 0 – 255.

Fig. 7 shows the result of image segmentation of a slice through liver tumour using the water shed segmentation method [1]. This segmentation method gives good results and is frequently used in practice. It has, however, one great disadvantage – the result of segmentation without prior image processing is oversegmented and the image must practically always be adapted in an appropriate way. Fig. 7 b) gives the result of watershed image segmentation after prior processing of the grey-tone image by thresholding (with automatic threshold search) and by transforming the binary image into a grey-tone image representing the Euclidean distance of every single pixel of the binary image from the background. The watershed image segmentation with prior image processing gives very good results. However, the method is dependent on an appropriate determination of the threshold of primary segmentation and the segmented region of the tumour reaches into the liver region since the method responds to any tiny interruption of the edge by merging the regions.

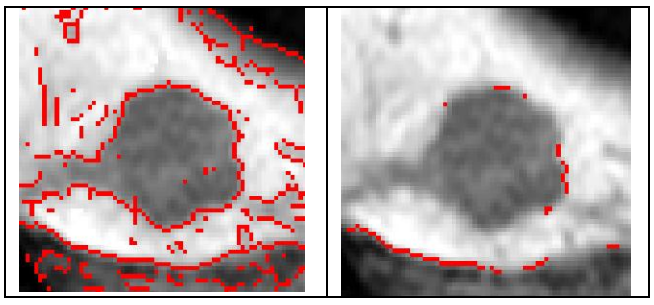


Figure 6. Result of edge-based analysis of a slice through liver tumour using the Sobel mask, with threshold values a) 0.05, c) 0.15.

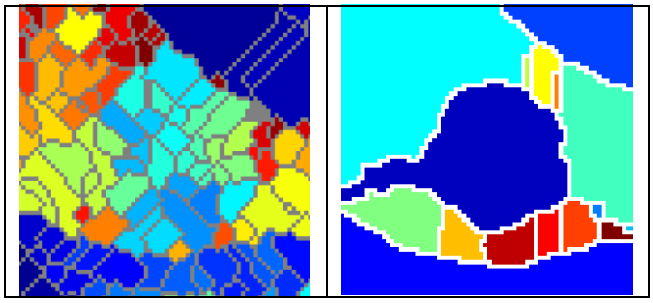


Figure 7. Result of image segmentation of a slice through liver tumour using the watershed segmentation method; a) oversegmented result without pre-processing, b) with pre-processing.

VIII. SUMMARY

Results of image processing show that the active contour method based on the level set principle is very appropriate for the segmentation of both low-contrast images and regions with interrupted edges. An example of the first type of task, namely the segmentation of regions in low-contrast images, was demonstrated on the processing of MR images in the TMJ region, specifically the TMJ disc. The segmentation of regions with interrupted edges, on the other hand, is demonstrated by the application of active contours in the processing of MR images of human liver. In contrast to other traditional segmentation methods, the active contour method was always able to segment the given region of interest. The segmentation result is always a closed curve delineating the tissue sought.

IX. CONCLUSIONS AND FUTURE WORKS

The segmentation of regions of interest in MR images is an important part of the image processing chain. The quality of separating the region of interest from the background determines the quality of further processing. This may include the quantification of the delineated regions such as establishing the dimensions, area and volume or three-dimensional reconstruction and visualization of objects.

The future work will be concerned with registration of segmented images and creation of the 3D model of the region of interest and its visualization.

ACKNOWLEDGMENT

This work was supported within GACR 102/11/0318, CZ.1.05/2.1.00/01.0017 (ED0017/01/01), and FEKT-S-11-5/1012.

REFERENCES

[1] Deng, X. and Du, G. Editorial: 3D Segmentation in the Clinic: A Grand Challenge II – Liver Tumour Segmentation. 2008.

[2] Seo, K. and S., Chung, T. W. Automatic Boundary Tumour Segmentation of a Liver. ICCSA 2005, LNCS 3483, pp. 836-842, 2005.

[3] Li, B., N., Chui, Ch., K., Ong, S., H. and Chang, S. Integrating FCM and Level Sets for Liver Tumour Segmentation. ICBME 2008, Proceedings 23, pp. 202-205, 2009.

 ISBN: 978-1-61208-184-7

[4] Smeets, D., Loeckx, D., Stijnen, B., Dobbelaer, B., Vandeurmeulen, D. and Suetens, P. Semi-automatic level set segmentation of liver tumours combining a spiral-scanning technique with supervised fuzzy pixel classification. Medical Image Analysis, vol. 14, 2010, pp. 13-20.

[5] Stawiaski, J., Decenciere, E. and Bidault, F. Interactive Liver Tumour Segmentation Using Graph-cuts and Watershed. MICCAI 2008.

[6] Abdel-Massieh, N., H., Hadhoud, M., M. and Amin, K., M. Automatic Liver Tumour Segmentation from CT Scans with Knowledge-based Constraints. 5th Cairo International Biomedical Engineering Conference, 2010, pp. 215-218.

[7] Qi, Y., Xiong, W., Leow, W. K., Tian, Q., Yhou, J., and Liu, J. Semi-automatic Segmentation of Liver Tumours from CT Scans Using Bayesian Rule-based 3D Region Growing. MICCAI 2008 Workshop 3D Segmentation in the Clinic, 25.

[8] Mikulka, J., Gescheidtová, E. and Bartušek, K. Processing of MR slices of human liver for volumetry. In PIERS 2010 in Xi' an Proceedings. 2010. pp. 202-204. ISBN: 978-1-934142-12- 7.

[9] Li, Ch., Xu, Ch., Gui, Ch. and Fox, M., D. Level set evolution without re-initialization: A new variational formulation. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'05. San Diego (USA): IEEE Computer Society Washington, DC, USA, 2005, pp. 430–436. ISBN 0-7695-2372-2.

[10] Aubert, G. and Kornprobst, P. *Mathematical problems in image processing: Partial differential equations and the calculus of variations*. 2nd edition. New York : Springer Science + Business Media, LLC, 2006. ISBN 0-387-32200-0.

 ISBN: 978-1-61208-184-7

Magnetic Susceptibility Measurement from Spatially Mapped Reaction Field

Petr Marcon, Eva Gescheidtova
Brno University of Technology, Department of Theoretical and Experimental Electrical Engineering
Brno, Czech Republic
marcon@feec.vutbr.cz
gescha@feec.vutbr.cz

Karel Bartusek
Institute of Scientific Instruments, Academy of Sciences of the Czech Republic
Brno, Czech Republic
bar@isibrno.cz

***Abstract*—This article is focused on the principles of the magnetic susceptibility measurement of the non-ferromagnetic substances using NMR tomography. Magnetic susceptibility is calculated from changes of the magnetic field close to the cylinder shaped specimen. Changes of the magnetic field in the 3D vicinity of the specimen were integrated. Before integrating of the magnetic field changes it was necessary to filter, unwrap and detect the accurate position of the specimen. Magnetic susceptibility calculated from measured data is in comparison to theoretical value, as well as model value, only slightly different.**

Keywords-NMR; magnetic susceptibility; reaction field; 3D vicinity of specimen

I. INTRODUCTION

An inhomogeneous static magnetic field (B_0 filed) generates distortion in magnetic resonance images. Measuring the spatial variation of B_0 is essential for automatic shimming. The domain source of field inhomogeneity in many MRI experiment is the variation of magnetic susceptibility of both the tissues of the human body and the implant materials.

Magnetic susceptibility provides information on the tissue relative iron concentration that is useful for diagnosis and treatment of a number of diseases such as sickle cell disease, aplastic anaemia, thalassemia, haemochromatosis and Parkinson´s disease. In magnetic resonance imaging, the susceptibility effects have Essentials relevance for imaging contrast and artifact correction, functional brain imaging, molecular imaging and the measurement of blood oxygenation. Thus, it is highly significant to develop methods that can measure arbitratry susceptibility distributions.

The knowledge of the magnetic susceptibility of tissues or various implants can help us to minimize the effect of magnetic susceptibility in MRI images via modification of pulse sequences, i.e. to correct artifacts in MRI images. These artifacts are manifested by the loss of signal, and new artifacts appear in the vicinity. For example, functional brain imaging using the gradient-echo planar imaging is based on blood oxygenation level-dependent (BOLD) susceptibility effects, which are believed to be dependent on neuronal activity in specific regions of the brain, as a result of cognitive tasks of human subjects (as well as animals) [1, 2]. Depending on the location of the cortical areas that are involved in an fMRI study, the BOLD effect is strongly influenced by local field inhomogeneities created by differences in magnetic susceptibility – between air and tissue for example, and results in severe image distortion and signal loss. Signal loss is a major problem in fMRI studies [3, 4]. The measurement of magnetic susceptibility of magnetically compatible materials (the NMR measuring method enables obtaining a signal even inside the specimen) has been the subject of study undertaken by Lin, who calculates susceptibility using a delineated area inside the specimen [5, 6].

In this paper, we deal with magnetic susceptibility measurement of non ferromagnetic materials in macroscopic view. For calculating of magnetic susceptibility we used 3D (three dimensional) mapped reaction filed in the vicinity of specimen. Chapter II and III the basis of method of magnetic susceptibility measurement is described. Our experimental measurements and signal processing are mentioned in chapter IV. The conclusions and discussions about our results are presented in chapter V.

II. METHODS

This method of susceptibility measurement is based on the assumption of constant magnetic flux in the working space of superconducting magnet. Inserting a specimen with magnetic susceptibility χ_s causes local deformation of previously homogeneous magnetic field – for illustration see Fig. 1.

The magnitude of these deformations depends on the difference of magnetic susceptibility of the specimen χ_s and of its vicinity χ_v, on the volume and shape of the specimen, and on the magnitude of basic field B_0.

 ISBN: 978-1-61208-184-7

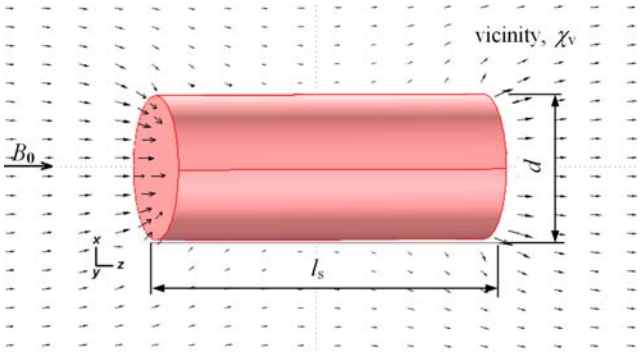


Figure 1. Magnetic flux density field deformation due to paramagnetic specimen.

III. THEORETICAL ANALYSIS

The method was verified via an experiment with a number of specimens on a 4.7 T/76 mm MR tomograph in ISI AS Brno (^{1}H ≈ 200 MHz). The specimens to be measured were of different materials in the shape of cylinders 3 mm in diameter and 10 mm in length (No. 2 in Fig. 2). They were placed in a glass container in the shape of a 40 x 40 x 40 mm cube (No. 1 in Fig. 2) filled with a solution of water with this concentration: 1 liter deionized water, 1.2 gram $NiSO_4$ and 2.6 gram NaCl in order to shorten the relaxation times to $T_1 = T_2 = 130$ ms. Magnetic susceptibility of this solution was $\chi_v = -13.0 \cdot 10^{-6}$. When measuring materials with nearly the same susceptibility, the reaction field will not be induced.

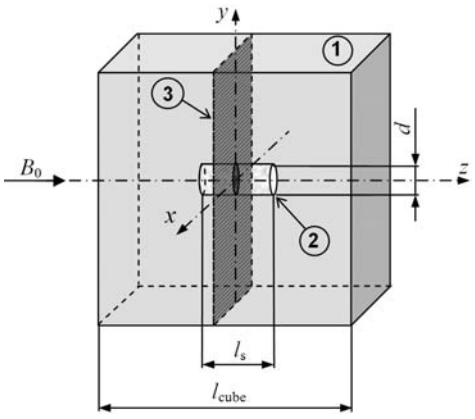


Figure 2. Configuration of system – vicinity with specimen.

One of the MR measurement methods – the Gradient echo (GE) method is very sensitive to inhomogeneities of the static magnetic field and this can be useful for susceptibility measurement [5]. Because the reaction field is generated proportionally to material susceptibility, it is possible to use the GE method for its measurement. The GE sequence depicted in Fig. 3 with the parameters: echo time T_E =17 ms, repetition time T_R = 5 s, was used to obtain an MR image of the reaction field in the vicinity of the measured specimen.

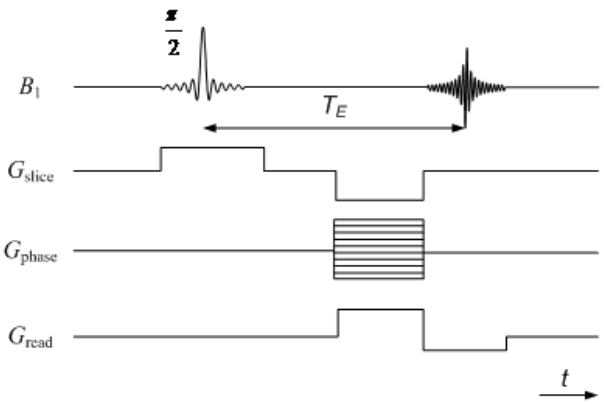


Figure 3. Diagram of the Gradient echo measurement sequence used.

The MR image obtained using the GE technique is phase-modulated by the magnetic induction change and, on condition of proper experiment arrangement we can obtain the image of magnetic field distribution in the specimen vicinity. For the calculation of the reaction field ΔB we can give the following relation:

$$\Delta B = B - B_0, \tag{1}$$

which is the material's own field caused by magnetization. Transversal magnetization $M_\perp$ is for the GE method described by the equation:

$$M_\perp(T_E) = M_0(T_E)\, e^{-\frac{T_E}{T_2^*}} e^{-j\gamma \Delta B T_E}, \tag{2}$$

where M_0 is the transversal magnetization obtained immediately after excitation, which has been exponentially decreased in time by e^{-T_E/T_2^*}. Here T_2^* is effective relaxation time. The term $e^{-j\gamma \Delta B T_E}$ describes the phase modulation of magnetization, induced by reaction field ΔB. It is evident that the phase part of complex image can be used to obtain the spatial distribution of reaction magnetic field flux density ΔB (see [7, 8]):

$$\Delta B = \frac{\Delta \varphi}{\gamma \cdot T_E}, \tag{3}$$

where γ is the gyromagnetic ratio of reference substance, $\Delta\varphi$ is the phase image, and T_E is the echo time of the GE measuring sequence. From (2) we can see two opposite requirements for the echo time: with longer time T_E we have a magnetization which is more sensitive to the reaction field, but due to relaxation time T_2^* we also have a lower signal-to-noise ratio.

 ISBN: 978-1-61208-184-7

IV. EXPERIMENTS AND RESULTS

From magnetic resonance system we obtain two dimensional data (image matrix 128 x 128 px). An example of 1 slice you can see in Fig. 2 point 3. The measurement was made for 64 slices. These data were further processed in the Marevisi and Matlab programs. To remove the effect of the inhomogeneities of magnetic field background we used two measurements – with embedded specimen and blind measurement (without specimen). The procedure of processing data from the two measurements is indicated in Fig. 4. Both data matrices were transformed using FFT in the Marevisi program. Marevisi is a program for data processing and visualization for MRI [9]. The program was developed by Jana and Zenon Starcuk and Piotr Kozlowski. Further processing continues in the Matlab program, using the algorithm created. From the complex 3D data points, only the phase component was further used. Because the phase image was periodically wrapped (this means discontinuities in the phase change between $-\pi$ and π), the next operation was unwrapping. To map the reaction field, measuring with the GE pulse sequence for two echo times T_E can be employed [10,11]. The blind phase image was subsequently subtracted from the one with specimen to eliminate the inhomogeneity of the basic tomograph field. The differential phase image was converted to the reaction field magnetic flux density using (3). Inside the specimen are any hydrogen atoms. Therefore we obtain any inside the specimen any useful signal, but only noise. For calculation of magnetic susceptibility we use only vicinity of the specimen and we replace the measured data inside in specimen with zeros.

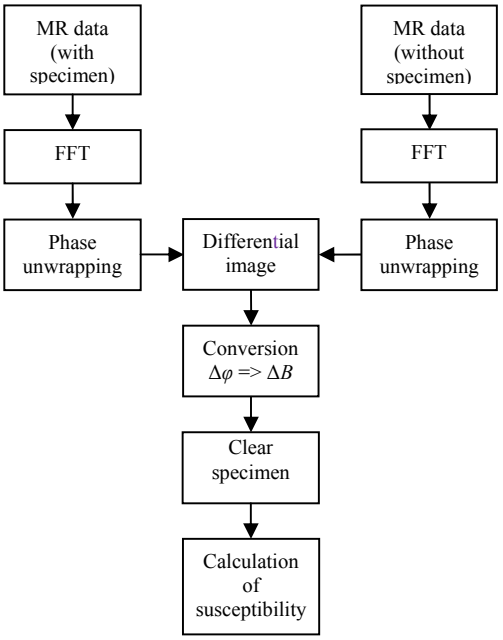


Figure 4. Diagram of image processing of obtained MR GE phase images. Input data are two complex matrices (MR image with and without specimen).

Fig. 5 is the graphical representation of the distribution of reaction magnetic field ΔB in a section through the aluminum specimen measured (the section corresponds to point 3 in Fig. 2). The picture was created in Matlab program from the measured values of reaction field in the vicinity of measured specimen. In the picture you can see the zeros place, there was measurement. Magnetic susceptibility was calculated from 3D distribution of the reaction magnetic field by relation (4).

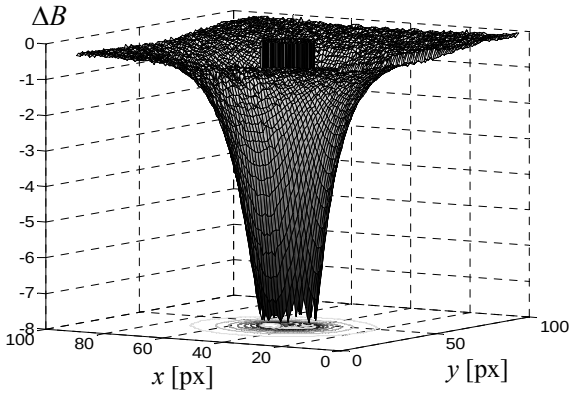


Figure 5. Distribution of reaction magnetic field ΔB in a section through plane x,y in the centre of aluminium cylinder measured.

The last step is susceptibility calculation. For discrete processing of the data measured will lead to equation (4). This equation will be used to calculate differential magnetic susceptibility χ_Δ. To obtain magnetic susceptibility of the specimen χ_s we substitute the value χ_Δ into equation (5).

$$\chi_\Delta = \frac{1}{M \cdot N \cdot P} \sum_{x=1}^{M} \sum_{y=1}^{N} \sum_{z=1}^{P} \Delta B_{xyz} = \frac{1}{|V_s|} \sum_{V_s} \Delta B \,, \qquad (4)$$

where V_s is the volume of the measured specimen in the 3D image.

By relation (5), magnetic susceptibility of the specimen χ_Δ can be calculated from the differential value of magnetic susceptibility:).

$$\chi_s = -\frac{2\chi_\Delta + \chi_v(\chi_\Delta + 1)}{\chi_\Delta - 1}, \qquad (5)$$

where χ_v is magnetic susceptibility of water in the cylinder vicinity.

The results of our measurement you can see in the Table I. There, the known values of susceptibility are denoted χ_k and the values obtained by measuring and data processing are denoted χ_m.

 ISBN: 978-1-61208-184-7

TABLE I. RESULTS OF MAGNETIC SUSCEPTIBILITY OF SPECIMEN

Material	Purity [%]	Shape	Size [mm]	χ_k [$\times 10^{-6}$]	χ_m [$\times 10^{-6}$]
Al	99.50	Cylindrical bar	d=4.00; l_s=10.00	22.00	22.71
Cu	99.91	Cylindrical bar	d=2.70; l_s=10.05	-9.60	-9.72

V. CONCLUSION

The calculation of magnetic susceptibility by relations (4) and (5) was verified experimentally on an MR tomography system and via processing the data measured as shown in Fig. 4. The specimens were a copper and an aluminium cylinder, of known magnetic susceptibility and purity (see Table I). In comparison with the known magnetic susceptibility values the measuring error is in both cases less than 3.23 %. However, there are some limitations to the proposed method. A disadvantage of the proposed method consists in that the accuracy of magnetic susceptibility calculation depends on the shape of specimen (have to be cylinder or block). In the future, the measuring method can be used to establish the susceptibility of tissues or implants in the human body and, thanks to this knowledge; it will be possible to attenuate the artifacts in MRI images produced by these materials.

ACKNOWLEDGMENT

This work supported within GA102/1/0318 and CZ.1.05/2.1.00/01.0017 (ED0017/01/01) and FEKT-S-11-5/1012.

REFERENCES

[1] S. Ogawa, T. M. Lee, A. S. Nayak and P. Glynn, "Oxygenation-sensitive contrast in magnetic resonance imaging of rodent brain at high magnetic fields," Magnetic Resonance in Medicine, vol. 14,1990, pp. 68-78, 1990.

[2] R. Deichmann, O. Josepshs, D. Hutton, D. R. Corfiled and R. Turner, "Compensation of susceptibility- induced BOLD sensitivity losses in echo-planar fMRI imaging," Neuroimage, vol. 15, pp. 120-135.

[3] J. Y. Chung, H. W. Yoon, Y. B. Kim, H. W. Park and Z. H. Cho, "Susceptibility compensated fMRI study using a tailored RF echo-planar imaging sequence," Journal of Magnetic Resonance Imaging, vol. 29, 2009, pp. 221-228.

[4] L. Li, "Magnetic susceptibility quantification for arbitrarily shaped objects in inhomogeneous fields," Magnetic Resonance in Medicine, vol. 46, 2001, pp. 907-916.

[5] P. Marcon, K. Bartusek, M. Burdkova and Z. Dokoupil, "Magnetic Susceptibility measurement using 2D magnetic resonance imaging," Measurement Science and Technology, vol. 22, 2011, doi:10.1088/0957-0233/22/10/10570.

[6] K. Bartusek, Z. Dokoupil, E. Gescheidtova, "Magnetic field mapping around metal implants using an asymmetric spin-echo MRI sequence," Measurement Science and technology, vol. 17, 2006, pp. 3293-3300.

[7] M. Steinbauer and K. Bartusek, "Magnetic susceptibility measurement using magnetic resonance tomograph," Acta Technica CSAV, vol. 53, 2008, pp. 45-63.

[8] M. Vlaardingerbroek, "Magnetic Resonance imaging," Springer Verlag, 2000.

[9] K. Bartusek, E. Kadlecova, P. Fiala and J. Mikulka, "Magnetic field deformation numerical modelling in relation to measured susceptibility using MR system," Radioengineering, vol. 17, 2008, pp. 249-266.

[10] B. Guru and H. Hiziroğlu, "Electromagnetic field theory fundamentals," Cambrigdge University Press, 2004.

[11] K. Bartusek, Z. Dokoupil and E. Gescheidtova, "Mapping of magnetic field around small coils using the magnetic resonance method, Measurement Science and Technology, vol. 18, 2007, pp. 2223-2230.

[12] P. Andris and I Frollo, "Optimized measurement of magnetic field maps using NMR", Measurement Science and Technology, vol. 22. 2011.

 ISBN: 978-1-61208-184-7

Hierarchical PLABs, CLABs, TLABs in Hotspot

Christoph M. Kirsch
University of Salzburg
ck@cs.uni-salzburg.at

Hannes Payer
University of Salzburg
hpayer@cs.uni-salzburg.at

Harald Röck
University of Salzburg
hroeck@cs.uni-salzburg.at

***Abstract*—Thread-local allocation buffers (TLABs) are widely used in memory allocators of garbage-collected systems to speed up the fast-path (thread-local allocation) and reduce global heap contention yet at the expense of increased memory fragmentation. Larger TLABs generally improve performance and scalability but only up to the point where more frequent garbage collection triggered by increased memory fragmentation begins dominating the overall memory management overhead. Smaller TLABs decrease memory fragmentation but increase the frequency of executing the slow-path (global allocation) and thus may reduce performance and scalability. In the Hotspot JVM a complex, TLAB-growing strategy implemented in several thousand lines of code determines the TLAB size based on heuristics. We introduce hierarchical allocation buffers (HABs) and present a three-level HAB implementation with processor- and core-local allocation buffers (PLABs, CLABs) in between the global heap and TLABs. PLABs and CLABs require low-overhead OS-provided information on which processor or core a thread executes. HABs may speed up the slow-path of TLABs in many cases and thus allow using smaller TLABs decreasing memory fragmentation and garbage collection frequency while providing the performance and scalability of otherwise larger TLABs. Our implementation works with or without the TLAB-growing strategy and requires two orders of magnitude less code. We evaluate our implementation in the Hotspot JVM and show improved performance for a memory-allocation-intensive benchmark.**

***Keywords*-memory management, garbage collection, virtual machines, scalability**

I. Introduction

Memory management in runtime systems like Java virtual machines (JVMs) may be a scalability bottleneck in applications with multiple threads accessing the global heap frequently. Thread-local allocation buffers (TLABs) reduce global heap contention by preallocating large pieces of memory from the global heap. The preallocated memory is stored thread-locally to handle allocation requests of a given thread. This approach does not only reduce contention on the global heap but also allows a fast-path for memory allocation that does not require any synchronization or atomic operations since the TLAB of a thread is not shared with any other threads. However, larger TLABs introduce additional memory fragmentation that depends linearly on the number of threads since large blocks of memory are committed to thread-local use only. High memory fragmentation may result in more frequent garbage collection which may decrease application throughput. To trade-off scalability and memory fragmentation in modern JVMs complex TLAB-growing strategies incorporate different factors like allocation rate, number of threads, heap size and feedback from the garbage collector to determine TLAB sizes for all threads. The implementation of such a strategy in the garbage collector of the Hotspot JVM [7] requires several thousand lines of code and thus significantly contributes to its complexity.

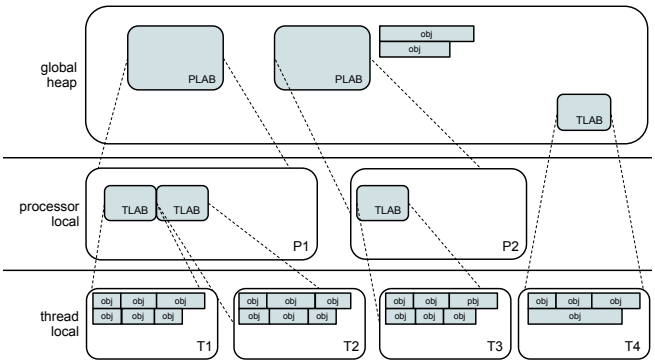


Figure 1. Hierarchical allocation buffers (HABs) with three levels: on the highest level is the global heap, in the middle are the PLABs or CLABs $P1$ and $P2$, and on the lowest level are the TLABs $T1 - T4$.

We introduce hierarchical allocation buffers (HABs), which consist of multiple levels of allocation buffers where an allocation buffer on a given level preallocates memory out of an allocation buffer on the next higher level. The traditional approach with TLABs is thus a two-level HAB system with the global heap on top and TLABs below. For recent multi-core architectures with several cores per CPU and several CPUs per machine we propose to use a three-level HAB system with one more level in between as depicted in Figure 1. In our implementation this level uses processor- or core-local allocation buffers (PLABs, CLABs) which require low-overhead OS-provided information on which processor or core a thread executes. PLABs and CLABs speed up the slow-path of TLABs in many cases and thus allow using smaller TLABs decreasing memory fragmentation and garbage collection frequency while providing the performance and scalability of otherwise larger TLABs. We show in experiments that a statically configured HAB system may provide similar performance as a TLAB-only system using a TLAB-growing strategy.

Our three-level HAB implementation reflects the under-

 ISBN: 978-1-61208-184-7

lying processor architecture of a server machine with four Intel Xeon E7 processors where each processor comes with ten cores and two hardware threads per core. The allocation buffers of the middle level can be configured to be PLABs or CLABs. The TLABs on the lowest level allocate from the PLABs or CLABs associated with the processor or core on which the allocating thread is currently running on. We evaluate the performance of our three-level HAB implementation integrated into the Hotspot JVM and show performance improvements due to better cache utilization and less contention on the global heap.

We summarize the contributions of this paper: (1) the notion of hierarchical allocation buffers (HABs), (2) the three-level HAB implementation with processor- or core-local allocation buffers (PLABs, CLABs), and (3) an experimental evalulation of HABs in the Hotspot JVM.

In Section II, we present the design of HABs. In Section III, we discuss the implementation of HABs in the Hotspot JVM and the required operating system support. Related work is discussed in Section IV, our experiments are presented in Section V, and conclusions are in Section VI.

II. PLABs, CLABs, TLABs

TLABs are an architecture-independent concept for implementing allocation buffers. Each thread maintains its private allocation buffer for fast allocation of memory. However, allocation buffers may also be implemented in an architecture-dependent fashion. For example, allocation buffers can be assigned to processors, i.e., a thread running on a processor may use the allocation buffer of the processor for its allocation requests [3]. We study the use of processor-local allocation buffers (PLABs) as well as core-local allocation buffers (CLABs) situated in between TLABs and the global heap. A PLAB is assigned to a given processor which may comprise of multiple cores. Threads running on different cores but on the same processor share the same PLAB. A CLAB is assigned to a given core. Threads running on the same core share the same CLAB. Using PLABs may increase parallelism and cache utilization and thus reduce contention. On multicore machines using CLABs over PLABs may increase parallelism and cache utilization even further. Note that the size of PLABs and CLABs should be multiples of the TLAB size to avoid additional internal memory fragmentation.

Access to PLABs and CLABs is done in two steps. First, the processor or core on which a given thread currently runs is determined (selection). Then, the actual allocation in the selected PLAB or CLAB is performed atomically (allocation). Selection and allocation is done non-atomically for better performance and scalability. Thus a thread may migrate to another processor or core in between selection and allocation resulting in what we call a foreign allocation. Note that the probability of foreign allocations is low and we show in experiments that foreign allocations indeed rarely happen.

III. Implementation

In this section, we discuss the implementation of HABs in the Hotspot JVM and present simplified pseudo-code of the core algorithm. Garbage-collection-specific details were removed for simplicity. We modified the heap implementation of Hotspot's parallel garbage collector for the upcoming JDK7. The modifications are limited to the code path of allocating new and refilling existing TLABs. Additionally, we disallow direct inlined access to the global heap. In our benchmarks we did not observe any slow down when removing inlined access to the global heap.

Listing 1 shows the method for allocating a TLAB of a given size. If a thread's TLAB is full the thread invokes this method to allocate a new memory region for its TLAB. TLABs larger than PLAB size are directly allocated from the global heap. For smaller TLABs, we try to allocate them in a PLAB with preference to the PLAB of the current processor. We iterate over all PLABs starting at the PLAB of the current processor and try to allocate a new TLAB. As soon as a TLAB is successfully allocated it is returned to the caller. If a TLAB could not be allocated in any PLAB, we fall back to allocate memory from the global heap. If this also fails the method returns NULL, and eventually a GC cycle will be triggered.

Listing 2 shows the method for allocating a TLAB of a given size on a dedicated processor (or core), and if the PLAB is full how it is refilled. If an allocation request cannot be handled the method returns NULL. The implementation uses an optimistic non-blocking approach to avoid expensive locking. The access to a PLAB is synchronized using the PLAB's top variable. The top variable indicates where the free memory in the PLAB starts or whether the PLAB is currently being refilled. If a thread detects that the PLAB is currently being refilled by another thread it returns NULL indicating to the caller that no allocations are currently possible in this PLAB. The top variable is always modified using a compare-and-swap operation.

If top is a valid pointer, the thread attempts to allocate the new TLAB in the PLAB. The allocation in the PLAB advances the top pointer by the size of the new TLAB using a compare-and-swap retry cycle. If the new TLAB does not fit into the PLAB it returns NULL, and the protocol to refill the PLAB with a new memory region is started. The refill protocol starts by setting top to PLAB_REFILL_IN_PROGRESS. The succeeding thread allocates a new memory region from the global heap and reinitializes the PLAB with the new memory region. If the allocation fails at any step, for example, when trying to set top to PLAB_REFILL_IN_PROGRESS or when trying to allocate a new PLAB, the method returns NULL. As mentioned in the previous section we tolerate context switches in

 ISBN: 978-1-61208-184-7

```
1 HeapWord* allocate_new_tlab(size_t size) {
2   if (size <= PLAB_SIZE) {
3     int hw_id = get_hw_id();
4     for (int i = 0; i < PLABS; i++) {
5       HeapWord* tlab = allocate_on_processor(size, (hw_id + i) % PLABS);
6       if (tlab != NULL) {
7         return tlab;
8       }
9     }
10  }
11  return allocate_in_global(size);
12 }
```

Listing 1. TLAB allocation

```
1 HeapWord* allocate_on_processor(size_t size, int id) {
2   HeapWord* top = plabs_[id].top();
3   if (top == PLAB_REFILL_IN_PROGRESS) {
4     return NULL;
5   }
6   HeapWord* tlab = plabs_[id].allocate(size);
7   if (tlab == NULL) {
8     HeapWord* result = cmpxchg(plabs_[id].top_addr(), top, PLAB_REFILL_IN_PROGRESS);
9     if (result == NULL) {
10      return NULL;
11    }
12    tlab = allocate_in_global(PLAB_SIZE);
13    if (tlab == NULL) {
14      plabs_[id].reset();
15      return NULL;
16    }
17    plabs_[id].init(tlab, size);
18  }
19  return tlab;
20 }
```

Listing 2. TLAB allocation on a given processor (or core)

between reading the processor ID and performing the actual allocation, which may result in foreign allocations. Moreover, a foreign allocation may also be performed by a thread that encounters that the PLAB_REFILL_IN_PROGRESS flag has been set. Note that this is a completely lock-free implementation for avoiding lock-related complications with system invariants in the stop-the-world garbage collector.

A. Operating System Support

Our current implementation requires that the underlying operating system provides a mechanism for threads to look up the processor and core they are running on. In recent Linux kernels the `getcpu()` system call is optimized to provide a low-overhead mechanism for determining the CPU on which the invoking thread runs. However, it is not guaranteed that the thread is still executing on the CPU after the call returns. Therefore, allocation in PLABs and CLABs have to be synchronized and could even result in foreign allocations where a thread on processor A allocates memory assigned to processor B.

In order to reduce the additional overhead when accessing PLABs or CLABs or even allow unsynchronized access to CLABs we would require additional support of the OS. For instance, a notification when the thread is preempted would suffice to detect possible migrations and context switches. If a thread detects that it was preempted in between determining the current processor and the actual allocation in the PLAB it could restart the operation. In [2] the authors introduce multi-processor restartable critical sections (MB-RCS) for SPARC Solaris, which provide a mechanism for user-level threads to know on which processor they are executing and to safely manipulate CPU-specific data.

IV. Related Work

Several JVMs already provide specific support for different processor architectures. For example, the latest version of the Hotspot JVM already supports NUMA architectures where the heap is divided into dedicated parts for each NUMA node.

PLABs were previously discussed in [3]. The implementation is based on a special mechanism called multi-

 ISBN: 978-1-61208-184-7

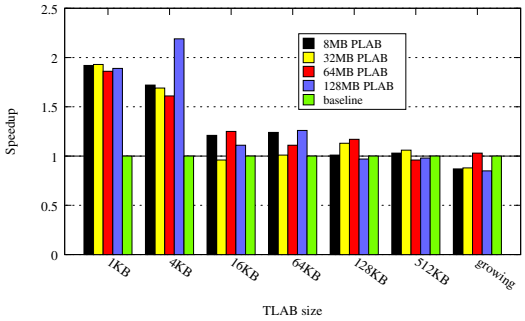
processor restartable critical section which allows to manipulate processor-local data consistently and guarantees that a thread always uses the PLAB of the processor it is running on. In our implementation we do not provide that guarantee. If there is a context switch between determining processor and PLAB operation and the thread is scheduled after that on a different processor we tolerate that. Moreover, just using PLABs eliminates the fast-path provided by TLABs. In our work we combine the benefits of both PLABs and CLABs with TLABs.

Thread- and processor-local data is not only relevant in allocators of JVMs but also in explicit memory management systems. Multi-processor restartable critical sections are used in [2] to implement a memory allocator that holds allocator-specific metadata processor-locally to take advantage of the cache and to reduce contention. McRT-Malloc [4] is a non-blocking memory management algorithm, which avoids atomic operations on typical code paths by accessing thread-local data only. Hoard [1] is a memory allocator that combines, in more recent versions, thread-local with non-thread-local allocation buffers.

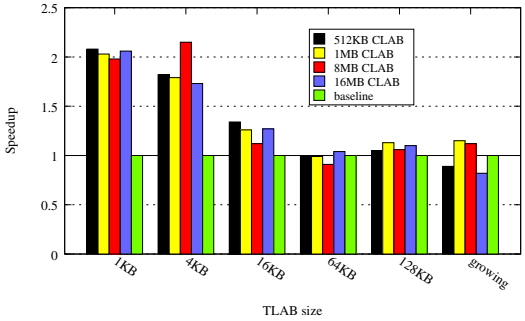
V. EXPERIMENTS

For our experimental evaluation we used a server machine with four Intel Xeon E7 processors where each processor comes with ten cores and two hardware threads per core, 24MB shared L3-cache, and 128GB of memory running Linux 2.6.39. We ran the SPECjvm2008 memory-allocation-intensive javac benchmark [5] with 80 threads on the Hotspot JVM in server mode [6]. Each benchmark run was configured to last six minutes with two minutes of warm-up time in the beginning. We repeated each experiment seven times and measured the performed operations per minute. The default generational garbage collector of the JVM is configured with a maximum heap size of 30GB and a new generation size of 10GB. Parallel garbage collection is performed in the old and new generation.

For the data in Figure 2 we removed the maximum and minimum from the seven runs and calculated the average of the remaining five runs. On the x-axis are TLAB sizes of increasing but fixed size, except where it says “growing” indicating that the TLAB-growing strategy of the unmodified Hotspot JVM is used. Note that with the TLAB-growing strategy the TLAB size settles at around 2MB. On the y-axis the speedup over the corresponding TLAB-only configurations of the unmodified Hotspot JVM (baselines) is depicted. In Figure 2(a) the results using HABs with PLABs and in Figure 2(b) the results using HABs with CLABs are depicted. For smaller TLABs a higher speedup can be achieved since the slow-path is triggered more often. However, performance also improves with the TLAB-growing strategy. In the presented results CLABs perform on average slightly better then PLABs.

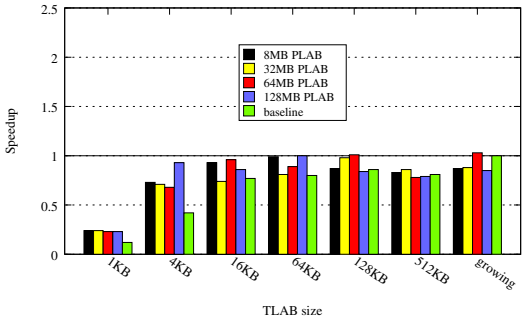


(a) PLABs

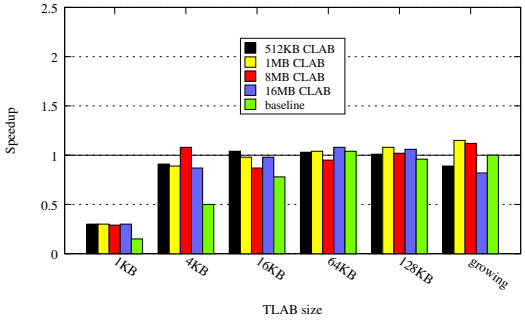


(b) CLABs

Figure 2. Speedup of using HABs with different TLAB and PLAB/CLAB configurations over the corresponding TLAB-only configurations of the unmodified Hotspot JVM.



(a) PLABs



(b) CLABs

Figure 3. The data of Figure 2 but using the unmodified Hotspot JVM with the TLAB-growing strategy as common baseline.

 ISBN: 978-1-61208-184-7

	PLAB				
TLAB	4MB	8MB	16MB	32MB	64MB
No	0.32%	0.33%	0.51%	0.94%	1.93%
1KB	0.11%	0.06%	0.09%	0.05%	0.15%
4KB	0.11%	0.06%	0.06%	0.06%	0.19%
8KB	0.13%	0.10%	0.08%	0.10%	0.29%
16KB	0.10%	0.08%	0.11%	0.08%	0.12%
32KB	0.11%	0.08%	0.08%	0.11%	0.30%
64KB	0.22%	0.15%	0.16%	0.29%	0.45%
128KB	0.52%	0.47%	0.63%	1.31%	2.60%
growing	0.55%	0.47%	0.64%	1.10%	2.23%

Table I
PERCENTAGE OF FOREIGN ALLOCATIONS WITH DIFFERENT PLAB AND TLAB CONFIGURATIONS.

	CLAB			
TLAB	512KB	1MB	8MB	16MB
1KB	0.00%	0.01%	0.05%	0.07%
4KB	0.01%	0.01%	0.06%	0.15%
16KB	0.01%	0.02%	0.07%	0.13%
64KB	0.03%	0.03%	0.10%	0.17%
128KB	0.06%	0.05%	0.15%	0.26%
growing	-	-	1.42%	2.81%

Table II
PERCENTAGE OF FOREIGN ALLOCATIONS WITH DIFFERENT CLAB AND TLAB CONFIGURATIONS.

Figure 3 is based on the same data as presented in Figure 2 but the y-axis depicts the speedup over the TLAB-only configuration of the unmodified Hotspot JVM using the TLAB-growing strategy, which is the default setting of the JVM. The results confirm that for smaller TLAB sizes than with the TLAB-growing strategy (around 2MB) similar or even better performance can be achieved. Figure 3(a) shows the results using HABs with PLABs and Figure 3(b) shows the results using HABs with CLABs. In particular, the results show that HABs with a small TLAB size provide similar performance as the original TLAB-growing strategy of the Hotspot JVM. In this case, a statically configured HAB implementation may thus replace a significantly more complex implementation of a TLAB-growing strategy.

In Table I and Table II, the amount of foreign allocations using different PLAB and CLAB configurations with different TLAB sizes are presented. The amount of foreign allocations increases with increasing PLAB or CLAB size. Overall, however, the amount of foreign allocations is low, which shows that allowing allocations in PLABs or CLABs that do not match the current processor or core the thread is running on can be tolerated here. For the 512KB and 1MB CLAB sizes the TLAB-growing strategy immediately determines to use TLABs larger then the given CLAB size, so CLABs are not used at all. Evaluating different synchronization strategies and allocations policies remains future work.

VI. CONCLUSION

We introduced hierarchical allocation buffers (HABs) for improving performance and scalability of memory management on state-of-the-art multiprocessor and multicore server machines. We implemented and evaluated three-level HABs in the Hotspot JVM and showed performance improvements in a memory-allocation-intensive benchmark due to better cache utilization and less contention on the global heap. The results show that taking the underlying hardware architecture of general purpose machines into account even more than before may require significantly less complex code than architecture-oblivious solutions without a loss in performance. The concept of HABs is just a first step. In the future we plan to consider a cooperative thread scheduling mechanism for executing threads that share a significant amount of data on the same processor to further benefit from caching. Moreover, we plan to integrate the HABs architecture into the TLAB-growing strategy of the Hotspot JVM for tuning not only TLAB sizes but also PLAB or CLAB sizes automatically. Considering other HABs configurations may also be beneficial, e.g., a four-level system with TLABs, CLABs, PLABs, and the global heap.

Acknowledgements

This work has been supported by the EU ArtistDesign Network of Excellence on Embedded Systems Design and the National Research Network RiSE on Rigorous Systems Engineering (Austrian Science Fund S11404-N23).

REFERENCES

[1] E. Berger, K. McKinley, R. Blumofe, and P. Wilson. Hoard: a scalable memory allocator for multithreaded applications. In *Proc. ASPLOS*, pages 117–128. ACM, 2000.

[2] D. Dice and A. Garthwaite. Mostly lock-free malloc. In *Proc. ISMM*, pages 163–174. ACM, 2002.

[3] A. Garthwaite, D. Dice, and D. White. Supporting per-processor local-allocation buffers using lightweight user-level preemption notification. In *Proc. VEE*, pages 24–34, New York, NY, USA, 2005. ACM.

[4] R. Hudson, B. Saha, A. Adl-Tabatabai, and B. Hertzberg. Mcrt-malloc: a scalable transactional memory allocator. In *Proc. ISMM*, pages 74–83. ACM, 2006.

[5] SPEC. SPECjvm2008 benchmarks, 2008. http://www.spec.org/jvm2008.

[6] Sun Microsystems. Description of HotSpot GCs: Memory Management in the Java HotSpot Virtual Machine White Paper. http://java.sun.com/j2se/reference/whitepapers/memorymanagement_whitepaper.pdf, 2006.

[7] Sun Microsystems. The Java Hotspot Virtual Machine White Paper. http://java.sun.com/products/hotspot/docs/whitepaper/Java_HotSpot_WP_Final_4_30_01.html, 2011.

 ISBN: 978-1-61208-184-7

Intelligent Processing of Video Streams for Visual Customer Behavior Analysis

Johannes Kröckel, Freimut Bodendorf
Institute of Information Systems
University of Erlangen-Nuremberg
Nuremberg, Germany
johannes.kroeckel@wiso.uni-erlangen.de, bodendorf@wiso.uni-erlangen.de

***Abstract* - In today's society purchasing goods through web shops has become habitual. Some years ago only a few products like books, computer games and music CDs were intensely sold by online retailers. Today's internet shops are offering almost every imaginable product and service. This also leads to an increasing competition for traditional retailers offering products in stationary retail stores. Losing more and more customers stationary retailers need to think of new approaches for customer retention. Since customer retention is based on knowledge about the customers and their behavior store managers have to come up with new concepts for gaining and using customer knowledge to compete with bargain prices and 24/7 availability. In order to gain this knowledge without using vague customer surveys or short-time observations an automated solution is desirable. In this paper an approach is introduced, which allows to track and analyze customer movements through the store. Person tracking is accomplished by using aerial mounted cameras and a set of computer vision algorithms. Based on the captured movement data customer behavior analysis is performed by applying the dbscan algorithm and Markov models. The approach is illustrated by a test environment showing considerable differences in customer behavior for two settings.**

Keywords - Customer tracking; video analysis; behavior analysis; retail; point of sale.

I. INTRODUCTION

Low prices, short delivery periods and 24/7 availability are only three of the greatest advantages of web shops over stationary retailers. Moreover, highly exchangeable products like books need no physical experience and therefore lack reasons for buying them in a stationary shop. Consequently, stationary shop operators need to come up with sophisticated, individual approaches for attracting and retaining customers. This requires knowledge about the customers, their actual context as well as their on-site buying behavior.

Stationary retailers lack sources of information about their customers and their individual context, while click paths, bounce rates and page impressions as well as time spent on websites are common key figures for internet shops [1-3]. Sales figures and product combinations recorded by electronic checkout counters don't reveal individual customer behavior. Approaches like the one described by Underhill [4] try to overcome this knowledge shortage by manually conducted observations. However, these strategies are limited. Continuous observation over a longer period of time like weeks or months require too expensive human resources. Besides that, an objective documentation of results by the observing persons cannot be guaranteed.

The approach presented in this paper aims at detecting customer behavior patterns by analyzing movements of customers within retail environments. Therefore, customer movement data is extracted by cameras recording raw data and computer vision algorithms extracting movement information. Subsequently, the discrete movement datasets are analyzed regarding frequently attended areas and the customers' movements between them. Finally, the hotspots and movements between them are used for customer behavior analysis.

II. RELATED WORK

Extraction and analysis of location data is strongly discussed in the field of ubiquitous computing (see [5-7]). The approaches described in those papers apply cell phone compatible technologies like GSM or GPS for location tracking in outdoor environments. The presented approach requires position determination inside a store. That means, it has to be more accurate than GSM and in contrast to GPS available indoor.

Data recorded by GSM and GPS are mainly used for location based services which are a surplus for the service users but not for companies. The approach presented here uses historic customer movements in order to gain valuable insights into customer behavior and to predict future actions. Thus, it is especially interesting for retail managers.

 ISBN: 978-1-61208-184-7

The prediction and analysis of movement data derived from GPS is among others addressed by Stauffer and Grimson [8], Andrienko et al. [9] or Ashbrock and Starner [10]. The presented approach is used to predict peoples' movements by location data collected from mobile devices. Highly frequented places are extracted and matched with well-known points of interest in their surrounding area. Prediction is based on first-order Markov models. Gutjahr [11] extends the work of Ashbrock and Starner [10] by a greater variety of methods for location capturing. These authors consider static points of interest like buildings or squares. However, due to continuing structural changes (e.g., bargain bins, seasonal products) such approaches cannot be applied to retail environments. Points of interest are only recognized during a limited period of time. In addition, there is no consideration of behavioral information that can be revealed from movement patterns.

III. MOVEMENT TRACKING

Person detection and tracking approaches are part of the field of computer vision. Algorithms are among others proposed by Bischop [12], Wang et al. [13] and Perl [14]. Fields of application are mainly the surveillance of places and facilities. Little attention is being paid to customer behavior within retail environments yet. The overall concept presented in this work is inspired by Fillbrandt [15]. In his doctoral thesis he introduces an approach for a modular single or multi camera system tracking human movements in a well-known environment. Fillbrandt's approach is applied for tracking people in airplanes. Persons are detected on images by a set of computer vision algorithms and position estimation is executed.

Camera based person tracking can be accomplished by two similar settings. The lateral approach uses cameras being mounted edgewise [16] whereas the aerial approach utilizes cameras mounted on the ceiling. While lateral mounted cameras enable the observation of larger areas, approaches using aerial mounted cameras reveal more accurate results. Therefore, an aerial approach is chosen.

By using aerial mounted cameras the size of the observed area depends on the camera's focal distance and altitude. For the detection and tracking of persons within a dedicated area a set of algorithms is applied. First, background differencing (among others described by Piccardi [17] and Yoshida [18]) is used for object detection in single frames being captured by a camera.

In a first step, a reference image is needed which shows the captured areas without any objects. Comparing the reference image with the actual considered frame excludes all similarities between the two pictures. Differences are highlighted (see Fig. 1, upper right area). After that, image noise is reduced. Eliminating objects that are smaller than the average human shape reveals all objects that could be considered as persons. This step is mostly accomplished by using a template or contour matching algorithm as described by Hu [19] or Zhang and Burckhardt [20]. However, this is not feasible for the presented approach. Contour or template matching algorithms are not able to detect human shapes with high reliability as a result of the varying distance and view of the camera. Besides that, people carrying bags or driving shopping carts as well as disabled people using wheel chairs would not be recognized as humans by the algorithm. Therefore, the detected shapes are filtered by a minimum area.

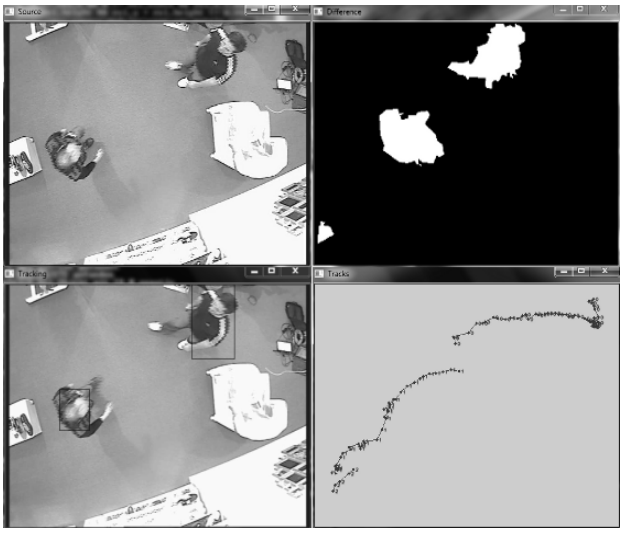


Figure 1. Aerial person tracking

This leads to significantly better results, i. e., persons can be recognized correctly in most cases.

Subsequently, the continuously adaptive mean shift (camshift) algorithm presented by Bradski [21] is applied for tracking the detected persons. The algorithm is based on the mean-shift algorithm originally introduced by Fukunaga and Hostetler [22]. The approach was originally invented for face tracking. Thenceforth, it has been applied for a great variety of tracking purposes.

The mean-shift algorithm is used to track motions of objects by iteratively computing the center of mass of the HUV (hue, saturation, value) vectors within a defined window [23]. For every frame of a video stream the centers of mass are calculated and consequently defined as new centers of the corresponding windows (see Fig. 2). By connecting subsequently occurring centers of windows a trajectory of the movement is obtained. Defining windows as smallest rectangle areas covering shapes of persons extracted by the background differencing approach enables to apply this concept for person tracking purposes.

While the mean-shift algorithm considers windows of static size, the camshift implementation adapts the

 ISBN: 978-1-61208-184-7

window size dynamically. This is especially important for the presented application because persons moving away from or to the center of the observed area occur in different sizes. Using the mean-shift algorithm would lead to an increasing amount of vectors from areas around the considered person. If the amount of these vectors becomes too high, the scope on the person will be lost and errors occur.

To come to better results, especially for crowded places the good features to track algorithm by Shi and Tomasi [24] and the optical flow algorithm by Lucas and Kanade [25] are applied. The good features to track algorithm uses corner detection to find pixels, which differ from those in their surrounding area.

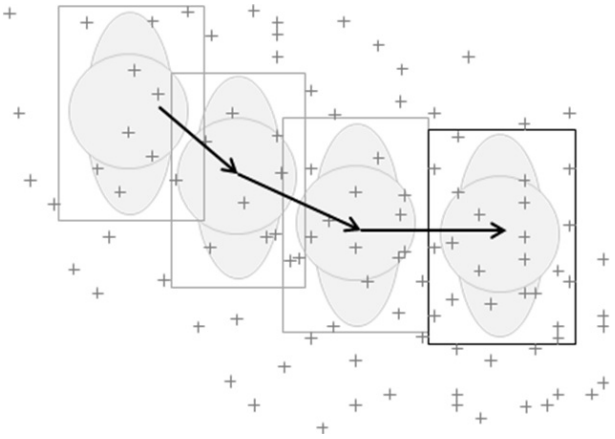


Figure 2. Mean-shift: window shifts

The optical flow algorithm compares sequential images regarding noticeable changes. So, prominent pixels are obtained from the good features to track algorithm. Subsequently, the algorithm tries to find these pixels in the following frame within the surrounding area of their original location on the image.

The combination with a color constancy algorithm (e.g., Barrera et al. [26]) enables to track persons across several cameras and therefore several corridors. This implies that persons leaving one camera area to another one have to be handed over while crossing an overlapping area (see Fig. 3).

Due to the fish eye effect of the camera's lens especially the locations of persons being further away from the camera are perspectively distorted. That means, they cannot be used for true to scale calculations yet. In consequence a perspective transformation by calculating a 3x3 warp matrix based on four source and four destination points is executed. The points have to mark equal positions on the image and on a true to scale map to calculate the factor of distortion. An evaluation study was performed for a test environment including one corridor and two shelves. The corridor between these two shelves has a width of 2.0 m and a length of 4.0 m. The width compares with the typical scales of a small grocery store. The camera is placed 3.0 m aboveground. The sample footage consists of 16,000 frames showing 30 different people with a maximum of three people walking through the observed area at the same time. Lossless tracking is obtained for ~82% of the observed walks. The average deviation between the real and the automatically determined position is 0.11 m.

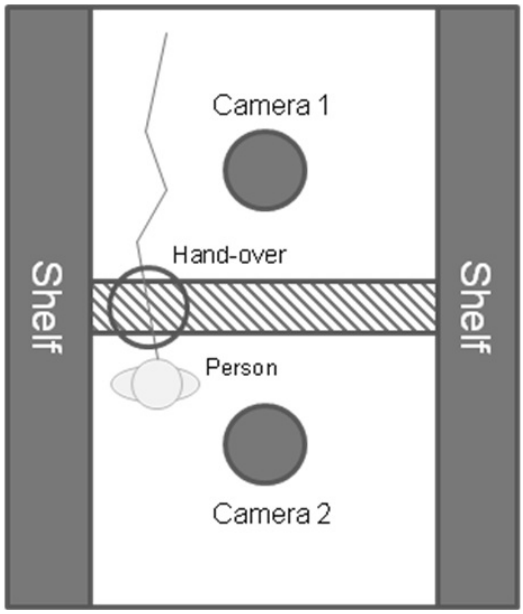


Figure 3. Camera hand-over

IV. MOVEMENT ANALYSIS

A. DBScan

The algorithm called 'density-based spatial clustering of applications with noise' originally proposed by Ether et al. [27] was developed to distinguish between clusters and noise in spatial databases. Clusters are defined as areas with a considerable higher density than outside of the cluster. To distinguish clusters from noise the following steps have to be accomplished. First, an arbitrary point p is selected. All points that can be reached from p are retrieved. If p turns out to be a core point of a cluster a new cluster is formed. Limitations are made regarding the minimum points (minPts) to be reached by p as well as the distance between p and the considered neighboring points. If one of the constraints is not met no new cluster is formed and another point is considered.

The overall datasets of all trajectories extracted by the movement tracking approach are analyzed by the dbscan algorithm using a minimum threshold (minPts) of 5 points

 ISBN: 978-1-61208-184-7

and a maximum real world distance of 0.02 m. The analysis reveals an amount of 551 clusters.

For further processing all clusters including points from less than 60% of the trajectories are eliminated. This reveals 3 clusters comprising points of between 60% and 90.5% of the trajectories within the test environment (see Fig. 4). The areas covered by the clusters are considered as hotspots that are significantly higher visited than other areas of the test retail environment.

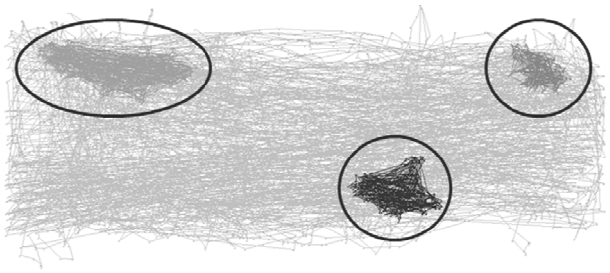


Figure 4. Clusters revealed by DBScan algorithm

B. Markov Chains

A first-order Markov model includes states of a system as well as transition probabilities between them [28]. A transition probability is defined as the probability of a system change from one state to another one. In the presented approach probabilities describe the chances of moves between two clusters. Recursive transitions are neglected because for the presented approach only the succession of movements between different states (i.e. clusters) is relevant. That means, a transition between two hotspot clusters exists when two temporally succeeding points of a customer trajectory belong to two different clusters. The points do not have to be temporally succeeding points in the database but all of the intermediate points must not be part of another hotspot.

Regarding the movements between clusters, the datasets resulting from the computer vision algorithms described in section III are taken into account. Points that are not part of one of the three considered clusters are ignored.

V. BEHAVIOR ANALYSIS

Considering the clusters and the movements between them allows a closer look on how customers act within a retail store. As use cases two shopping scenarios within a prototypical retail environment have been analyzed. The test environment comprises eight product categories (see Fig. 5).

The first scenario describes a regular product setup without any advertisements and signs and therefore observes the regular behavior of customers. The second one is based on the findings of the first scenario and includes advertisements for selected products. Both of the scenarios are compared eventually.

For the first scenario three clusters exceeding the 60% threshold are found. One of them covers the area with shelves containing dairy products. The second, smaller one is located near shelves with crisps ad chocolate. The third one covers the area in front of the shelves with consumer electronics.

Fig. 5 shows these three clusters as well as the transitions between them. The percentage values describe the percentage of transitions been made between two clusters compared to the total transitions. Transition paths below the limit of a 5% share are ignored.

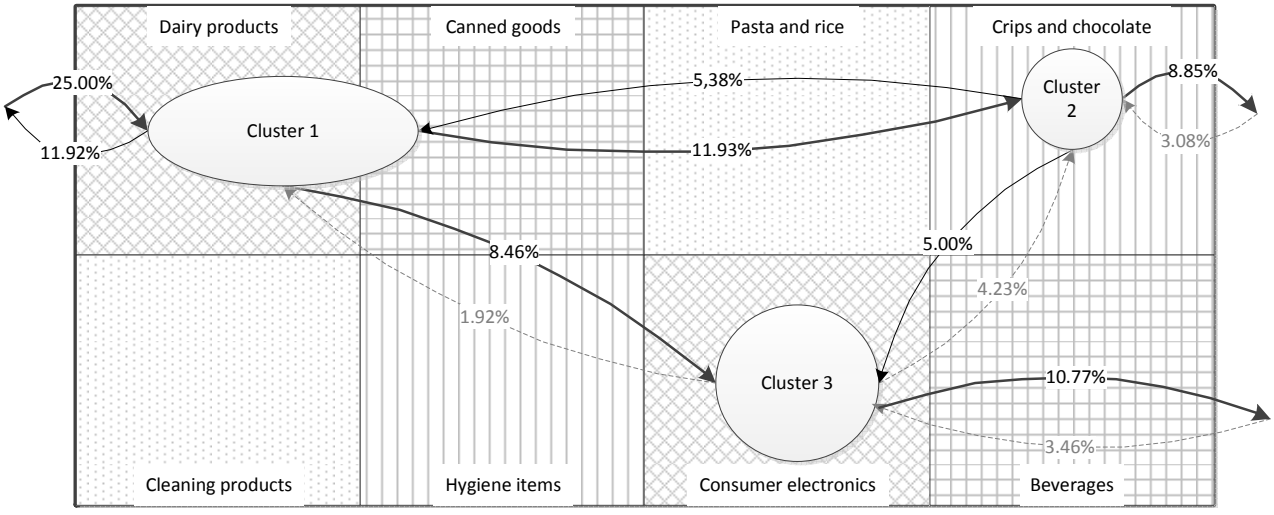


Figure 5. Prototypical retail environment – Scenario 1

 ISBN: 978-1-61208-184-7

For the given scenario the majority of customers enter the corridor from the left side heading for the first hotspot (dairy products). Afterwards they are more likely moving on to the second one (crisps and chocolate). Then, either they go back to the area of cluster 1 (dairy products) or go on to the one of cluster 3 (consumer electronics). After that, the customers are most likely leaving the observed area. Besides showing hotspots within the retail environment the graph of Fig. 5 also reveals typical paths customers use to move through the store. Looking at visited products it is apparent that products located on the lower left (cleaning products and hygiene items) are less of advertisements near frequently used paths to call attention for these products.

This idea is seized for the second scenario (see Fig. 6.). The prototypical retail store is extended by two promotional signs for cleaning and hygiene products. This leads to notable changes of the customers' behavior. While the first scenario leads to three hotspots the second one includes four hotspots. An additional hotspot covers the area between cleaning and hygiene products.

Considering the transitions customers still enter the observed retail environment most likely from the left side attending the area near dairy products first. Consequently, they are most likely moving on either to crisps and chocolate or the area in front of the shelves containing consumer electronics.

While most of the consumers move from crisps and chocolate back to the area of dairy products there is also a notable percentage of customers walking to an area in front of cleaning and hygiene products. This could mean that the promotion campaign was successful. But this approach cannot only be used for advertisement evaluation. Furthermore, it is possible to evaluate the entire structure of a retail environment regarding product placement or shelf structure.

VI. CONCLUSION

This paper introduces an approach for the analysis of customer behavior on the basis of video recordings of customer movements within a real-world shopping environment. Surveillance cameras produce large amounts of video data. Intelligent methods for processing this data are crucial in order to gain customer insight especially over a longer period of time. Therefore, a method for capturing and extracting movements using network cameras and algorithms from the field of computer vision is provided. The resulting data is used for customer behavior analysis. Analysis is done using the dbscan algorithm. Finally, the extracted clusters and the movements of customers between them are represented as Markov chains. Both, the cluster areas and the transitions are used to assess the quality of the retail environment. The transitions reveal possibilities for marketing campaigns considering highly frequented paths. On the one hand, this enables retailers to advertise products with locations away from the main paths as described in the preceding section. On the other hand, retailers can promote additional products located near previously visited hotspots. Moreover, based on the tracked trajectories next customer movements can be predicted and individualized messages posted on advertisement screens in real-time.

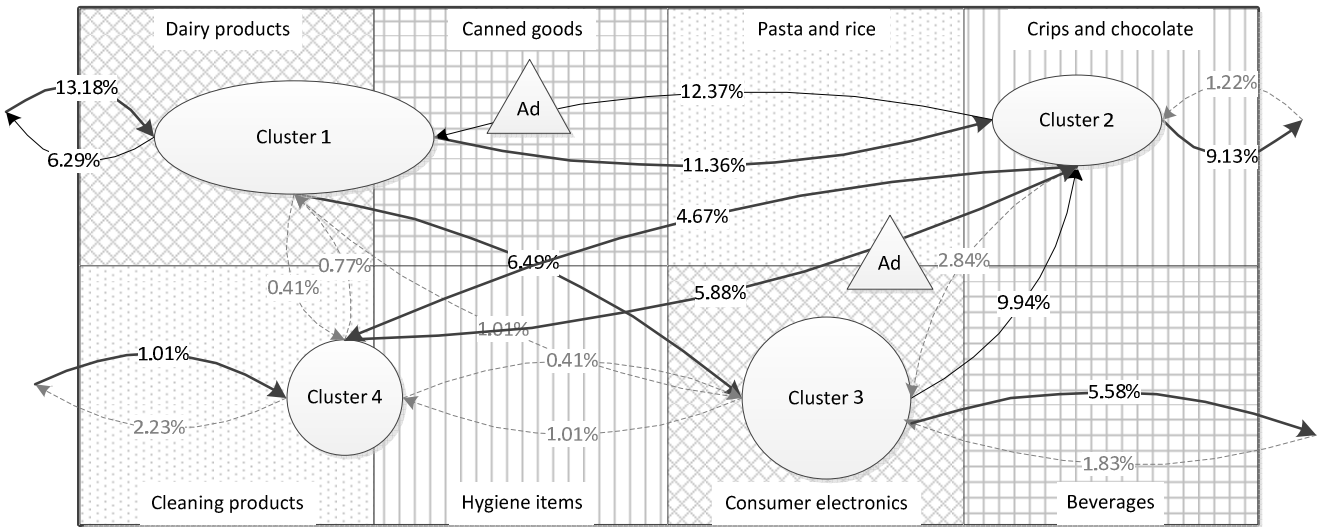


Figure 6. Prototypical retail environment – Scenario 2

Considering longer periods of time might reveal different customer behavior not only for different setups but also for different times of a day, for different days of the week or for different seasons. In addition to that, comparative studies of different stores of the same chain are possible. Information gained from these analyses is the basis for planning dynamic product placements or seasonal offers. This knowledge about customers is also an additional source for management information systems. It helps to identify rarely visited areas or products and therefore enables retail store managers to analyze and optimize their shopping environment. New settings can be evaluated by considering the ex-ante and the post change status.

REFERENCES

[1] H.-F. Li, S.-Y. Lee, and M.-K. Shan, "DSM-TKP: Mining Top-K Path Traversal Patterns over Web Click-Streams," Proc. 2005 IEEEWICACM International Conference on Web Intelligence WI05, 2005, pp. 326-329.

[2] I. Nagy and C. Gaspar-Papanek, "User Behavior Analysis Based on Time Spent on Web Pages," in Web Mining Applications in Ecommerce and Eservices, Springer Berlin / Heidelberg, 2009, pp. 117-136.

[3] A. Goldfarb, "Analyzing Website Choice Using Clickstream Data," Advances in Applied Microeconomics A Research Annual, vol. 11, 2001, pp. 26.

[4] P. Underhill, Why we buy: The Science of Shopping. Textere, 2000.

[5] C. Feature, "Systems for Ubiquitous Computing," Computer, vol. 34, August 2001, pp. 57-66, 2001.

[6] C. Becker and F. Dürr, "On location models for ubiquitous computing," Personal and Ubiquitous Computing, vol. 9, no. 1, 2004, pp. 20-31.

[7] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," Computer, vol. 34, no. 8, 2001, pp. 57-66.

[8] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, 2000, pp. 747-757.

[9] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti, "Interactive visual clustering of large collections of trajectories," Proc. IEEE Symposium on Visual Analytics Science and Technology, no. ii, 2009, pp. 3-10.

[10] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," Personal and Ubiquitous Computing, vol. 7, no. 5, 2003, pp. 275-286.

[11] A. Gutjahr, "Bewegungsprofile und -vorhersage," 2008.

[12] H. Bischof, "Robust Person Detection for Surveillance Using Online Learning," Proc. of the Ninth International Workshop on Image Analysis for Multimedia Interactive Services, 2008, p. 1.

[13] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," Pattern Recognition, vol. 36, no. 3, 2003, pp. 585-601.

[14] J. Perl, "A neural network approach to movement pattern analysis," Human Movement Science, vol. 23, no. 5, 2004, pp. 605-620.

[15] H. Fillbrandt, "Videobasiertes Multi-Personentracking in komplexen Innenräumen," Rheinisch-Westfälische Technische Hochschule Aachen, 2008.

[16] J. Kröckel and F. Bodendorf, "Extraction and Application of Person Trajectories in Retail Environments," Proc. of the IADIS International Conference - Intelligent Systems and Agents, 2010, pp. 109-113.

[17] M. Piccardi, "Background subtraction techniques: a review," Proc. IEEE International Conference on Systems Man and Cybernetics IEEE, vol. 4, no. C, 2004, pp. 3099-3104.

[18] T. Yoshida, "Background differencing technique for image segmentation based on the status of reference pixels," Proc. International Conference on Image Processing, ICIP '04., vol. 1, no. 1, 2004, pp. 3487-3490.

[19] M.-K. Hu, "Visual pattern recognition by moment invariants," IEEE Trans Information Theory, vol. 8, no. 2, 1962, pp. 179-187.

[20] W. Zhang, C. K. Chang, H.-i Yang, and H.-yi Jiang, "A Hybrid Approach to Data Clustering Analysis with K-means and Enhanced Ant-based Template Mechanism", 2010, vol. 1.

[21] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," Interface, vol. 2, no. 2, 1998, pp. 12–21.

[22] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," IEEE Transactions on Information Theory, vol. 21, no. 1, 1975, pp. 32-40.

[23] D. Comaniciu and P. Meer, "Mean shift analysis and applications," Proc. of the Seventh IEEE International Conference on Computer Vision, vol. 2, no. 2, 1999, pp. 1197-1203.

[24] J. Shi and C. Tomasi, "Good features to track," Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 94, 1994, pp. 593-600.

[25] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proc. International Joint Conference on Artificial Intelligence, vol. 3, 1981, pp. 674-679.

[26] P. Barrera, J. M. Canas, and V. Matellán, "Visual object tracking in 3D with color based particle filter," International Journal of Information Technology, vol. 2, no. 1, pp. 61–65, 2005.

[27] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Proc. of the 2nd International Conference on Knowledge Discovery and Data mining, vol. 1996, pp. 226–231.

[28] A. A. Markov, "Extension of the limit theorems of probability theory to a sum of variables connected in a chain (Reprint in Appendix B)," John Wiley and Sons, 1971.

 ISBN: 978-1-61208-184-7

A Cellular Automata Model for Wireless Sensor Networks

Yijun Wang, Zhihong Qian, Dayang Sun, Ce Zhou
College of Communication Engineering
Jilin University
Changchun, China
e-mails:wyjs-107@163.com,dr.qzh@163.com,www.sunday@gmail.com,z-prayer@163.com

***Abstract*—Modeling for wireless sensor networks is very challenging because the modeling needs to adapt network dynamics and find out multiple optimizing paths from the microscopic point of view. In this paper, we propose a cellular automata model that focuses on dynamic network topology, multipath data transmission mechanism and energy overhead. Each node in a network is represented by a cell, and any permutations and combinations that represent any links between two cells constitute the cellular space. A wireless sensor network is modeled by related cell states and cell evolution rule. A cell transmission model is derived for multipath transmission of information. All these aim to ensure network reliability with the minimum resource requirements. Presented analytical work is proved validly by simulations.**

Keywords- Cellular Automata; Microscopic mode; Modeling; Wireless Sensor Networks.

I. INTRODUCTION

With rapid development of the Internet of Things (IOTs) [1], Wireless Sensor Networks (WSNs) [2] play an more and more pivotal role in bridging the gap between the physical and virtual worlds, which can enable things to respond to changes in their physical environment. Therefore, with WSNs technology improving, a new model that could adapt various application environments is needed to describe characteristics and goals of WSNs exactly. Concerning IOTs, the model must be applied to a wide range of WSNs.

Tarik *et al.*, [3] analyze some methods about WSNs modeling, and currently, these models have been explored with different features. It examines this emerging field to classify wireless micro-sensor networks according to different communication functions, data delivery models, and network dynamics. A service-centric model focuses on services provided by a WSN and views a WSN as a service provider [4]. The service-centric model only provides a holistic approach to measuring and presenting WSNs effectiveness. There is no description from the microscopic point of view in this model. The methodology for the modeling and the worst-case dimensioning of cluster-tree WSNs shows the fundamental performance limits of cluster-tree WSNs [5]. Although it presents a general and flexible framework, the node function in such model is hierarchical. The Cluster Head nodes that are used for transmitting data packets will consume more energy. Then nodes inequality will reduce the network life systematically. Yet some routing protocols use another model that is based on data-centric [6], and this model depends on data identifiers and specified locations, therefore, it isn't appropriate to the dynamic and randomly-deployed WSNs, and it have no strong convergence.

Cellular automata (CA) is the dynamical system that evolves in the discrete time dimension according to some local rules, which is essentially defined in a cell space constituted of cells with discrete and finite state [7][8]. The composition of cellular automata is shown in Fig. 1.

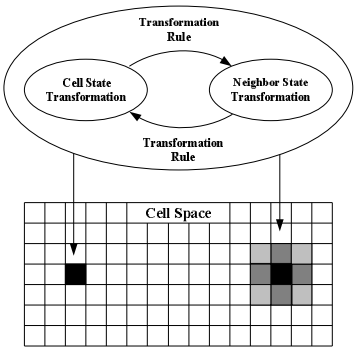


Figure. 1 Cellular automata schematic diagram

In this paper, a cellular automata modeling view is proposed for WSNs. Our basic goal and idea are to ensure WSNs continuing and effective work by using some simple parameters, connection and operation rules, and finally simulate complex and rich applications of WSNs. Such a model does not only guarantee equality of nodes within the network, and ensure the energy balance distribution, but also guarantee strong convergence under conditions of dynamic network topology. It provides a flexible multipath data transmission mechanism to ensure network reliability. Just as importantly, the cellular automata model describes WSNs characteristics from the microscopic point of view.

The remaining organization of the paper is as follows. Section II presents the cellular automata model, which includes network model and data traffic model. Section III provides a simulation analysis based on this cellular automata model. Finally, Section IV concludes the paper and proposes the future work.

II. THE CELLULAR AUTOMATA MODEL FOR WSNS

WSNs have many challenges compared with traditional wireless networks, and communication in WSNs differs from

 ISBN: 978-1-61208-184-7

that in other types of networks. More specifically, WSNs energy is often limited since it is impossible to recharge sensor nodes if the networks are deployed in the uninhabited areas. This paper uses the following models to evaluate the performance of WSNs, including the dynamic network topology and data transmission reliability, and finally realize energy saving and prolong networks lifetime, which could keep the model practicality and simplicity.

A. Network model

In order to describe the model better, the following definitions are given firstly.

Definition 1: The set of all the nodes $C = \{c_1, c_2, \dots, c_n\}$, Then any permutations and combinations L in set C constitute the cellular space, where $L = \{Lk = (c_1, \dots, c_i, \dots, c_j, \dots c_n) | c_i, c_j \in C, c_i \neq c_j, i, j = 1, 2, \dots, n, k \in Z\}$, and each node is a cell (The cellular space is a two-dimensional grid).

Definition 2: Let S represents limited and discrete states set of cells, where $S = \{s_1, s_2, s_3\}$, s_1: information transmission state, s_2: wait state, s_3: idle state, and three states represent three types of cells respectively: center cell(CC), neighbor cell(NC) and idle cell(IC).

Definition 3: Let N represents cell link neighborhood, where $N = \{Lk^* | difference(Lk_1 - Lk_2) \leq d, Lk_1, Lk_2 \in Lk\}$, $difference(Lk_1 - Lk_2)$ is the difference between two permutations and combinations, d is the degree of difference.

Fig. 2 shows the relationship diagram of cell parameters. Cell [52] has eight cell neighbors, including cell [44], cell [45], cell [46], cell [51], cell [53], cell [58], cell [59], cell [60]. When cell [52] requires communication with cell [0], the state of cell [52] is s_1, its neighbors' states are s_2, and cells out of this local space keep idle state. As shown in Fig. 2, the difference of two links Lk_1 and Lk_2 between cell [44] and cell [0] represent link neighborhood.

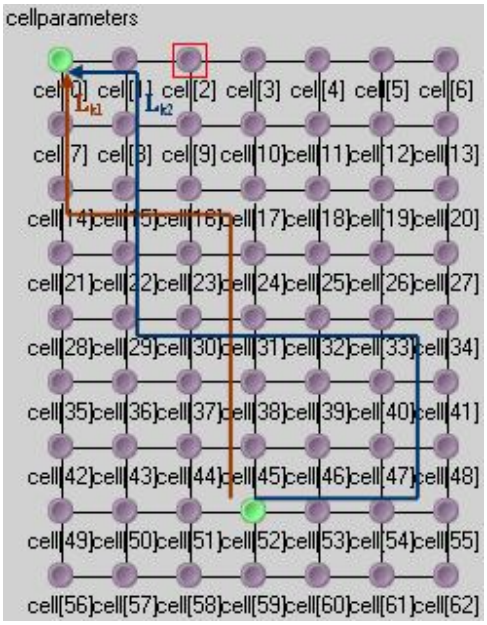


Figure. 2 The relationship diagram of cell parameters

Based on above definitions, the paper uses the following rule to update model in parallel.

Rules $F : S_t \rightarrow S_{t+1}$:

Step 1: Initialize the parameters and define cell state, $C = \{c_1, c_2, \dots, c_n\}$, $S = \{s_1, s_2, s_3\}$.

Step 2: Assuming that I_n is the number of idle cells, N_n is the number of neighbor cells, then determine the next hop link by judging the states of different types of cell.

Case (1): If CC has I_n idle cells(ICs) around($I_n \leq 6$), it forwards the data packets to the next hop by cell transmission model(described in the next section), which uses an IC.

Case (2): If CC has no ICs but has N_n neighbor cell(NCs) around, it forwards the data packets to the next hop by cell transmission model, which uses a NC.

Case (3): If CC has no ICs and NCs around, that is, its neighbor cells are all in information transmission state, then this CC stores the data packets into the cache, and once it receives cell release message, it chooses this cell to forward the data packets.

Step 3: With the dynamic changes in topology, establish a number of transmission links, and use $N = \{Lk^* | difference(Lk_1 - Lk_2) \leq d, Lk_1, Lk_2 \in Lk\}$ to compare neighborhood of links, then choose the most effective link for related information transmission.

Step 4: Update link, store the multiple link information.

The network model is constructed by cellular automata. The aim of this cellular automata model is to group cell nodes that have similar processing needs into unit cell families and sink node that meet these needs into each cell in the network.

B. Cell dynamic rate

To verify convergence of the model in a dynamic environment better, this paper defines the concept of cell dynamic rate that is represented by δ_n, and using simulation tool OMNeT++, we identify the range of the dynamic rate δ_n. The dynamic rate is defined as follow:

Definition 4: the dynamic rate δ_n represents average moving rate of all the cells in cellular space. In addition, let δ_n express dynamic topological properties of cellular space.

OMNeT++ is an object-oriented modular discrete event network simulation framework. It has a generic architecture, OMNeT++ itself is not a simulator of anything concrete, but it rather provides infrastructure and tools for writing simulations.

One of the fundamental ingredients of this infrastructure is a component architecture for simulation models. We use wireless network simulator model by OMNeT++, where network protocol is IEEE 802.15.4. In this paper, we only consider the two-dimensional space situation, therefore, cells are laid out randomly in a 50m×50m two-dimensional area, and the cell network topology schematic diagram is illustrated in Fig. 3.

 ISBN: 978-1-61208-184-7

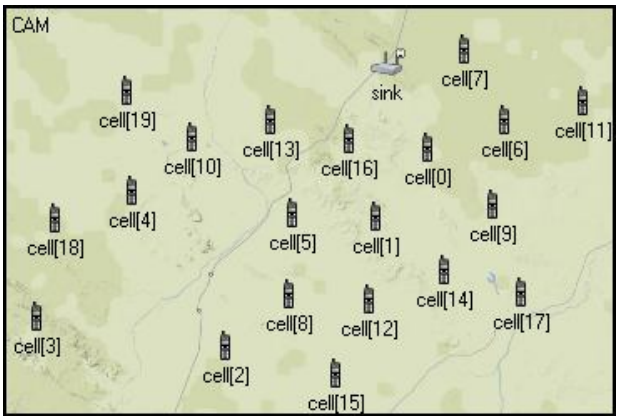


Figure. 3 The topology of mobile cells

The mobile model of cells is random waypoint mobility. The paper verifies the WSNs allowable dynamic range by simulation experiments, and δ_n is divided into four cases, as shown in Table I. When $\delta_\infty \geq 6m/s$, the network is hardly able to establish. However, When dynamic rate $\delta_n \leq 6m/s$, it is divided into three levels. According to different δ_n , we give networks convergence time step in different networks scales. From Fig. 4, as δ_n varying, the network set-up time of the cellular space is almost no differences in the same network size. Also with increasing of the network size, the network set-up time has increased a little. Therefore, we come to the conclusion that the cellular automata model in the dynamic case for WSNs has better convergence.

TABLE I. THE DEFINITION OF DYNAMIC RATE

Dynamic Rate	Cell Average Speed (m/s)	Whether Impact Network formation
δ_1	0 ~ 2.5	no
δ_2	2.6 ~ 4.4	no
δ_3	4.5 ~ 5.9	no
δ_∞	⩾6	Yes

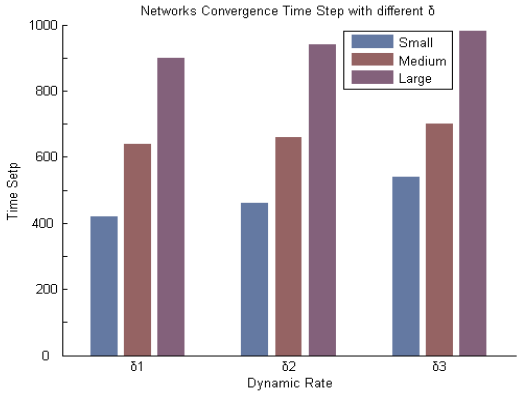


Figure. 4 Networks convergence time step

C. Cell transmission model

The following assumptions are adopted for simplifying the model:

- Communication radius of the center cell (CC) is its six neighbor cells.
- Except Sink cell, each cell has the same initial energy.
- Communication is symmetric.
- Each cell keeps static or movement according to the dynamic rate δ_n .

For facilitating the model description, we define the following variables:

f_l :data frame length;
l_c :channel length;
v_e:transmission rate of electromagnetic waves in the channel;
$v_i(t)$:data rate of the cell i at time t;
$D_{i,j}$:time delay, from cell i to cell j;
$B_{i,j}$:highest data rate in unit time that could be reached from cell i to cell j;
$q_{i,max}(t)$:max upstream data flow of the cell i at time t;
$h_{i,j}$:hops, from cell i to cell j;
$n_{i,j}$:the number of links, from cell i to cell j.

In order to describe flow relationship between CC and neighbor cells, firstly, the metrics delay that we consider mainly between two cells can be formulated as

$$D_{i,j} = d_t + d_p \quad (1)$$

where d_t is transmission delay, d_p is propagation delay. Typically, we have

$$d_t = f_l / v_i(t) \quad (2)$$

$$d_p = l_c / v_e \quad (3)$$

Secondly we consider another metrics delay-bandwidth product,

$$\tau = D_{i,j} \times B_{i,j} \quad (4)$$

It represents the number of bits that this cellular link could accommodate. Thirdly, the data flow direction is divided into upstream and downstream. Normally, most event-driven messages in WSNs are forwarded from individual cell to the sink in upstream direction, thus in this paper we focus on the max upstream data flow of cell i at time t , $q_{i,max}(t)$. Therefore, based on $v_i(t)$, τ and $q_{i,max}(t)$, we define maximum carrying capacity of cell i, N_i, and we have

$$N_i = balance\ \{ v_i(t), \tau , q_{i,max}(t)\} \quad (5)$$

Then each CC determines the next link by comparing N_i value among the neighbor cells.

We consider

$$Q_{i,j} = \sum_{i=1}^{h_{i,j}} N_i \quad (6)$$

it represents the total maximum carrying capacity of one link from cell i to cell j. Then the mean value of variable N_i can be calculated as

$$\vec{\mu} = E[\vec{N_i}] = (E[N_1], E[N_2], \cdots, E[N_{h_{i,j}}])^T \quad (7)$$

From (7), cell transmission equation of multipath selection is given by

$$P = \{ max\ \vec{\mu}(n)\ \ (n \leqslant n_{i,j}),\ min\ h_{i,j} \} \quad (8)$$

 ISBN: 978-1-61208-184-7

Given such a cell transmission model, we address multipath transmission of the information for WSNs compared with other forwarding strategy [9], and aim to ensure the network reliability with the minimum resource requirements.

III. SIMULATION RESULTS

Setting 60 sensor nodes and a sink node in an area of 50m×50m to certify the effectiveness of the network model, where the simulation setting parameters are shown as Table II.We use IEEE 802.15.4 as the Physical layer and Data link layer protocol. The maximum datagram size of each cell is 127-bytes.The maximum transmission distance that cells can be reached is 12m, and the network range is 50m×50m.In addition, data flow style is CBR.

TABLE II. SIMULATION PARAMETERS

Parameters	Set Value
PHY/MAC	IEEE 802.15.4
Maximum Datagram Size	127 bytes
Max Transmission distance	12m
Network Range	50m×50m
Data Flow	CBR, 100bytes

The simulation results is verified by three models, which is including service-centric model (SCM) [4], cluster model (CM) [5], and our cellular automata model (CAM). SCM focuses on services provided by a WSN and views a WSN as a service provider. A WSN is modeled at different levels of abstraction. For each level, a set of services and a set of metrics are defined. Services and their interfaces are defined in a formal way to facilitate automatic composition of services, and enable interoperability and multitasking of WSNs at the different levels. CM provide a fine model of the worst-case cluster-tree topology characterized by its depth, the maximum number of child routers and the maximum number of child nodes for each parent router.

The first experiment is the comparison of models conver- gence. We apply the classical LEACH (Low Energy Adaptive Clustering Hierarchy) [10] protocol to CM and CAM. Fig. 5 shows the simulation result of convergence, which is measured by the metrics of Energy*Delay [11] with changing of time step. Energy*Delay model, introduces a great energy-effective solution to the communication from source nodes to destination nodes and significantly simplifies the topology of networks. From the research and simulation results that described in [11], an significant effect on determining the increment of pheromones by minimizing the Energy*Delay model was obtained.

In the initial stage, there is a huge fluctuation about the metrics of Energy*Delay in the CAM, CM and SCM, this becauses the initiation of cellular automatas. After such processing, the Energy*Delay tends to a fixed value. As shown in Fig. 5 (a) and (b), CAM gradually converges to 10, however, CM gradually converge to 80. In addition, as we known that, SCM is a macro-model, then from Fig. 5 (c), we can find that SCM has a huge fluctuation almostly all the time by the metrics of Energy*Delay, so it has the worst convergence during three models. Therefore, it is concluded that CAM that use cellular automata scheme has better convergence performance than CM and SCM.

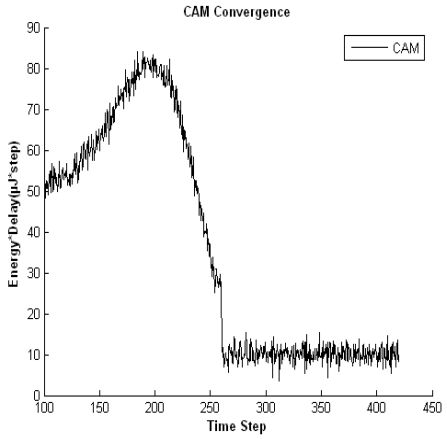


Figure. 5 (a) CAM Convergence

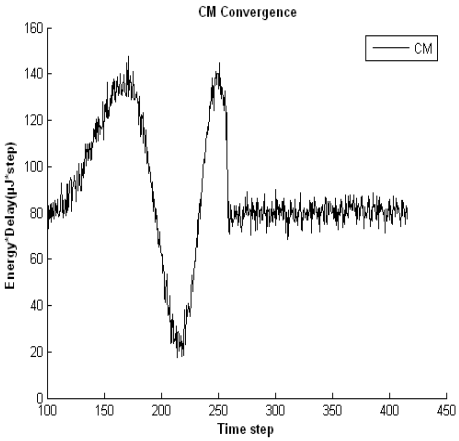


Figure. 5 (b) CM Convergence

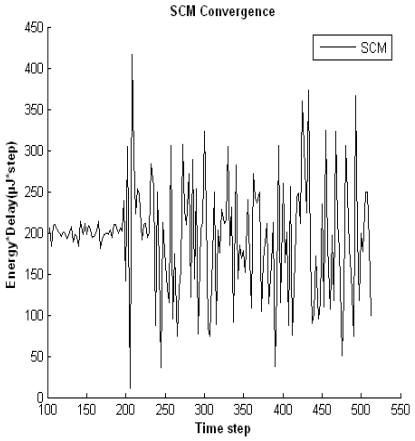


Figure. 5 (C) SCM Convergence

 ISBN: 978-1-61208-184-7

Secondly, we simulate the network on different models for comparing the number of paths between two cells. From Fig. 6, CAM take less time to forward the data packets through the network than the other two models. The reason is that CAM has the characteristic of multi-path transmission by cells auto selection method. And as shown in Fig. 6, althought CAM finds less paths in local time slot, the tendency of finding more paths about CAM is increasing step by step compared with the other two models.

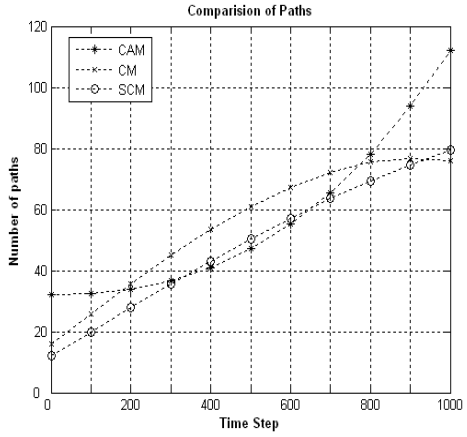


Figure. 6 Paths Comparison of Models

Finally, Fig. 7 presents the overhead results of the three models in dynamic environment described in Section II. A global analysis shows that CAM gives the best performance with the increasing of the dynamic rate δ_n, which is almost independent of δ_n. This is because that CAM uses the cellular evolution rules to optimize the network formation process, and decrease the number of messages exchange.

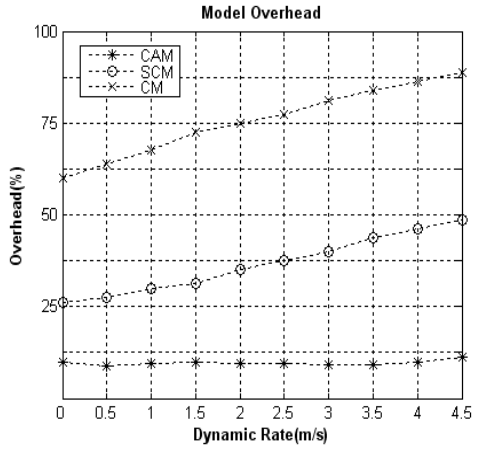


Figure. 7 Overhead Comparison of Models

All these simulation results ensure network reliability with the minimum resource requirements by the cellular automata model.

IV. CONCLUSION AND FUTURE WORK

In this paper, a cellular automata model is devised for WSNs application research. The key issues considered in this model are dynamic network topology struture, multipath data transmission, and WSNs network energy efficient. Simulation results by the comparison of convergence, multi-path transmission and overhead, verify the effectiveness of the related work.

However, there are much uncertainty for popularization of the IOTs, and many technical aspects based on WSNs that need to be broken through [12]. In the future work, we plan to apply this model to research WSNs routing algorithm and time synchronization strategy for solving WSNs localization problems [13] [14]. Moreover, these researches will provide a foundation that achieves the integration between WSNs and Internet of things.

ACKNOWLEDGMENT

The National Natural Science Foundation of China (No.60940010, No.61071073) and the Doctoral Fund of Ministry of Education of China (No. 20090061110043) support this work.

REFERENCES

[1] K. Matthias, H. Paul, S. Albrecht, “Embedded Interaction: Interacting with the Internet of Things”, IEEE Internet Computing, vol. 14, no. 2, pp. 46-53, Mar.-Apr. 2010.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A survey on sensor networks”, IEEE Communications Magazine, vol. 43, no. 8, pp. 102-114, 2002.

[3] S. Talik, N. B. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless microsensor network models”, SIGMOBILE Mobile Comput.Commun. Rev., vol. 6, no. 2, pp. 28–36, Apr. 2002.

[4] D. Gracanin, M. Eltoweissy, A. Wadaa, amd L. A. DaSilva,“A Service-Centric Model for Wireless Sensor Networks”, IEEE Journal on Selected Areas in Communications, vol. 23, no. 6, pp. 1159-1165, Jun. 2005.

[5] A. Koubaa, M. Alves, and E. Tovar, “Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks”, Proceedings of 27th IEEE International Real-Time Systems Symposium, pp. 412-421, 2006.

[6] J.Yang, M. Xu, W. Zhao, and B. G. Xu, “A Multipath Routing Protocol Based on Clustering and Ant Colony Optimization for Wireless Sensor Networks”, Sensor, vol. 10, no. 5, pp. 4521-4540, May. 2010.

[7] T. Toffoli, and N. H. Margolus, “Cellular Automata Machines: a New Environment for Modeling”, The MIT Press(1987).

[8] S. Wolfram, A New Kind of Science, Champaign Illinois: Wolfram Media(2002).

[9] M. A. Azim, M. Rubaiyat Kibria, and A. Jamalipour, “An optimized forwarding protocol for lifetime extension of wireless sensor networks”, Wiress Communication & Mobile Computing, vol. 9, no. 1, pp. 103-115, Jan. 2009.

[10] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks”, IEEE Transactions on Wireless Communication, vol. 1, no. 4, pp. 660-670, 2002.

[11] Y. F. Wen, Y. Q. Chen, M. Pan, “Adaptive ant-based routing in wireless sensor networks using Energy*Delay metrics”, Journal of Zhejiang University: Science A, vol. 9, no. 4, pp. 531-538, Apr. 2008.

[12] G. Kortuem, F. Kawsar, “Market-based user innovation in the Internet of things”, Internet of things(IOT), vol. 1 no. 8, 2010.

 ISBN: 978-1-61208-184-7

[13] M. C. Qasim, S. Erchin, and Q. Khalid, “On Maximum Likelihood Estimation of Clock Offset and Skew in Networks With Exponential Delays”, IEEE Transactions on Signal Processing, vol. 56, no. 4, pp. 1685-1697, 2008.

[14] Y. Liu, ZH. H. Qian, X. Wang, and Y. N. Li, “Wireless sensor network centroid localization algorithm based on time difference of arrival”, Journal of Jilin University (Engineering and Technology Edition), vol. 40, no. 1, pp. 245-249, Jan. 2010.

 ISBN: 978-1-61208-184-7

Intelligent Safety Verification for Pipeline Process Order Control Based on bf-EVALPSN

Kazumi Nakamatsu
School of Human Science and Environment
University of Hyogo
Himeji, Japan
Email: nakamatu@shse.u-hyogo.ac.jp

Jair Minoro Abe
GP in Production Eng., ICET
Paulista University
Sao Paulo, Brazil
Email: jairabe@uol.com.br

Seiki Akama
C-corporation
Kawasaki, Japan
Email:akama@jcom.home.ne.jp

***Abstract*—A paraconsistent logic program called Extended Vector Annotated Logic Program with Strong Negation(abbr. EVALPSN) has been developed for dealing with defeasible deontic reasoning and plausible reasoning, and also applied to various kinds of intelligent safety verification and control. Moreover, in order to deal with before-after relation between processes (time intervals), another EVALPSN called bf(before-after)-EVALPSN has been developed recently. In this paper, we review the reasoning system for before-after relation between processes based on bf-EVALPSN and introduce how to apply the reasoning system to real-time pipeline process order safety verification and control with an example.**

Keywords- before-after relation; paraconsistent logic program; safety verification; pipeline process order; reasoning system.

I. INTRODUCTION

It has already passed over two decades since paraconsistent annotated logic and its logic programming have been developed [3], [4]. Based on the original annotated logic program, we have developed four kinds of paraconsistent annotated logic program, ALPSN (Annotated Logic Program with Strong Negation), VALPSN (Vector ALPSN), EVALPSN (Extended VALPSN) that can deal with defeasible deontic and plausible reasonings, and bf (before-after)-EVALPSN that can deal with before-after relation between processes (time intervals) [9]. We note that "before-after" is abbreviated as just "bf" hereafter. Those annotated logic programs have been applied to various kinds of intelligent control and safety verification such as pipeline valve control based on safety verification [7] and real-time process order control based on safety verification [10], and so on. Moreover, it has been shown that EVALPSN can be implemented on microchips as electronic circuits, which implies that EVALPSN is suitable for real-time control [8].

In this paper, we review the reasoning system for process before-after relation in bf-EVALPSN [10] and show how to apply it to the safety verification for process order with an example. The before-after relation reasoning system based on bf-EVALPSN consists of two groups of inference rules called *basic bf-inference rules* and *transitive bf-inference rules*, both of which can be represented in bf-EVALPSN.

In bf-EVALPSN, a special annotated literal, $R(p_m, p_n, t) : [(i, j), \mu]$ called *bf-literal* whose non-negative integer vector annotation (i, j) represents the before-after relation between processes Pr_m and Pr_n at time t is introduced. The integer components i and j of the vector annotation (i, j) represent the after and before degrees between processes Pr_m and Pr_n, respectively, and before-after relations between two processes are represented in vector annotations.

In the bf-EVALPSN reasoning system, the basic bf-inference rules are used for determining the vector annotation of a bf-literal in real-time according to the start/finish time information of two processes; on the other hand, the transitive bf-inference rules are used for determining the vector annotation of a bf-literal in real-time based on the vector annotations of two related bf-literals as follows. Suppose that there are three processes, Pr_0, Pr_1 and Pr_2 starting in sequence, then the before-after relation between processes Pr_0 and Pr_2 can be determined from the before-after relation between processes Pr_0 and Pr_1, and that between processes Pr_1 and Pr_2. Such process before-after relation reasoning is also formalized as transitive bf-inference rules in bf-EVALPSN. The transitive bf-inference system can contribute to reduce the frequency of basic bf-inference rules and it is a unique feature of our system.

This paper is organized as follows: first, EVALPSN is reviewed briefly and bf-EVALPSN is defined in details; next, it is shown how to reason before-after relations in bf-EVALPSN with a simple example of process order control, and basic bf-inference rules and transitive bf-inference rules are introduced; furthermore, a simple practical process order verification system is provided as an example; last, a related work of treating before-after relation of time intervals in a logical system and our future work are introduced.

II. EVALPSN

In this section, we review EVALPSN briefly [5]. Generally, a truth value called an *annotation* is explicitly attached to each literal in annotated logic programs [3]. For example, let p be a literal, μ an annotation, then $p : \mu$ is called

 ISBN: 978-1-61208-184-7

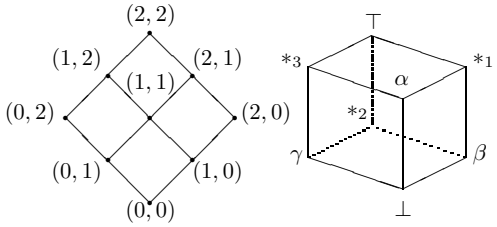


Figure 1. Lattice $\mathcal{T}_v(2)$ and Lattice $\mathcal{T}_d$

an *annotated literal*. The set of annotations constitutes a complete lattice. An annotation in EVALPSN has a form of $[(i, j), \mu]$ called an *extended vector annotation*. The first component (i, j) is called a *vector annotation* and the set of vector annotations constitutes the complete lattice,

$$\mathcal{T}_v(n) = \{\ (x, y) | 0 \leq x \leq n, 0 \leq y \leq n, \\ x, y, n \text{ are integers } \}$$

in Fig. 1. The ordering($\preceq_v$) of $\mathcal{T}_v(n)$ is defined as : let (x_1, y_1), $(x_2, y_2) \in \mathcal{T}_v(n)$,

$$(x_1, y_1) \preceq_v (x_2, y_2) \text{ iff } x_1 \leq x_2 \text{ and } y_1 \leq y_2.$$

For each extended vector annotated literal $p : [(i, j), \mu]$, the integer i denotes the amount of positive information to support the literal p and the integer j denotes that of negative one. The second component μ is an index of fact and deontic notions such as obligation, and the set of the second components constitutes the complete lattice,

$$\mathcal{T}_d = \{\bot, \alpha, \beta, \gamma, *_1, *_2, *_3, \top\}.$$

The ordering ($\preceq_d$) of $\mathcal{T}_d$ is described by the Hasse's diagram in Fig. 1. The intuitive meaning of each member of $\mathcal{T}_d$ is

$\bot$ (unknown), α (fact), β (obligation),
γ (non-obligation), $*_1$ (fact and obligation),
$*_2$ (obligation and non-obligation),
$*_3$ (fact and non-obligation), $\top$ (inconsistency).

Then the complete lattice $\mathcal{T}_e(n)$ of extended vector annotations is defined as the product $\mathcal{T}_v(n) \times \mathcal{T}_d$. The ordering ($\preceq_e$) of $\mathcal{T}_e(n)$ is defined: let $[(i_1, j_1), \mu_1]$, $[(i_2, j_2), \mu_2] \in \mathcal{T}_e$,

$$[(i_1, j_1), \mu_1] \preceq_e [(i_2, j_2), \mu_2] \quad \text{iff} \\ (i_1, j_1) \preceq_v (i_2, j_2) \text{ and } \mu_1 \preceq_d \mu_2.$$

There are two kinds of *epistemic negation* ($\neg_1$ and $\neg_2$) in EVALPSN, both of which are defined as mappings over $\mathcal{T}_v(n)$ and $\mathcal{T}_d$, respectively.

Definition 1 (epistemic negations $\neg_1$ and $\neg_2$ in EVALPSN)

$$\neg_1([(i, j), \mu]) = [(j, i), \mu], \quad \forall \mu \in \mathcal{T}_d,$$
$$\neg_2([(i, j), \bot]) = [(i, j), \bot], \quad \neg_2([(i, j), \alpha]) = [(i, j), \alpha],$$
$$\neg_2([(i, j), \beta]) = [(i, j), \gamma], \quad \neg_2([(i, j), \gamma]) = [(i, j), \beta],$$
$$\neg_2([(i, j), *_1]) = [(i, j), *_3], \quad \neg_2([(i, j), *_2]) = [(i, j), *_2],$$
$$\neg_2([(i, j), *_3]) = [(i, j), *_1], \quad \neg_2([(i, j), \top]) = [(i, j), \top].$$

If we regard the epistemic negations as syntactical operations, the epistemic negations followed by literals can be eliminated by the syntactical operations. For example,

$$\neg_1(p : [(2, 0), \alpha]) = p : [(0, 2), \alpha] \quad \text{and}$$
$$\neg_2(q : [(1, 0), \beta]) = p : [(1, 0), \gamma].$$

There is another negation called *strong negation* ($\sim$) in EVALPSN, and it is treated as well as classical negation.

Definition 2 (strong negation $\sim$) (see [4]) Let F be any formula and $\neg$ be $\neg_1$ or $\neg_2$.

$$\sim F =_{def} F \rightarrow ((F \rightarrow F) \wedge \neg(F \rightarrow F)).$$

Definition 3 (well extended vector annotated literal) Let p be a literal.

$$p : [(i, 0), \mu] \quad \text{and} \quad p : [(0, j), \mu]$$

are called *well extended vector annotated literals*, where $i, j \in \{1, 2, \cdots, n\}$, and $\mu \in \{\ \alpha,\ \beta,\ \gamma\ \}$.

Definition 4 (EVALPSN) If $L_0, \cdots, L_n$ are weva-literals,

$$L_1 \wedge \cdots \wedge L_i \wedge \sim L_{i+1} \wedge \cdots \wedge \sim L_n \rightarrow L_0$$

is called an *EVALPSN clause*. An *EVALPSN* is a finite set of EVALPSN clauses.

Here we comment that if the annotations α and β represent fact and obligation, notions "fact", "obligation", "forbiddance" and "permission" can be represented by extended vector annotations, $[(m, 0), \alpha]$, $[(m, 0), \beta]$, $[(0, m), \beta]$, and $[(0, m), \gamma]$, respectively, in EVALPSN, where m is a non-negative integer.

III. BEFORE-AFTER EVALPSN

In this section, we review bf-EVALPSN. The details are found in [10] The reasoning system in bf-EVALPSN consists of two kinds of inference rules called *basic bf-inference rule* and *transitive bf-inference rule*, which will be introduced with some simple examples of real-time process order control in the following sections.

In bf-EVALPSN, a special annotated literal $R(p_m, p_n, t) : [(i, j), \mu]$ called *bf-literal* whose non-negative integer vector annotation (i, j) represents the before-after relation between processes Pr_m and Pr_n at time t is introduced. The integer components i and j of the vector annotation (i, j) represent the after and before degrees between processes $Pr_m(p_m)$ and $Pr_n(p_n)$, respectively, and before-after relations are represented paraconsistently in vector annotations.

 ISBN: 978-1-61208-184-7

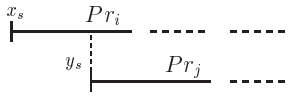


Figure 2. Bf-relations Before(be)/After(af)

Figure 3. Bf-relations Disjoint Before (db)/After (da)

Definition 5(bf-EVALPSN)
An extended vector annotated literal $R(p_i, p_j, t) : [(i, j), \mu]$ is called a *bf-EVALP literal* or a *bf-literal* for short, where (i, j) is a vector annotation and $\mu \in \{\alpha, \beta, \gamma\}$. If an EVALPSN clause contains bf-EVALP literals, it is called a *bf-EVALPSN clause* or just a *bf-EVALP clause* if it contains no strong negation. A *bf-EVALPSN* is a finite set of bf-EVALPSN clauses.

We provide a paraconsistent before-after interpretation for vector annotations representing bf-relations in bf-EVALPSN, and such a vector annotation is called a *bf-annotation*. Exactly speaking, bf-relation has fifteen meaningful kinds according to bf-relations between each start/finish time of two processes in bf-EVALPSN. Let us start from the most basic bf-relations in bf-EVALPSN.

Before (be)/After (af)
are defined according to the bf-relation between each start time of two processes. If one process has started before/after another one starts, then the bf-relations between them are defined as "before/after", which are represented in Fig. 2.

We introduce other kinds of bf-relations as well as before (be)/after (af). The original idea of the classification of process before-after relations has introduced in [1].

Disjoint Before (db) /After (da)
are defined as there is a time lag between the earlier process finish time and the later one start time, which are described in Fig. 3.

Immediate Before (mb)/After (ma)
are defined as there is no time lag between the earlier process finish time and the later one start time, which are described in Fig. 4.

Joint Before (jb)/After (ja)
are defined as two processes overlap and the earlier process had finished before the later one finished, which are described in Fig. 5.

S-included Before (sb), S-included After (sa)

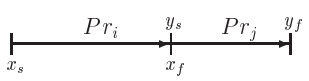


Figure 4. Bf-relations Immediate Before(mb)/After(ma)

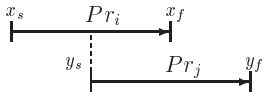


Figure 5. Bf-relations, Joint Before/After

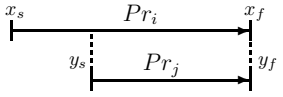


Figure 6. Bf-relations S-included Before(sb)/After(sa)

are defined as one process had started before another one started and they have finished at the same time, which are described in Fig. 6.

Included Before (ib)/After (ia)
are defined as one process had started/finished before/after another one started/finished, which are described in Fig. 7.

F-included Before (fb)/After (fa)
are defined as the two processes have started at the same time and one process had finished before another one finished, which are described in Fig. 8.

Paraconsistent Before-after (pba)
is defined as two processes have started at the same time and also finished at the same time, which is described in Fig. 9.

The epistemic negation over bf-annotations, be, af, db, da, mb, ma, jb, ja, ib, ia, sb, sa, fb, fa, pba is defined and the complete lattice of bf-annotations is shown in Fig. 10.

Definition 6 (Epistemic Negation $\neg_1$ for Bf-annotations)
The epistemic negation $\neg_1$ over the bf-annotations is obviously defined as the following mappings :

$$\begin{array}{lll} \neg_1(\mathtt{af}) = \mathtt{be}, & \neg_1(\mathtt{be}) = \mathtt{af}, & \neg_1(\mathtt{da}) = \mathtt{db}, \\ \neg_1(\mathtt{db}) = \mathtt{da}, & \neg_1(\mathtt{ma}) = \mathtt{mb}, & \neg_1(\mathtt{mb}) = \mathtt{ma}, \\ \neg_1(\mathtt{ja}) = \mathtt{jb}, & \neg_1(\mathtt{jb}) = \mathtt{ja}, & \neg_1(\mathtt{sa}) = \mathtt{sb}, \\ \neg_1(\mathtt{sb}) = \mathtt{sa}, & \neg_1(\mathtt{ia}) = \mathtt{ib}, & \neg_1(\mathtt{ib}) = \mathtt{ia}, \\ \neg_1(\mathtt{fa}) = \mathtt{fb}, & \neg_1(\mathtt{fb}) = \mathtt{fa}, & \neg_1(\mathtt{pba}) = \mathtt{pba}. \end{array}$$

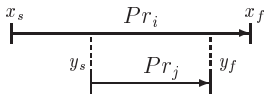


Figure 7. Bf-relations Included Before (ib)/After (ia)

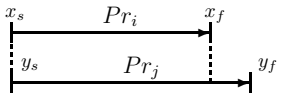


Figure 8. Bf-relations F-included Before(fb)/After(fa)

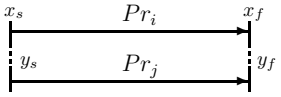


Figure 9. Bf-relation, Paraconsistent Before-after

Therefore, each bf-annotation can be translated into vector annotatations as $\mathtt{bf}=(0,8)$, $\mathtt{db}=(0,12)$, $\mathtt{mb}=(1,11)$, $\mathtt{jb}=(2,10)$, $\mathtt{sb}=(3,9)$, $\mathtt{ib}=(4,8)$, $\mathtt{fb}=(5,7)$, $\mathtt{pba}=(6,6)$.

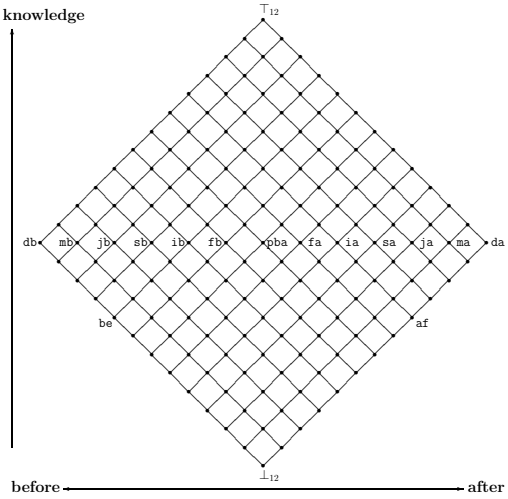


Figure 10. The Complete Lattice $\mathcal{T}_v(12)_{bf}$ of Bf-annotations

IV. Reasoning System in Bf-EVALPSN

In this section, we introduce the bf-relation reasoning system in bf-EVALPSN, which consists of two kinds of inference rules called basic bf-inference rules and transitive bf-inference rules.

A. Basic Before-after Inference Rule

In order to represent basic bf-inference rules in bf-EVALPSN, we newly introduce two literals:

$st(p_i,t)$, which is intuitively interpreted that process Pr_i starts at time t, and

$fi(p_i,t)$, which is intuitively interpreted that process Pr_i finishes at time t,

which are used for expressing process start/finish information and have one of the vector annotations, $(0,0)$, $\mathtt{t}(1,0)$, $\mathtt{f}(0,1)$, $(1,1)$, where annotations $\mathtt{t}$ and $\mathtt{f}$ can be intuitively interpreted as "true" and "false", respectively.

Here we introduce the first group of basic bf-inference rules to be applied at the initial stage, which are called $(0,0)$-*rules*.

(0,0)-rules

Suppose that no process has started yet and the vector annotation of bf-literal $R(p_i,p_j,t)$ is $(0,0)$, which shows that there is no knowledge in terms of the bf-relation between processes Pr_i and Pr_j.

$(0,0)$-rule-1 If process Pr_i started before process Pr_j starts, then the vector annotation $(0,0)$ of bf-literal $R(p_i,p_j,t)$ should turn to bf-annotation $\mathtt{be}(0,8)$.

$(0,0)$-rule-2 If both processes Pr_i and Pr_j have started at the same time, then the vector annotation $(0,0)$ of bf-literal $R(p_i,p_j,t)$ should turn to $(5,5)$.

Basic bf-inference rules $(0,0)$-rule-1 and 2 can be translated into the bf-EVALPSN clauses,

$$R(p_i,p_j,t):[(0,0),\alpha] \wedge st(p_i,t):[\mathtt{t},\alpha] \wedge \sim st(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(0,8),\alpha], \quad (1)$$

$$R(p_i,p_j,t):[(0,0),\alpha] \wedge st(p_i,t):[\mathtt{t},\alpha] \wedge st(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(5,5),\alpha]. \quad (2)$$

Next, suppose that one of basic bf-inference rules $(0,0)$-rule-1 and 2 has been applied, then the vector annotation of bf-literal $R(p_i,p_j,t)$ should be $(0,8)$ or $(5,5)$. Therefore, we have the following two groups of basic bf-inference rules to be applied after basic bf-inference rules $(0,0)$-rule, which are called $(0,8)$-*rules* and $(5,5)$-*rules*.

(0,8)-rules

Suppose that the vector annotation of bf-literal $R(p_i,p_j,t)$ is $\mathtt{be}(0,8)$. Then we have the following bf-inference rules to be applied after basic bf-inference rule $(0,0)$-rule-1.

$(0,8)$-rule-1 If process Pr_i has finished before process Pr_j starts, and process Pr_j starts immediately after process Pr_i finished, then the vector annotation $(0,8)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{mb}(1,11)$.

$(0,8)$-rule-2 If process Pr_i has finished before process Pr_j starts, and process Pr_j has not started immediately after process Pr_i finished, then the vector annotation $(0,8)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{db}(0,12)$.

$(0,8)$-rule-3 If process Pr_j starts before process Pr_i finishes, then the vector annotation $(0,8)$ of bf-literal $R(p_i,p_j,t)$ should turn to $(2,8)$ that is the greatest lower bound of the bf-annotations, $\mathtt{jb}(2,10)$, $\mathtt{sb}(3,9)$, $\mathtt{ib}(4,8)$.

Basic bf-inference rules $(0,8)$-rule-1,2 and 3 can be translated into the bf-EVALPSN clauses,

$$R(p_i,p_j,t):[(0,8),\alpha] \wedge fi(p_i,t):[\mathtt{t},\alpha] \wedge st(p_j,t):[\mathtt{t}] \rightarrow R(p_i,p_j,t):[(1,11),\alpha], \quad (3)$$

$$R(p_i,p_j,t):[(0,8),\alpha] \wedge fi(p_i,t):[\mathtt{t},\alpha] \wedge \sim st(p_j,t):[\mathtt{t}] \rightarrow R(p_i,p_j,t):[(0,12),\alpha], \quad (4)$$

$$R(p_i,p_j,t):[(0,8),\alpha] \wedge \sim fi(p_i,t):[\mathtt{t},\alpha] \wedge st(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(2,8),\alpha]. \quad (5)$$

(5,5)-rules

Suppose that the vector annotation of bf-literal $R(p_i,p_j,t)$ is $(5,5)$. Then we have the following bf-inference rules to be applied after basic bf-inference rule $(0,0)$-rule-2.

$(5,5)$-rule-1 If process Pr_i has finished before process Pr_j finishes, then the vector annotation $(5,5)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{sb}(5,7)$.

 ISBN: 978-1-61208-184-7

$(5,5)$-rule-2 If both processes Pr_i and Pr_j have finished at the same time, then the vector annotation $(5,5)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{pba}(6,6)$.

$(5,5)$-rule-3 If process Pr_j has finished before process Pr_i finishes, then the vector annotation $(5,5)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{sa}(7,5)$.

Basic bf-inference rules $(5,5)$-rules-1,2 and 3 can be translated into the bf-EVALPSN clauses,

$$R(p_i,p_j,t):[(5,5),\alpha] \wedge fi(p_i,t):[\mathtt{t},\alpha] \wedge \sim fi(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(5,7),\alpha], \quad (6)$$

$$R(p_i,p_j,t):[(5,5),\alpha] \wedge fi(p_i,t):[\mathtt{t},\alpha] \wedge fi(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(6,6),\alpha], \quad (7)$$

$$R(p_i,p_j,t):[(5,5),\alpha] \wedge \sim fi(p_i,t):[\mathtt{t},\alpha] \wedge fi(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(7,5),\alpha]. \quad (8)$$

If one of basic bf-inference rules $(5,5)$-rule-1,2 and 3, and $(0,8)$-rule-1 and 2 has been applied, then complete bf-relations such as $\mathtt{jb}(2,10)/\mathtt{ja}(10,2)$ should be inferred. On the other hand, if basic bf-inference rule $(0,8)$-rule-3 has been applied, no complete bf-annotation could be inferred. Therefore, a group of basic bf-inference rules called $(2,8)$-*rules* should be considered after basic bf-inference rule $(0,8)$-rule-3.

(2,8)-rules

Suppose that the vector annotation of bf-literal $R(p_i,p_j,t)$ is $(2,8)$. Then we have the following bf-inference rules to be applied after basic bf-inference rule $(0,8)$-rule-3.

$(2,8)$-rule-1 If process Pr_i finished before process Pr_j finishes, then the vector annotation $(2,8)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{jb}(2,10)$.

$(2,8)$-rule-2 If both processes Pr_i and Pr_j have finished at the same time, then the vector annotation $(2,8)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{fb}(3,9)$.

$(2,8)$-rule-3 If process Pr_j has finished before Pr_i finishes, then the vector annotation $(2,8)$ of bf-literal $R(p_i,p_j,t)$ should turn to $\mathtt{ib}(4,8)$.

Basic bf-inference rules $(2,8)$-rule-1,2 and 3 can be translated into the bf-EVALPSN clauses,

$$R(p_i,p_j,t):[(2,8),\alpha] \wedge fi(p_i,t):[\mathtt{t},\alpha] \wedge \sim fi(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(2,10),\alpha], \quad (9)$$

$$R(p_i,p_j,t):[(2,8),\alpha] \wedge fi(p_i,t):[\mathtt{t},\alpha] \wedge fi(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(3,9),\alpha], \quad (10)$$

$$R(p_i,p_j,t):[(2,8),\alpha] \wedge \sim fi(p_i,t):[\mathtt{t},\alpha] \wedge fi(p_j,t):[\mathtt{t},\alpha] \rightarrow R(p_i,p_j,t):[(4,8),\alpha]. \quad (11)$$

The application orders of all basic bf-inference rules are summarized in Table I.

B. Transitive Before-after Inference Rule

Here we introduce another kind of bf-inference rules, transitive bf-inference rule.

Table I
APPLICATION ORDERS OF BASIC BF-INFERENCE RULES

vector	rule	vector	rule	vector	rule	vector
$(0,0)$	rule-1	$(0,8)$	rule-1	$(0,12)$		
			rule-2	$(1,11)$		
			rule-3	$(2,8)$	rule-1	$(2,10)$
					rule-2	$(3,9)$
					rule-3	$(4,8)$
	rule-2	$(5,5)$	rule-1	$(5,7)$		
			rule-2	$(6,6)$		
			rule-3	$(7,5)$		

Suppose that there are three processes Pr_i,Pr_j and Pr_k starting sequentially, then we consider to reason the vector annotation of bf-literal $R(p_i,p_k,t)$ from those of bf-literals $R(p_i,p_j,t)$ and $R(p_j,p_k,t)$ transitively. First of all, we will show a simple example for forming transitive bf-inference rules as introduction.

Example 1

Suppose that both processes Pr_i and Pr_j have already started at time t but process Pr_k has not started yet as shown in Fig. 11, then we have obtained the vector annotation $(2,8)$ of bf-literal $R(p_i,p_j,t)$ by basic bf-inference rule $(0,8)$-rule-3 and the vector annotation $(0,8)$ of bf-literal $R(p_j,p_k,t)$ by basic bf-inference rule $(0,0)$-rule-1. Then obviously the vector annotation of bf-literal $R(p_i,p_k,t)$ is reasoned as bf-annotation $\mathtt{be}(0,8)$. Thus, we may obtain the following bf-EVALP clause as a transitive bf-inference rule,

$$R(p_i,p_j,t):[(2,8),\mu] \wedge R(p_j,p_k,t):[(0,8),\mu] \rightarrow R(p_i,p_k,t):[(0,8),\mu], \qquad \mu \in \{\alpha,\beta,\gamma\}.$$

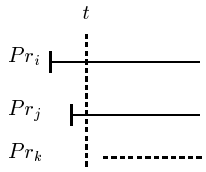


Figure 11. Process Time Chart Ex-3

Here we list all transitive bf-inference rules. The details of how to construct transitive bf-inference rules are in [10] For simplicity, we represent a transitive bf-inference rule,

$$R(p_i,p_j,t):[(n_1,n_2),\alpha] \wedge R(p_j,p_k,t):[(n_3,n_4),\alpha] \rightarrow R(p_i,p_k,t):[(n_5,n_6),\alpha]$$

by only vector annotations and logical connectives, $\wedge$ and $\rightarrow$, as follows: $(n_1,n_2) \wedge (n_3,n_4) \rightarrow (n_5,n_6)$ in the list of transitive bf-inference rules.

Transitive Bf-inference Rules

$$\mathbf{TR0} \quad (0,0) \wedge (0,0) \rightarrow (0,0)$$

$\mathbf{TR1}$ $(0,8) \wedge (0,0) \rightarrow (0,8)$

$\mathbf{TR1-1}$ $(0,12) \wedge (0,0) \rightarrow (0,12)$

$\mathbf{TR1-2}$ $(1,11) \wedge (0,8) \rightarrow (0,12)$

$\mathbf{TR1-3}$ $(1,11) \wedge (5,5) \rightarrow (1,11)$

$\mathbf{TR1-4}$ $(2,8) \wedge (0,8) \rightarrow (0,8)$

$\mathbf{TR1-4-1}$ $(2,10) \wedge (0,8) \rightarrow (0,12)$

$\mathbf{TR1-4-2}$ $(4,8) \wedge (0,12) \rightarrow (0,8)$ (12)

$\mathbf{TR1-4-3}$ $(2,8) \wedge (2,8) \rightarrow (2,8)$

$\mathbf{TR1-4-3-1}$ $(2,10) \wedge (2,8) \rightarrow (2,10)$

$\mathbf{TR1-4-3-2}$ $(4,8) \wedge (2,10) \rightarrow (2,8)$ (13)

$\mathbf{TR1-4-3-3}$ $(2,8) \wedge (4,8) \rightarrow (4,8)$

$\mathbf{TR1-4-3-4}$ $(3,9) \wedge (2,10) \rightarrow (2,10)$

$\mathbf{TR1-4-3-5}$ $(2,10) \wedge (4,8) \rightarrow (3,9)$

$\mathbf{TR1-4-3-6}$ $(4,8) \wedge (3,9) \rightarrow (4,8)$

$\mathbf{TR1-4-3-7}$ $(3,9) \wedge (3,9) \rightarrow (3,9)$

$\mathbf{TR1-4-4}$ $(3,9) \wedge (0,12) \rightarrow (0,12)$

$\mathbf{TR1-4-5}$ $(2,10) \wedge (2,8) \rightarrow (1,11)$

$\mathbf{TR1-4-6}$ $(4,8) \wedge (1,11) \rightarrow (2,8)$ (14)

$\mathbf{TR1-4-7}$ $(3,9) \wedge (1,11) \rightarrow (1,11)$

$\mathbf{TR1-5}$ $(2,8) \wedge (5,5) \rightarrow (2,8)$

$\mathbf{TR1-5-1}$ $(4,8) \wedge (5,7) \rightarrow (2,8)$ (15)

$\mathbf{TR1-5-2}$ $(2,8) \wedge (7,5) \rightarrow (4,8)$

$\mathbf{TR1-5-3}$ $(3,9) \wedge (5,7) \rightarrow (2,10)$

$\mathbf{TR1-5-4}$ $(2,10) \wedge (7,5) \rightarrow (3,9)$

$\mathbf{TR2}$ $(5,5) \wedge (0,8) \rightarrow (0,8)$

$\mathbf{TR2-1}$ $(5,7) \wedge (0,8) \rightarrow (0,12)$

$\mathbf{TR2-2}$ $(7,5) \wedge (0,12) \rightarrow (0,8)$ (16)

$\mathbf{TR2-3}$ $(5,5) \wedge (2,8) \rightarrow (2,8)$

$\mathbf{TR2-3-1}$ $(5,7) \wedge (2,8) \rightarrow (2,10)$

$\mathbf{TR2-3-2}$ $(7,5) \wedge (2,10) \rightarrow (2,8)$ (17)

$\mathbf{TR2-3-3}$ $(5,5) \wedge (4,8) \rightarrow (4,8)$

$\mathbf{TR2-3-4}$ $(7,5) \wedge (3,9) \rightarrow (4,8)$

$\mathbf{TR2-4}$ $(5,7) \wedge (2,8) \rightarrow (1,11)$

$\mathbf{TR2-5}$ $(7,5) \wedge (1,11) \rightarrow (2,8)$ (18)

$\mathbf{TR3}$ $(5,5) \wedge (5,5) \rightarrow (5,5)$

$\mathbf{TR3-1}$ $(7,5) \wedge (5,7) \rightarrow (5,5)$ (19)

$\mathbf{TR3-2}$ $(5,7) \wedge (7,5) \rightarrow (6,6)$

Note : the bottom vector annotation $(0,0)$ in the list of transitive bf-inference rules implies that any bf-EVALP clause $R(p_j, p_k, t):[(n,m),\alpha]$ satisfies it.

Here we indicate two important points in terms of transitive bf-inference rules.

(I) The number chain 1-4-3 of transitive bf-inference rule TR1-4-3 show the rule applicable order, that is to say, rule TR1-4-3 should be applied after rule TR1-4 and rule TR1-4 should be applied after TR1, if they are applicable.

(II) Transitive bf-inference rules, TR1-4-2 (12), TR2-2 (16), TR1-4-3-2 (13), TR2-3-2 (17), TR1-4-6 (14), TR2-5 (18), TR1-5-1 (15), TR3-1 (19) have no following rule to be applied at the following stage, even though they cannot derive the final bf-relations. For example, suppose that rule TR1-4-3-2 has been applied, then the vector annotation $(2,8)$ of the bf-literal $R(p_i, p_k, t)$ just implies that the final bf-relation between processes Pr_i and Pr_k is one of three bf-annotations, $\mathtt{jb}(2,10)$, $\mathtt{sb}(3,9)$ and $\mathtt{ib}(4,8)$. Therefore, if one of the eight transitive bf-inference rules has been applied, one of basic bf-inference rules $(0,8)$-rule, $(2,8)$-rule or $(5,5)$-rule should be applied for deriving the final bf-annotation. For instance, if rule TR1-4-3-2 has been applied, basic bf-inference rule $(2,8)$-rule should be applied at the following stage.

V. Application of Bf-EVALPSN to Pipeline Process Order Verification

In this section, we present a simple example for applying the bf-relation reasoning system in bf-EVALPSN to process order verification.

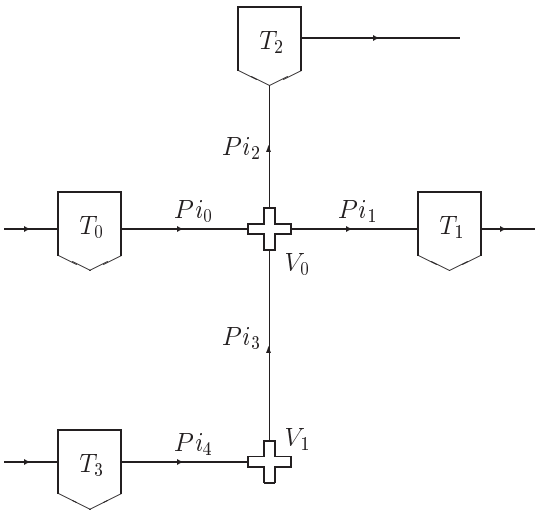


Figure 12. Pipeline Network

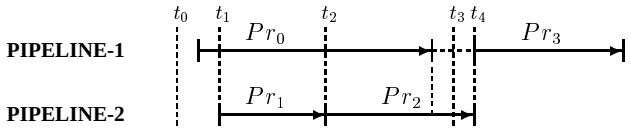


Figure 13. Pipeline Process Schedule

Example 2

We consider a simple brewery pipeline network in Fig. 12, which consists of four tanks, $\{T_0, T_1, T_2, T_3\}$; five pipes,

$\{Pi_0,Pi_1,Pi_2,Pi_3,Pi_4\}$; two valves, $\{V_0, V_1\}$. Two kinds of pipeline cleaning processes by nitric acid and cold water, respectively, are supposed to be processed after brewery processes, and the following four processes in the brewery pipeline network are scheduled according to the time chart in Fig. 13:

- process Pr_0 (brewery process 1), beer is transfered in the pipeline,

$$T_0 \ \rightarrow \ Pi_0 \ \rightarrow \ V_0 \ \rightarrow \ Pi_1 \ \rightarrow \ T_1;$$

- process Pr_1 (cleaning process 1), the pipeline,

$$T_3 \ \rightarrow \ pi_4 \ \rightarrow \ V_1 \ \rightarrow \ Pi_3 \ \rightarrow \ V_0 \ \rightarrow \ Pi_2,$$

  is cleaned by nitric acid;
- process Pr_2 (cleaning process 2), the pipeline,

$$T_3 \ \rightarrow \ pi_4 \ \rightarrow \ V_1 \ \rightarrow \ Pi_3 \ \rightarrow \ V_0 \ \rightarrow \ Pi_2$$

  is cleaned by cold water;
- process Pr_3 (brewery process 2), beer is transfered in the pipelines,

$$\begin{array}{r} T_0 \ \rightarrow \ Pi_0 \ \rightarrow \ V_0 \ \rightarrow \ Pi_1 \ \rightarrow \ T_1, \\ T_3 \ \rightarrow \ Pi_4 \ \rightarrow \ V_1 \ \rightarrow \ Pi_3 \ \rightarrow \ V_0 \ \rightarrow \ Pi_2 \ \rightarrow \ T_2 \end{array}$$

  with mixing beer at valve V_0.

Moreover the pipeline system has four safety properties $SPR-i(i = 0, 1, 2, 3)$ to be strictly secured in terms of process order.

SPR-0 process Pr_0 must start before any other processes, and process Pr_0 must finish before process Pr_2 finishes.

SPR-1 process Pr_1 must start after process Pr_0 starts.

SPR-2 process Pr_2 must start immediately after process Pr_1 finishes.

SPR-3 process Pr_3 must start immediately after processes Pr_0 and Pr_2 finish.

Our safety verification system consists of the following three steps:

STEP 1 in order to verify the safety of process order, the safety properties should be translated into bf-EVALPSN and stored.

STEP 2 the before-after relations between processes at each time should also be translated into bf-EVALP clauses and added to the stored bf-EVALPSN in the previous step.

STEP 3 the safety of starting a process is verified as bf-EVALPSN logic programming at each time.

We introduce the above safety verification steps with the pipeline process order. First of all, the safety properties are translated. Safety property *SPR-0* can be translated into the bf-EVALPSN clauses,

$$\sim R(p_0, p_1, t) : [(0, 8), \alpha] \rightarrow st(p_1, t) : [\mathtt{f}, \beta], \quad (20)$$
$$\sim R(p_0, p_2, t) : [(0, 8), \alpha] \rightarrow st(p_2, t) : [\mathtt{f}, \beta], \quad (21)$$
$$\sim R(p_0, p_3, t) : [(0, 8), \alpha] \rightarrow st(p_3, t) : [\mathtt{f}, \beta], \quad (22)$$
$$st(p_1, t) : [\mathtt{f}, \beta] \wedge st(p_2, t) : [\mathtt{f}, \beta] \wedge st(p_3, t) : [\mathtt{f}, \beta]$$
$$\rightarrow st(p_0, t) : [\mathtt{f}, \gamma], \quad (23)$$
$$\sim fi(p_0, t) : [\mathtt{f}, \beta] \rightarrow fi(p_0, t) : [\mathtt{f}, \gamma], \quad (24)$$

where bf-EVALPSN clauses (20),(21) and (22) declare that if process Pr_0 has not started before any other processes, it should be forbidden to start processes $Pr_i(i = 1, 2, 3)$; the bf-EVALPSN clause (23) declares that if each process $Pr_i(i = 1, 2, 3)$ is forbidden from starting, it should be permitted to start process Pr_0; and the bf-EVALPSN clause (24) declares that if there is no forbiddance from finishing process Pr_0, it should be permitted to finish process Pr_0.

Safety property *SPR-1* can be translated into the EVALP -SN clauses,

$$\sim st(p_1, t) : [\mathtt{f}, \beta] \rightarrow st(p_1, t) : [\mathtt{f}, \gamma], \quad (25)$$
$$\sim fi(p_1, t) : [\mathtt{f}, \beta] \rightarrow fi(p_1, t) : [\mathtt{f}, \gamma], \quad (26)$$

where EVALPSN clauses (25) and (26) declare that if there is no forbiddance from starting and finishing process Pr_1, respectively, it should be permitted to start and finish process Pr_1.

Safety property *SPR-2* can be translated into the bf-EVALPSN clauses,

$$\sim R(p_2, p_1, t) : [(11, 0), \alpha] \rightarrow st(p_2, t) : [\mathtt{f}, \beta], \quad (27)$$
$$\sim st(p_2, t) : [\mathtt{f}, \beta] \rightarrow st(p_2, t) : [\mathtt{f}, \gamma], \quad (28)$$
$$\sim R(p_2, p_0, t) : [(10, 2), \alpha] \rightarrow fi(p_2, t) : [\mathtt{f}, \beta], \quad (29)$$
$$\sim fi(p_2, t) : [\mathtt{f}, \beta] \rightarrow fi(p_2, t) : [\mathtt{f}, \gamma], \quad (30)$$

where bf-EVALPSN clause (27) declares that if process Pr_1 has not finished before process Pr_2 starts, it should be forbidden to start process Pr_2; the vector annotation $(11, 0)$ of bf-literal $R(p_2, p_1, t)$ is the greatest lower bound of the set $\{\mathtt{da}(12, 0), \mathtt{ma}(11, 1)\}$, which implies that process Pr_1 has finished before process Pr_2 starts in either way; EVALPSN clauses (28) and (30) declare that if there is no forbiddance from starting and finishing process Pr_2, it should be permitted to start and finish process Pr_2, respectively; and bf-EVALPSN clauses (29) declares that if process Pr_0 has not finished before process Pr_2 finishes, it should be forbidden to finish process Pr_2.

Safety property *SPR-3* can be translated into the bf-EVALPSN clauses,

$$\sim R(p_3, p_0, t) : [(11, 0), \alpha] \rightarrow st(p_3, t) : [\mathtt{f}, \beta], \quad (31)$$
$$\sim R(p_3, p_1, t) : [(11, 0), \alpha] \rightarrow st(p_3, t) : [\mathtt{f}, \beta], \quad (32)$$
$$\sim R(p_3, p_2, t) : [(11, 0), \alpha] \rightarrow st(p_3, t) : [\mathtt{f}, \beta], \quad (33)$$
$$\sim st(p_3, t) : [\mathtt{f}, \beta] \rightarrow st(p_3, t) : [\mathtt{f}, \gamma], \quad (34)$$
$$\sim fi(p_3, t) : [\mathtt{f}, \beta] \rightarrow fi(p_3, t) : [\mathtt{f}, \gamma], \quad (35)$$

 ISBN: 978-1-61208-184-7

where bf-EVALPSN clauses (31),(32) and (33) declare that if none of processes $Pr_i (i = 0, 1, 2)$ has not finished, it should be forbidden to start process Pr_3; and bf-EVALPSN clauses (34) and (35) declare that if there is no forbiddance from starting and finishing process Pr_3, it should be permitted to start and finish process Pr_3, respectively.

Now we show how the process order verification in bf-EVALPSN is carried out at time t_0, $\cdots$, time t_4 according to the process schedule in Fig. 13. Five bf-relations between processes Pr_0, Pr_1, Pr_2 and Pr_3, which are represented by bf-literals, $R(p_0, p_1, t)$, $R(p_0, p_2, t)$, $R(p_0, p_3, t)$, $R(p_2, p_1, t)$ and $R(p_3, p_2, t)$ are verified based on safety properties *SPR-0*, *SPR-1*, *SPR-2* and *SPR-3* in bf-EVALPSN.
Stage 0 (at time t_0) no process has started at time t_0, thus, the bf-EVALP clauses,

$$R(p_0, p_1, t_0) : [(0, 0), \alpha], \quad (36)$$
$$R(p_1, p_2, t_0) : [(0, 0), \alpha], \quad (37)$$
$$R(p_2, p_3, t_0) : [(0, 0), \alpha] \quad (38)$$

are obtained; also the bf-EVALP clauses,

$$R(p_0, p_2, t_0) : [(0, 0), \alpha], \quad (39)$$
$$R(p_0, p_3, t_0) : [(0, 0), \alpha] \quad (40)$$

are obtained by rule TR0; then bf-EVALP clauses (36) and (39) satisfy each body of bf-EVALPSN clauses (20), (21) and (22), respectively, therefore, the forbiddance from starting each process $Pr_i (i = 1, 2, 3)$,

$$st(p_1, t_0) : [\mathtt{f}, \beta], \quad (41)$$
$$st(p_2, t_0) : [\mathtt{f}, \beta], \quad (42)$$
$$st(p_3, t_0) : [\mathtt{f}, \beta] \quad (43)$$

are derived; moreover as bf-EVALP clauses (41), (42) and (43) satisfy the body of bf-EVALP clause (23), the permission for starting process Pr_0,

$$st(p_0, t_0) : [\mathtt{f}, \gamma]$$

is derived; therefore, process Pr_0 is permitted to start.
Stage 1 (at time t_1) process Pr_0 has already started but all other processes $Pr_i (i = 1, 2, 3)$ have not started yet; then the bf-EVALP clauses,

$$R(p_0, p_1, t_1) : [(0, 8), \alpha], \quad (44)$$
$$R(p_1, p_2, t_1) : [(0, 0), \alpha], \quad (45)$$
$$R(p_2, p_3, t_1) : [(0, 0), \alpha] \quad (46)$$

are obtained, where bf-EVALP clause (44) is derived by $(0, 0)$-rule-1; moreover the bf-EVALP clauses,

$$R(p_0, p_2, t_1) : [(0, 8), \alpha], \quad (47)$$
$$R(p_0, p_3, t_1) : [(0, 8), \alpha] \quad (48)$$

are obtained by rule TR1; as bf-EVALP clause (44) does not satisfy the body of bf-EVALPSN clause (20), the forbiddance from starting process Pr_1,

$$st(p_1, t_1) : [\mathtt{f}, \beta] \quad (49)$$

cannot be derived; then, as there is not the forbiddance (49), the body of bf-EVALPSN clause (25) is satisfied, and the permission for starting process Pr_1,

$$st(p_1, t_1) : [\mathtt{f}, \gamma]$$

is derived; on the other hand, as bf-EVALP clauses (47) and (48) satisfy the body of bf-EVALPSN clauses (27) and (31), respectively, the forbiddance from starting both processes Pr_2 and Pr_3,

$$st(p_2, t_1) : [\mathtt{f}, \beta], \quad \text{and} \quad st(p_3, t_1) : [\mathtt{f}, \beta]$$

are derived; therefore, process Pr_1 is permitted to start.
Stage 2 (at time t_2) process Pr_1 has just finished and process Pr_0 has not finished yet; then the bf-EVALP clauses,

$$R(p_0, p_1, t_2) : [(4, 8), \alpha],$$
$$R(p_1, p_2, t_2) : [(1, 11), \alpha],$$
$$R(p_2, p_3, t_2) : [(0, 8), \alpha]$$

are derived by $(2, 8)$-rule-3, $(0, 8)$-rule-2 and $(0, 0)$-rule-1, respectively; moreover the bf-EVALP clauses,

$$R(p_0, p_2, t_2) : [(2, 8), \alpha], \quad (50)$$
$$R(p_0, p_3, t_2) : [(0, 12), \alpha] \quad (51)$$

are obtained by rules TR1-4-6 and TR1-2, respectively; then, as bf-EVALP clauses (50), (50) and (50) do not satisfy the body of bf-EVALPSN clause (27), the forbiddance from starting process Pr_2,

$$st(p_2, t_2) : [\mathtt{f}, \beta] \quad (52)$$

cannot be derived; as there is not the forbiddance (52), it satisfies the body of bf-EVALPSN clause (28), and the permission for starting process Pr_2,

$$st(p_2, t_2) : [\mathtt{f}, \gamma]$$

is derived; on the other hand, as bf-EVALP clause (51) satisfies the body of bf-EVALPSN clause (31), the forbiddance from starting process Pr_3,

$$st(p_3, t_2) : [\mathtt{f}, \beta]$$

is derived; therefore, process Pr_2 is permitted to start, however, process Pr_3 is still forbidden from starting.
Stage 3 (at time t_3) process Pr_0 has finished, process Pr_2 has not finished yet, and process Pr_3 has not started yet; then the bf-EVALP clauses,

$$R(p_0, p_1, t_3) : [(4, 8), \alpha],$$
$$R(p_1, p_2, t_3) : [(1, 11), \alpha] \quad \text{and}$$
$$R(p_2, p_3, t_3) : [(0, 8), \alpha],$$

 ISBN: 978-1-61208-184-7

which have the same vector annotations as the previous stage are obtained; moreover the bf-EVALP clauses,

$$R(p_0, p_2, t_3) : [(2, 10), \alpha], \quad (53)$$
$$R(p_0, p_3, t_3) : [(0, 12), \alpha] \quad (54)$$

are obtained, where bf-EVALP clause (53) is derived by $(2, 8)$-rule-1; then bf-EVALP clause (53) satisfies the body of bf-EVALP clause (33), and the forbiddance from starting process Pr_3,

$$S(p_3, t_3) : [\mathtt{f}, \beta]$$

is derived; therefore, process Pr_3 is still forbidden from starting because process Pr_2 has not finished yet at this stage.

Stage 4 (at time t_4) process Pr_2 has just finished and process Pr_3 has not started yet; then the bf-EVALP clauses,

$$R(p_0, p_1, t_4) : [(4, 8), \alpha], \quad (55)$$
$$R(p_1, p_2, t_4) : [(1, 11), \alpha], \quad (56)$$
$$R(p_2, p_3, t_4) : [(1, 11), \alpha], \quad (57)$$
$$R(p_0, p_2, t_4) : [(2, 10), \alpha], \quad (58)$$
$$R(p_0, p_3, t_4) : [(0, 12), \alpha] \quad (59)$$

are obtained; the bf-EVALP clause (57) is derived by $(0, 8)$-rule-2; moreover, as bf-EVALP clauses (55), (58) and (59) do not satisfy the bodies of bf-EVALP clauses (31), (32) and (33), the forbiddance from starting process Pr_3,

$$st(p_3, t_4) : [\mathtt{f}, \beta] \quad (60)$$

cannot be derived; as there is not the forbiddance (60), the body of bf-EVALPSN clause (34) is satisfied, and the permission for starting process Pr_3,

$$st(p_3, t_4) : [\mathtt{f}, \gamma]$$

is derived; therefore, process Pr_3 is permitted to start because processes Pr_0, Pr_1 and Pr_2 have finished.

VI. Conclusion

In this paper, we have introduced a logical reasoning system for before-after relations between processes (time intervals) based on a paraconsistent annotated logic program bf-EVALPSN, which consists of two groups of inference rules in bf-EVALPSN called basic and transitive bf-inference rules.

As related work, an interval temporal logic has been proposed for developing practical planning and natural language understanding systems in Allen [1], [2]. In his logic, before-after relations between two time intervals are represented in special predicates and treated in a framework of first order temporal logic. On the other hands, in our bf-EVALPSN before-after reasoning system, before-after relations between processes are regarded as paraconsistency and represented more minutely in vector annotations of the special literal $R(p_i, p_j, t)$ called bf-literal, and treated in the framework of annotated logic programming. Moreover an efficient real-time before-after relation reasoning mechanism called transitive bf-inference is implemented in our system. Therefore, we would like to conclude that our bf-EVALPSN before-after relation reasoning system is more suitable for dealing with process order safety verification and control in real-time, with considering its hardware implementation such as on microchips.

Our system has a lot of applications though, our future work focuses on its application to logical design for various process order control systems based on the safety verification.

References

[1] Allen, J.F., "Towards a General Theory of Action and Time", *Artificial Intelligence* vol. 23, pp. 123–154, 1984.

[2] Allen, J.F. and Ferguson, G., "Actions and Events in Interval Temporal Logic", *J.Logic and Computation* vol. 4, pp. 531–579, 1994.

[3] Blair, H.A. and Subrahmanian, V.S., "Paraconsistent Logic Programming", *Theoretical Computer Science* vol. 68, pp. 135–154, 1989.

[4] da Costa, N.C.A., Subrahmanian, V.S., and Vago, C., "The Paraconsistent Logics P$\mathcal{T}$", *Zeitschrift für Mathematische Logic und Grundlangen der Mathematik* vol. 37, pp. 139–148, 1989.

[5] Nakamatsu, K., Abe, J.M., and Suzuki, A., "Annotated Semantics for Defeasible Deontic Reasoning", *Rough Sets and Current Trends in Computing*, LNAI vol. 2005, pp. 432–440, 2001.

[6] Nakamatsu, K., Abe, J.M., and Akama, S., "An intelligent safety verification based on a paraconsistent logic program", *Proc. 9th Intl. Conf. Knowledge-Based Intelligent Information and Engineering Systems(KES2005)*, LNAI vol. 3682, pp. 708–715, 2005.

[7] Nakamatsu, K., "Pipeline Valve Control Based on EVALPSN Safety Verification", *J.Advanced Computational Intelligence and Intelligent Informatics* vol. 10, pp. 647-656, 2006.

[8] Nakamatsu, K., Mita, Y., and Shibata, T., "An Intelligent Action Control System Based on Extended Vector Annotated Logic Program and its Hardware Implementation", *J.Intelligent Automation and Soft Computing* vol. 13, pp. 289–304, 2007.

[9] Nakamatsu, K. and Abe, J.M., "The development of Paraconsistent Annotated Logic Program", *Int'l J. Reasoning-based Intelligent Systems* vol. 1, pp. 92-112, 2009.

[10] Nakamatsu, K., Abe, J.M., and Akama, S., A Logical Reasoning System of Process Before-after Relation Based on a Paraconsistent Annotated Logic Program bf-EVALPSN, *Intl J. Knowledge-based and Intelligent Engineering Systems* vol. 15, pp. 145–163, 2011.

 ISBN: 978-1-61208-184-7

Secure Communication Based on Indirect Coupled Synchronization

Rupak Kharel
School of Engineering
Manchester Metropolitan University
Manchester, UK
r.kharel@mmu.ac.uk

Krishna Busawon, Zabih Ghassemlooy
School of Computing, Engineering and Information Sciences
Northumbria Unviversity
Newcastle Upon Tyne, UK
krishna.busawon@northumbria.ac.uk,
z.ghassemlooy@northumbria.ac.uk

Abstract— **In this paper, a secure communication system composed of four chaotic oscillators is proposed. Two of these oscillators are unidirectionally coupled and employed as transmitter and receiver. The other two oscillators are indirectly coupled and are employed as keystream generators. The novelty lies in the generation of the same chaotic keystream both in the transmitter and receiver side for encryption and decryption purposes. We show, in particular, that it is possible to synchronize the two keystream generators even though they are not directly coupled. So doing, an estimation of the keystream is obtained allowing decrypting the message. The main feature of the proposed communication scheme is that the keystream cannot be generated with the sole knowledge of the transmitted chaotic signal, hence making it very secure. The performance of the proposed communication scheme is shown via simulation using the Chua and Lorenz oscillators.**

Keywords- Chaotic communication systems; chaotic synchronization; Lorenz System; Chua System

I. INTRODUCTION

The importance of chaotic synchronization for the development of secure communication systems is well-understood by now [1-6]. In recent years, various chaotic synchronization methods have been proposed [3-5, 7, 8] together with a number of modulation methods for chaotic communication systems such as chaotic masking [1, 5], parameter modulation techniques [5], chaotic shift keying [2, 5], just to mention a few. Each of these methods requires chaotic synchronization for message extraction at the receiver side. On the other hand, different attacks methods have been derived in order to test the security of the modulation methods; namely the non-linear dynamics forecasting [9, 10], return maps analysis [11], artificial neural network analysis [12] and so on. As a result, methods like chaotic masking, parameter modulation techniques and chaotic shift keying were found not to be secure. Other proposed methods based on the projective synchronization [13], phase synchronization [14], generalized synchronized [15] were broken as well [16, 17]. Methods based on the time delay or the hyperchaos were also looked upon for increasing the security but they too were found not to be entirely convincing [18, 19]. Therefore, there is a need of developing a method which will resist all the attack methods.

In [6], a method based on encryption technique was proposed, where a different output from chaotic transmitter which was transmitted in the channel was used as a keystream to encrypt the message signal. The encrypted message signal masked with another output of the chaotic oscillator was employed as the transmitted signal. It was claimed that since the intruder could not get hold of the keystream, it was impossible for the attackers to extract the message. Unfortunately a later work done by Parker and Short [20] showed that it was still possible to extract the keystream from the transmitted chaotic signal since the keystream carried the information of the dynamics of the transmitter. In fact, since, both the carrier and keystream were the outputs of same oscillator; the carrier held the dynamics of the keystream as well. Therefore, it was impossible to hide the dynamics of the keystream from intruders, as a signal has to be transmitted from the transmitter to the receiver for synchronization and message transmission purpose. However, since the principle of the method proposed in [6] is nevertheless interesting, there is a real incentive for finding ways for improving the method by eliminating its shortcomings.

In effect, in this paper, based on the spirit of the work in [6], we propose a new chaotic communication scheme composed of four chaotic oscillators. Two of those oscillators are uni-directionally coupled and employed as transmitter and receiver. The other two oscillators are indirectly coupled and are employed as keystream generators. The key idea therefore is to generate a chaotic carrier signal from one oscillator while a chaotic keystream is generated from another chaotic oscillator. A suitable encryption rule is employed in order to encrypt the message using the generated keystream. The encrypted message is then modulated with the chaotic carrier in order to generate the transmitted signal. As a result, the transmitted signal does not contain the dynamics of the keystream oscillator, hence making it difficult for intruders to generate the keystream with the sole knowledge of the transmitted chaotic signal. At the receiver, the same keystream is generated and a decryption rule is applied to the recovered

 ISBN: 978-1-61208-184-7

encrypted message signal that has been obtained from chaotic synchronization. However, this scheme gives rise to an interesting question: *Is it possible to synchronize two independent chaotic oscillators such that they generate same required keystream*? It will be shown in the next section that, under some assumptions, it is still possible to synchronize two chaotic oscillators even though they are not uni-directionally coupled.

An outline of the paper is as follow: In Section II, the main methodology of the proposed technique is explained. In addition, indirect coupled synchronization is proven for a class of chaotic systems. In Section III, the proposed synchronization and secure chaotic communication scheme are implemented using the Lorenz system and Chua's system. In Section IV, simulation is carried out and results are outlined to show the performance of the proposed communication scheme. Finally, in Section V, concluding remarks are made.

II. THE PROPOSED COMMUNICATION SYSTEM

The proposed chaotic communication scheme, based on cryptography, is shown in Fig. 1. The novelty here lies in the generation of the keystream. The chaotic transmitter (T) is first used to generate two output signals, $y_1(t)$ and $y_2(t)$. The signal $y_1(t)$ is used for modulation purpose while output $y_2(t)$ is used to drive chaotic oscillator (A) whose structure is different from the transmitter (T). The output $k(t)$ of key generator (A) is used as a keystream to encrypt the message $m(t)$ using an encryption rule $\phi(.)$. The resulting encrypted signal $\phi(m(t))$ is masked using $y_1(t)$ yielding the transmitted signal $y_t(t)$. The output $y_t(t)$ is fed back into the transmitter in the form of an output injection with the aim of cancelling the effect of non-linearity while performing synchronization at the receiver side. The modulated transmitted signal $y_t(t)$ is sent through the channel to the receiver.

At the receiver end, upon receiving the signal $y_t'(t)$, the chaotic receiver (R) - which is similar in structure to the transmitter (T) - permits to obtain an estimate $\hat{y}_1(t)$ and $\hat{y}_2(t)$ of the signals $y_1(t)$ and $y_2(t)$ respectively by synchronization. This can be done by using any techniques existing in the literature such as observers, etc [3, 4, 7, 8]. The signals $\hat{y}_1(t)$ and $y_t'(t)$ are used to generate an estimate $\hat{\phi}(m(t))$ of the encrypted signal $\phi(m(t))$. The estimate $\hat{y}_2(t)$ is used to drive the chaotic key generator (B) - which is similar in structure to generator (A) – and which yields the keystream estimate $\hat{k}(t)$. Consequently, the message $m(t)$ can be recovered by using the decryption rule $\phi^{-1}(.)$.

Note that since, the chaotic key generators (A) and (B) are driven by $y_2(t)$ and $\hat{y}_2(t)$ respectively, an indirect coupled synchronization is required between these two chaotic oscillators. Also, $y_2(t)$ and $\hat{y}_2(t)$ are outputs of chaotic transmitter (T) and receiver (R) respectively and will be equal once synchronization is achieved. Intuitively, one would expect this synchronization to take place. However, in what follows this will be proven mathematically for a class of chaotic systems.

The important part of this method is the generation of the keystream. No information regarding the keystream is transmitted in the channel. In [6], it was possible to estimate the particular state which was used as keystream (as shown in [20]) since the state that was transmitted in the channel had some information of the dynamics of the keystream as they were the state variables of same chaotic oscillator.

In contrast, in this method, the keystream is generated from a chaotic oscillator with a totally different structure. It will not be possible to estimate the dynamics of the chaotic key generator from the signal being transmitted in the channel by using the method mentioned in [20]. Even if the intruder manages to get hold of the encrypted signal from the transmitted signal, without the knowledge of keystream, the message signal can't be decrypted back. Therefore, a secure communication link can be realized by implementing the proposed method.

Based on the communication scheme illustrated by Fig. 1, we assume that the transmitter oscillator (T) described by a dynamical system of the following form:

$$(\mathrm{T}):\begin{cases}\dot{x} = \mathrm{F}(y_t)x + g(t, y_t)\\ y_1 = h_1(x)\\ y_2 = h_2(x)\\ y_t = y_1 + e(m,k),\end{cases} \tag{1}$$

where the state $x \in \mathbb{R}^n$ with initial condition $x(0) = x_0$. The outputs of the oscillator $y_1 \in \mathbb{R}$ and $y_2 \in \mathbb{R}$. The matrix **F** is of appropriate dimension while h_1 and h_2 are analytical vector functions. The signal $y_t \in \mathbb{R}$ is the transmitted signal where $e(.)$ is the encryption function using key $k(t)$ and the function g is a smooth bounded function of time.

The keystream $k(t)$ is generated using another chaotic oscillator of similar form:

$$(\mathrm{A}):\begin{cases}\dot{z} = \mathrm{A}z + b_2(t, y_2)\\ k = h(z),\end{cases} \tag{2}$$

which is driven by the output $y_2(t)$. Here, $z \in \mathbb{R}^q$ (q is not necessarily equal to n), $k \in \mathbb{R}$ is the keystream, h is an analytical vector function and b_2 is a smooth bounded function of time. It is assumed that the channel is perfect and that no distortion of the transmitted signal has taken place; that is $y_t = y_t'$.

The receiving chaotic oscillator (R) is given by:

$$(\mathrm{R}):\begin{cases}\dot{\hat{x}} = \mathrm{F}(y_t)\hat{x} + g(t, y_t)\\ \hat{y}_1 = h_1(\hat{x})\\ \hat{y}_2 = h_2(\hat{x}).\end{cases} \tag{3}$$

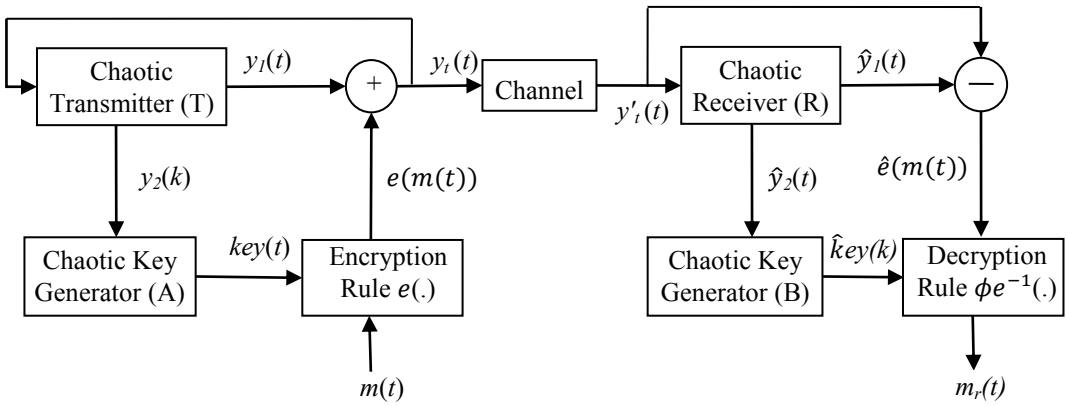


Fig. 1. Block diagram of the proposed chaotic communication based on cryptography.

Finally, the key generator (B) is given by:

$$(\mathrm{B}):\begin{cases}\dot{\hat{z}} = \mathrm{A}\hat{z} + b_2(t,\hat{y}_2) \\ \hat{k} = h(\hat{z}).\end{cases} \tag{4}$$

We shall make the following assumptions:

A1) There exist symmetric positive definite (SPD) matrices $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ such that

$$\mathbf{F}^{\mathrm{T}}\mathbf{P}_1 + \mathbf{P}_1\mathbf{F} = -\mathbf{Q}_1, \quad \mathbf{A}^{\mathrm{T}}\mathbf{P}_2 + \mathbf{P}_2\mathbf{A} = -\mathbf{Q}_2.$$

A2) The output function $h_2(x)$ is globally Lipschitzian with respect to x.

The objective is to show that the transmitter (T) and the receiver (R) synchronize as well as generators (A) and (B) are synchronized with each other even though there is no direct link between them. In effect, based on the above assumptions, we state the following:

Theorem 1. *Under the assumption A1), there exist two constants* $\lambda, \eta > 0$ *such that* $\|x(t) - \hat{x}(t)\| \le \eta e^{-\lambda t}\|x(0) - \hat{x}(0)\|$ *for all* $t \ge 0$. *In other words, the receiver (R) synchronizes exponentially with the transmitter (T).*

Proof: Let $\varepsilon(t) = x(t) - \hat{x}(t)$, then the error dynamics between transmitter (T) and receiver (R) is given by: $\dot{\varepsilon} = \mathbf{F}(y_t)\varepsilon$.

Owing to assumption A1), a candidate Lyapunov function of the above error dynamics can be chosen as: $V(\varepsilon) = \varepsilon^T\mathbf{P}_1\varepsilon$.

Differentiating $V(\varepsilon)$ with respect to time, yields:

$$\begin{aligned}\dot{V}(\varepsilon) &= \dot{\varepsilon}^T\mathbf{P}_1\varepsilon + \varepsilon^T\mathbf{P}_1\dot{\varepsilon} \\ &= \varepsilon^T\left[\mathbf{A}^T(y_t)\mathbf{P}_1 + \mathbf{P}_1\mathbf{A}(y_t)\right]\varepsilon = -\varepsilon^T\mathbf{Q}_1\varepsilon < 0.\end{aligned}$$

Since $\mathbf{Q}_1$ is SPD, there exist, c_1 , $c_2 > 0$ such that $c_1\varepsilon^T\mathbf{P}_1\varepsilon \le \varepsilon^T\mathbf{Q}_1\varepsilon \le c_2\varepsilon^T\mathbf{P}_1\varepsilon$. Consequently, $\dot{V}(\varepsilon) = -c_1V(\varepsilon)$. Integrating the last equation results in:

$$V(\varepsilon(t)) = e^{-c_1t}V(\varepsilon(0)). \tag{5}$$

Again, since $\mathbf{P}_1$ is SPD, there exist $\lambda_1, \lambda_2 > 0$ such that $\lambda_1\varepsilon^T\varepsilon \le \varepsilon^T\mathbf{P}_1\varepsilon \le \lambda_2\varepsilon^T\varepsilon$. Consequently: $\lambda_1\|\varepsilon(t)\|^2 \le \lambda_2 e^{-c_1t}\|\varepsilon(0)\|^2$.

In other words: $\|\varepsilon(t)\| \le \sqrt{\frac{\lambda_2}{\lambda_1}}e^{-\frac{c_1}{2}t}\|\varepsilon(0)\| = \eta e^{-\lambda t}\|\varepsilon(0)\|$.

That is:

$$\|x(t) - \hat{x}(t)\| \le \eta e^{-\lambda t}\|x(0) - \hat{x}(0)\|.$$

This means that $\hat{x}(t)$ converges to $x(t)$ exponentially. In other words, the receiver (R) synchronizes exponentially with the transmitter (T). This completes the proof of Theorem 1.

Theorem 2. *Assume that system (A) and (B) satisfies assumption A1), then* $\lim_{t\to\infty}\|z(t) - \hat{z}(t)\| = 0$. *That is, the keystream generator (A) synchronizes asymptotically with the keystream generator (B).*

Proof: Set $\zeta(t) = z(t) - \hat{z}(t)$, then the error dynamics between the keystream generator (A) and generator (B) is given by: $\dot{\zeta} = \mathbf{A}\zeta + b_2(t, y_2) - b_2(t, \hat{y}_2)$

Now consider the following candidate Lyapunov function $W = \zeta^T\mathbf{P}_2\zeta$. Differentiating W with respect to time yields

$$\begin{aligned}\dot{W} &= \zeta^T\mathbf{P}_2\dot{\zeta} + \dot{\zeta}^T\mathbf{P}_2\zeta = 2\zeta^T\mathbf{P}_2\dot{\zeta} \\ &= 2\zeta^T\mathbf{P}_2\left[\mathbf{A}\zeta + b_2(t, y_2) - b_2(t, \hat{y}_2)\right] \\ &= 2\zeta^T\mathbf{P}_2\mathbf{A}\zeta + 2\zeta^T\mathbf{P}_2\left[b_2(t, y_2) - b_2(t, \hat{y}_2)\right] \\ &\le -\zeta^T\mathbf{Q}_2\zeta + 2\left|\zeta^T\mathbf{P}_2\right|\left[b_2(t, y_2) - b_2(t, \hat{y}_2)\right] \\ &\le -\beta_1 W + \beta_2\|\zeta\|\|\varepsilon\| \\ &\le -\beta_1 W + \beta_3\sqrt{W}\|\varepsilon\|.\end{aligned}$$

Now,

$$\sqrt{\dot{W}} \le -\beta_1 W + \beta_3\|\varepsilon(t)\|.$$

Therefore,

$$\sqrt{W(\zeta(t))} \le -e^{-\beta_1 t}\sqrt{W(\zeta(0))} + \beta_3 \int_0^t e^{-\beta_1(t-\tau)} \|\varepsilon(t)\| d\tau.$$

From the above inequality, we can see that when $t \to +\infty$ $\|\zeta(t)\| \to 0$.

This completes the proof of Theorem 2 and therefore (A) converges with (B) asymptotically. Once the synchronization is obtained between (A) and (B), the message can be decrypted by applying the keystream.

III. APPLICATION OF THE PROPOSED TECHNIQUE USING THE CHUA AND THE LORENZ OSCILLATOR

In this section, the performance of the proposed communication system is demonstrated using the Lorenz system as the transmitter (T) and the receiver (R). More specifically, (T) and (R) are chosen as:

$$(\mathrm{T}):\begin{cases} \dot{u} = -\sigma u + \sigma v \\ \dot{v} = -20 y_t w + r y_t - v \\ \dot{w} = 5 y_t v - b w \\ y_1 = u \\ y_2 = v \\ y_t = y_1 + e(m,k). \end{cases}$$

$$(\mathrm{R}):\begin{cases} \dot{\hat{u}} = -\sigma \hat{u} + \sigma \hat{v} \\ \dot{\hat{v}} = -20 y_t \hat{w} + r y_t - \hat{v} \\ \dot{\hat{w}} = 5 y_t \hat{v} - b \hat{w} \\ \hat{y}_1 = \hat{u} \\ \hat{y}_2 = \hat{v}. \end{cases} \qquad (6)$$

Again it can easily be seen that (6) are in the form (1) and (3) with $\mathbf{F}(y_t)$ given as:

$$\mathbf{F}(y_t) = \begin{pmatrix} -\sigma & \sigma & 0 \\ 0 & -1 & -20 y_t \\ 0 & 5 y_t & -b \end{pmatrix}.$$

For these systems Assumption A1 hold true for the following choice of matrices $\mathbf{P}_1$ and $\mathbf{Q}_1$:

where $l_1, l_2, l_3, \sigma, b, r > 0$, $l_2 = -\frac{1}{4} l_3$ and $0 < l_1 < \frac{4}{\sigma} l_2$.

Remark 1. Note that, at first sight one would expect the matrices $\mathbf{P}_1$ and $\mathbf{Q}_1$ to be time dependent since $\mathbf{F}(y_t)$ is time dependent. However, interestingly, due to the particular form of $\mathbf{F}(y_t)$ the matrices turn out to be constants.

For the key generating oscillators A and B, the Chua's system is adopted given as below:

$$(\mathrm{A}):\begin{cases} \dot{p} = \alpha(q - p - f(y_2)) \\ \dot{q} = y_2 - q - s \\ \dot{s} = -\beta q - \gamma s \\ k = d_0 p. \end{cases} \quad (\mathrm{B}):\begin{cases} \dot{\hat{p}} = \alpha(\hat{q} - \hat{p} - f(\hat{y}_2)) \\ \dot{\hat{q}} = \hat{y}_2 - \hat{q} - \hat{s} \\ \dot{\hat{s}} = -\beta \hat{q} - \gamma \hat{s} \\ \hat{k} = d_0 \hat{p}. \end{cases} \qquad (7)$$

The non-linear function $f(\cdot)$ is a piecewise linear function given as:

$$f(\psi) = G_b \psi + 0.5(G_a - G_b)(|\psi + 1| - |\psi - 1|).$$

Note that 7 are in the form (2) and (4) respectively with $\mathbf{A}$ and $b_2(t, y_2)$ given as:

$$\mathbf{A} = \begin{pmatrix} -\alpha & \alpha & 0 \\ 0 & -1 & -1 \\ 0 & -\beta & -\gamma \end{pmatrix}, b_2(t, y_2) = \begin{pmatrix} -\alpha f(y_2) \\ y_2 \\ 0 \end{pmatrix}.$$

It can also be shown that Assumption A1) is satisfied for the following matrices $\mathbf{P}_2$ and $\mathbf{Q}_2$:

$$\mathbf{P}_2 = \begin{pmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{pmatrix} \& \mathbf{Q}_2 = \begin{pmatrix} 2\alpha l_1 & -\alpha l_1 & 0 \\ -\alpha l_1 & l_2 & 0 \\ 0 & 0 & 2\gamma l_3 \end{pmatrix},$$

where $l_1, l_2, l_3, \alpha > 0$, $\beta < 0, \gamma \ge 0$, $l_2 = -\beta l_3$ and $0 < l_1 < \frac{4}{\alpha} l_2$. Finally, it is obvious that A2) is satisfied. For the key generating oscillators A and B, the Lorenz system defined as is adopted:

The encryption function $e(.)$ used is a n-shift cipher algorithm given as: (as used in [6]):

$e(m(t)) = \underbrace{f_1(\ldots f_1(f_1}_{n}(m(t), \underbrace{k(t)), k(t)), \ldots, k(t))}_{n}$, where $f_1(.,.)$ is a non-linear function given by:

$$f(m,k) = \begin{cases} m + k + 2h, & \text{for } -2h \le m + k \le -h \\ m + k, & \text{for } -h \le m + k \le h \\ m + k - 2h, & \text{for } h \le m + k \le 2h \end{cases},$$

with h being an encryption parameter which is chosen such that m and k lie within the interval $[-h, h]$.

Once the keystream generator (A) synchronizes asymptotically with generator (B), the message $m(t)$ can be recovered using a decryption rule corresponding to the encryption rule and which is given by:

$m_r(t) = e^{-1}(\hat{e}(m(t))$

$= \underbrace{f_1(\ldots f_1(f_1}_{n}(\hat{e}(m(t), \underbrace{-\hat{k}(t)), -\hat{k}(t)), \ldots, -\hat{k}(t))}_{n}$, where $\hat{k}(t)$ is the estimated key stream and $\hat{e}(m(t)) = y_t - \hat{y}_1$.

In the next section, simulations are carried out using Matlab/Simulink and it will be shown that the proposed method is able to synchronize satisfactorily and extract the message successfully.

 ISBN: 978-1-61208-184-7

IV. SIMULATION RESULTS

The parameters employed in equation (15,16,18 and 19) are as follows:

$$\sigma = 16, r = 45.6, b = 4.2, \alpha = 10, \beta = -14.87$$

$$\gamma = 0, G_a = -1.27, G_b = -0.68, d_0 = 0.05.$$

The encryption parameter h is chosen to be 0.3 and the message $m(t) = 0.1\sin(2\pi t)$. Also in encryption rule, a 30-shift cipher is used. The initial conditions for each oscillator are chosen to arbitrarily different.

Fig. 2 shows the autocorrelation function of the keystream signal $k(t)$. It is clear that the keystream is not similar to itself with any amount of time shift so its autocorrelation function has only a single spike at point of zero time shift. This means the keystream generated is chaotic in nature and therefore has limited predictability. Fig. 3 shows the encrypted message signal using (21) and signal $k(t)$ as keystream. Fig. 4 depicts the transmitted chaotic carrier and it can be seen that message signal is totally buried inside it.

Fig. 5 illustrates the error in estimating the keystream and it can be seen that although two oscillators are starting from different initial conditions, the error converges rapidly to zero after some initial period taken for synchronization.

Fig. 6 shows the performance of the proposed method in decrypting the message signal back and it is readily seen that the transmitted message signal has been estimated convincingly. Next, the performance of the proposed secure communication method is tested in the presence of channel noise. For this purpose, the simulation is performed using the AWGN channel having SNR of 40 dB. The output is shown in Fig. 7, where it can be seen that message is extracted successfully. Apart from the jitter in amplitude, which can be removed from standard filtering operation, the necessary information about the message (form, frequency and amplitude) is obtained.

It is seen that the proposed method is used to transmit simple sinusoidal message signals. But the method is equally true for other message signals such as voice signals, square wave, etc. Also, the idea can be easily extended from analogue systems here to digital communication systems with proper modulation schemes. The modulation schemes can be PAM, FSK, PSK, etc. With digital communication systems, the SNR up to which the method works with noisy channel can easily be reduced from 40 dB. For, example, when PAM is used for transmitting digital bits then, after recovering the modulated square wave that has been corrupted with noise, it can easily be passed to matched filter and then threshold detected to recover the digital bits accurately.

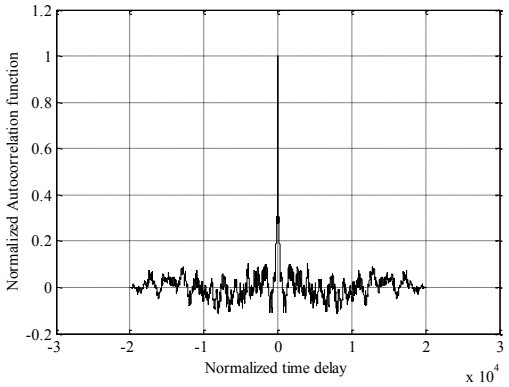


Fig. 2. Autocorrelation of key stream signal $k(t)$.

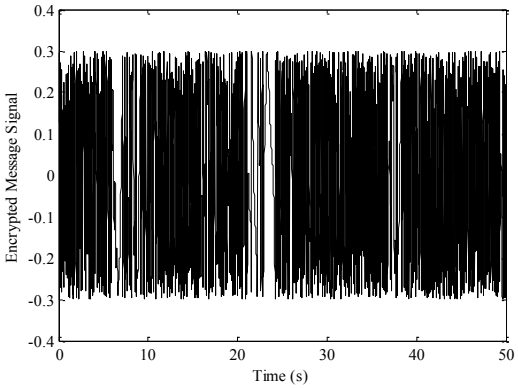


Fig. 3. Encrypted message signal $e(m(t))$.

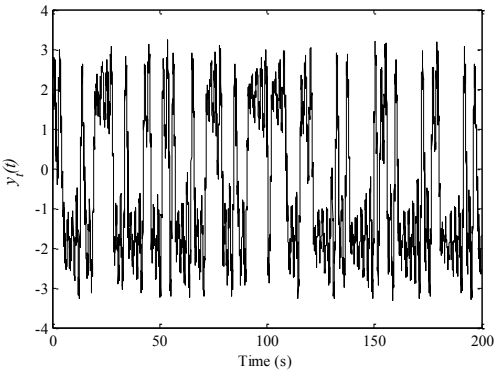


Fig. 4. Transmitted signal $y_t(t)$ generated from oscillator T.

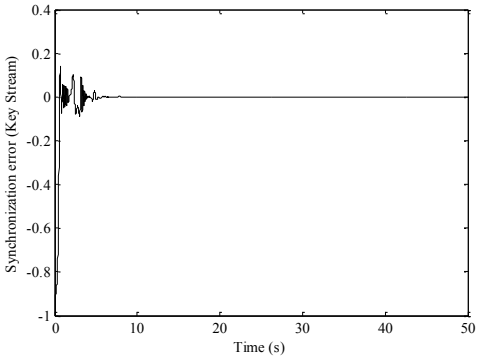


Fig. 5. Synchronization error in estimation of keystream.

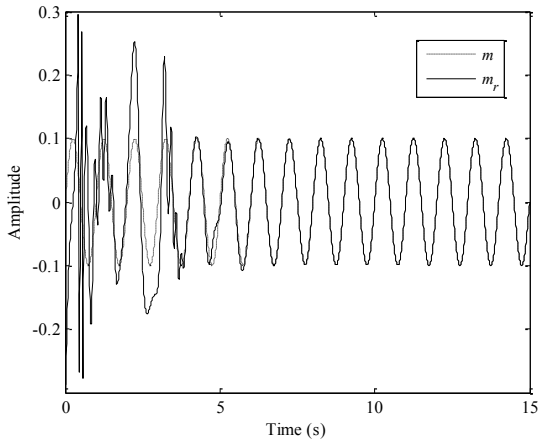


Fig. 6. Plot of the extracted message $m_r(t)$ and $m(t)$.

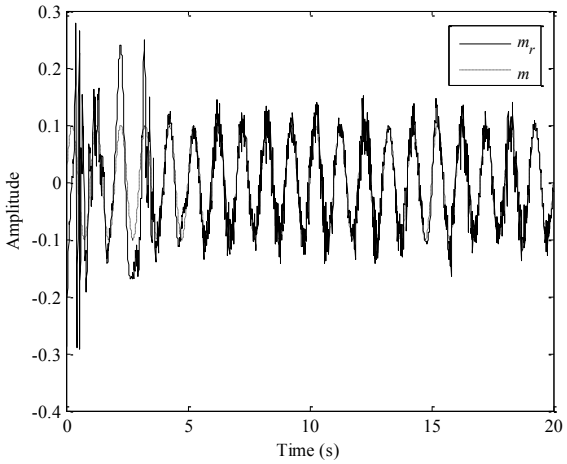


Fig. 7. Message extraction in AWGN channel of SNR 40 dB.

V. CONCLUSION AND FUTURE WORK

In this paper, a method of synchronizing two chaotic oscillators that are not directly coupled together in a master-slave configuration is proposed and applied to generate the keystream at transmitter and receiver. Synchronization is proven mathematically and simulation results are presented. The main advantage of the proposed method is that, unlike previous work on the topic, the keystream is generated from a different oscillator to that of the transmitter and hence improving the security of the system; since the transmitted signal does not include the information of the dynamics of the key generator. Consequently, even if the encrypted signal is known to the intruders, without the knowledge of the keystream extraction of the message signal will not be possible providing secure communication link. As future works, the communication scheme can be extended by employing more general chaotic systems and incorporating observers for the receiver and the key generator. Also, the scheme need to be implemented and tested practically.

REFERENCES

[1] K. M. Cuomo and A. V. Oppenheim, "Circuit implementation of synchronized chaos with applications to communications," Phys. Rev. Lett., vol. 71, pp. 65-68, 1993.

[2] L. Kocarev, K. S. Halle, and A. Shang, "Transmission of digital signals by chaotic synchronization," International Journal of Bifurcation and Chaos, vol. 2, pp. 973-977, 1992.

[3] M. L'Hernault, J.-P. Barbot, and A. Ouslimani, "Feasibility of Analog Realization of a Sliding-Mode Observer: Application to Data Transmission," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 55, pp. 614-624, 2008.

[4] O. Morgul, E. Solak, and M. Akgul, "Observer based chaotic message transmission," International Journal of Bifurcation and Chaos., vol. 13, pp. 1003-1017, 2003.

[5] T. Yang, "A survey of chaotic secure communication systems," International Journal of Computational Cognition, vol. 2, pp. 81-130, 2004.

[6] T. Yang, C. W. Wu, and L. O. Chua, "Cryptography based on chaotic systems," IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 44, pp. 469-472, 1997.

[7] T. L. Carroll and L. M. Pecora, "Synchronizing chaotic circuits," IEEE Transaction on Circuit and Systems-I: Fundamental Theory And Applications, vol. 38, pp. 453-456, 1991.

[8] H. Nijmeijer and I. M. Y. Mareels, "An observer looks at synchronization," IEEE Transactions on Circuits and Systems - I: Fundamental thoery and applications, vol. 44, pp. 882-890, 1997.

[9] K. M. Short, "Steps toward unmasking secure communications," International Journal of Bifurcation and Chaos, vol. 4, pp. 959-977, 1994.

[10] K. M. Short, "Unmasking a modulated chaotic communications scheme," International Journal of Bifurcation and Chaos, vol. 6, pp. 367-375, 1996.

[11] T. Yang, L. B. Yang, and C. M. Yang, "Cryptanalyzing chaotic secure communication using return maps," Physics Letters A, vol. 245, pp. 495-510, 1998.

[12] T. Yang, L. B. Yang, and C. M. Yang, "Application of neural networks to unmasking chaotic secure communication," Physica D, vol. 124, pp. 248-257, 1998.

[13] Z. Li and D. Xu, "A secure communication scheme using projective chaos synchronization," Chaos, Solitons & Fractals, vol. 22, pp. 477-481, 2004.

[14] J. Y. Chen, K. W. Wong, L. M. Cheng, and J. W. Shuai, "A secure communciation scheme based on the phase synchronization of chaotic systems," Chaos, vol. 13, pp. 508-514, 2003.

[15] M. Boutayeb, M. Darouach, and H. Rafaralahy, "Generalized State-Space Observers for Chaotic Synchronization and Secure Communication," IEEE Transaction on Circuit and Systems-I: Fundamental Theory And Applications, vol. 49, pp. 345-349, 2002.

[16] G. Alvarez, S. Li, F. Montoya, M. Romera, and G. Pastor, "Breaking projective chaos synchronization secure communication using filtering and generalized synchronization," Chaos Solitons & Fractals, vol. 24, pp. 775-883, 2005.

[17] G. Alvarez, F. Montoya, G. Pastor, and M. Romera, "Breaking a secure communication scheme based on the phase synchronization of chaotic systems," Chaos, vol. 14, pp. 274-278, 2004.

[18] K. M. Short and A. T. Parker, "Unmasking a hyperchaotic communication scheme," Physical Review E, vol. 58, pp. 1159-1162, 1998.

[19] C. Zhou and C. H. Lai, "Extracting messages masked by chaotic signals of time-delay systems," Physical Review E, vol. 60, pp. 320-323, 1999.

[20] A. T. Parker and K. M. Short, "Reconstructing the keystream from a chaotic encryption," IEEE Transaction on Circuit and Systems-I: Fundamental Theory And Applications, vol. 48, pp. 624-630, 2001.

 ISBN: 978-1-61208-184-7

Requirements Engineering for Software vs. Systems in General

Hermann Kaindl
Institute of Computer Technology
Vienna University of Technology
Vienna, Austria
kaindl@ict.tuwien.ac.at

Marko Jäntti
School of Computing
University of Eastern Finland
Kuopio, Finland
marko.jantti@uef.fi

Herwig Mannaert
Normalized Systems Institute
University of Antwerp
Antwerp,
Belgiumherwig.mannaert@ua.ac.be

Kazumi Nakamatsu
School of Human Science and Environment
University of Hyogo
Himeji, Japan
nakamatsu@shse.u-hyogo.ac.jp

Roland Rieke
Fraunhofer Institute for Secure Information Technology
Darmstadt, Germany
roland.rieke@sit.fraunhofer.de

***Abstract*—Are there fundamental technical differences between requirements engineering for software vs. systems in general? It seems as though even functional requirements can mean something more general for a system including mechanical parts than for software alone. Quality requirements on safety deal with humans and their relationship with some real artifacts in their environment, so they cannot be dealt with by software alone. However, reliability of underlying software will be important in this context. While the internal structure of software will not normally be specified in its requirements, structure of a more general system may well be. These are just examples of what should be discussed.**

With regard to intelligent enterprises, there exist defined methodologies for enterprise modeling. Much as any other complex system, an enterprise may be better understood through modeling. Once an enterprise is better understood, it may be easier to make it intelligent. Whatever technical system is to be developed in an enterprise, it needs to fit into. By connecting enterprise modeling and requirements engineering, the likelihood of such a fit is increased. For software development, such connections have been worked out and are part of defined methodologies, some of them based on object-oriented modeling. Are they applicable to the development of general systems?

Keywords-requirements engineering; software; systems; enterprises

I. INTRODUCTION

The panel discusses whether there are fundamental technical differences between requirements engineering for software as opposed to requirements engineering for systems in general. Each panelist has his own position as stated below.

II. PANELISTS AND THEIR POSITIONS

A. *Marko Jäntti*

In order to identify differences between requirements engineering of software and requirements engineering of systems one should start by clarifying the relationships between the concepts 'software' and 'system'. We can use a term information system to define the system. Besides software, an information system covers the hardware, infrastructure and people that use the system. Thus, system requirements engineering can be seen as a broader concept than software requirements engineering. Unified Modeling Language that is a widely used modeling notation can be used for modeling software structure and behavior [1]. UML can also be used to describe the physical nodes of a system (deployment diagram).

Unfortunately, software and system requirements engineering do not fully satisfy the needs of today's IT world that is becoming more and more service-oriented. Thus, the third aspect of requirements engineering is service requirements engineering. Service requirements typically include most of the functional and non-functional requirements of software products but also address some service-specific requirements such as service availability and quality of IT service support [2].

B. *Herwig Mannaert*

Though an information system is a much broader concept than software, the software on itself can be seen as a system as well. What software systems and various types of systems in general, including systems with mechanical parts and even enterprises [3], have in common, is that they can be regarded as modular structures. While no single generally accepted definition is known, the concept is most commonly associated with the process of subdividing a system into several subsystems, which is said to result in a certain degree of complexity reduction and facilitate change by allowing

 ISBN: 978-1-61208-184-7

modifications at the level of a single subsystem instead of the whole system [6, 7]. In software systems, one should strive to pay as much attention to the modular structure as mechanical systems currently do.

When considering systems in general — software systems, organizational systems, etc. — both a functional and constructional perspective should be taken into account [6]. The functional perspective focuses on describing what a particular system or unit does or what its function is. The structural perspective on the other hand, concentrates on the composition and structure of the system, i.e. which subsystems are part of the system and their relations. Equivalently, one could regard the functional system view as a blackbox representation, and the constructional system view as a whitebox representation. By blackbox we mean that only the input and output of a system is revealed by means of an interface, describing the way how the system interacts with its environment. As such, the user of the system does not need to know any details about the content or the inner way of working of the system. The main issue with respect to this approach in software systems, is that modules often exhibit hidden coupling that is not explicitly defined in the interfaces. The evolution towards service-oriented computing is, amongst other things, addressing this issue.

What also distinguishes software systems and software requirements from their mechanical counterparts, is that they are subject to change. Requirements evolve during the development of software systems, and both the requirements and the actual system will continue to evolve during the system lifecycle. It has been shown in [4, 5] that it is all but trivial for software systems to cope with these evolving requirements, and that this leads to structure degradation. This would also be the case for mechanical systems, but they are not required to evolve during their lifecycle.

C. *Kazumi Nakamatsu*

If we formalize logical structures of systems whatever they are software or human like systems, requirements for the system could be easily treated and implemented, especially for functional ones. However, if a system includes human factors, it would be much more complicated to model such systems than just mechanical systems. In order to model any kinds of systems, whatever human factors are included or not, we have developed a paraconsistent logic program called Extended Vector Annotated Logic Program with Strong Negation (abbr. EVALPSN)[8], which can deal with not only inconsistency but also human like reasoning such as plausible reasoning and some modalities such as obligation. Moreover we have used it for modeling man-machine systems such as the safety verification system for railway interlocking in order to avoid train accidents caused by human error.

As a conclusion, generally speaking, the EVALPSN based modeling is fitter for modeling systems including a lot of human factors than just software.

D. *Roland Rieke*

Architecting novel dependable systems or systems of systems poses new challenges to the system design process [9]. Dependability and security analysis is growing in complexity with the increase in functionality, connectivity, and dynamics of the systems. The application of models is becoming standard practice, in order to tackle this complexity and get the dependability and security requirements right, as early as possible in the system design process. A modeling framework for the specification of security and reliability requirements has to consider not only the structure and functional dependencies of a system but also the possible behavior. Actions in a model can represent software, hardware or human behavior. One way to specify requirements is, to define specific constraints regarding sequences of actions, which should occur or must not occur in a system's behavior. Actions in the model represent an abstract view on actions of the real system, therefore it has to be ensured, that the abstraction does not hide critical behavior. The requirements analysis should also consider the behavior of an attacker, which can be different in comparison to, e.g., the Byzantine fault model. An attack to physical components, for instance, to cut a vehicle's brake has to be done physically on site and so it can only attack one physical unit at a time. However, a remote attack to the software of a vehicular communication system could affect all vehicles at once.

E. *Hermann Kaindl*

Are all types of requirements equally relevant for software and systems in general? How can software achieve its functions? Actually, it is “dead” unless run on some hardware, mostly some general-purpose electronic computer. Only the calculations or symbol manipulations of such a computer as programmed by a piece of software may lead through myriads of state changes, i.e., some (internal) behavior. The results of the calculations or symbol manipulations to a given input are the functions of the software.

Contrast this with a chair, a very simply mechanical system. It achieves its function to support someone when sitting on it without any state change but only through its physical structure (and certain constraints on it). So, a general system including mechanical parts may have different ways of achieving functions than a software system alone.

So, an important difference to me between requirements engineering for software vs. systems in general is that mechanical parts may achieve functions by their structure and may, therefore, give rise to important structural requirements.

REFERENCES

[1] M. Jäntti, T. Toroi, “UML-Based Testing,” Proc. of the 2nd Nordic Workshop on the Unified Modeling Language (NWUML 2004), Aug. 2004, pp. 33-44.

 ISBN: 978-1-61208-184-7

[2] Office of Government Commerce, The Official Introduction to the ITIL Service Lifecycle. The Stationary Office, UK, 2007.

[3] D. Campagnolo and A. Camuffo, “The concept of modularity within the management studies: a literature review”, International Journal of Management Reviews, vol. 12, no. 3, pp. 259-283, 2009.

[4] Mannaert Herwig, Verelst Jan, Ven Kris.- Towards evolvable software architectures based on systems theoretic stability, Software practice and experience - ISSN 0038-0644 - 42(2012), p. 89-116.

[5] Mannaert Herwig, Verelst Jan, Ven Kris.- The transformation of requirements into software primitives : studying evolvability based on systems theoretic stability, Science of computer programming - ISSN 0167-6423 - 76:12(2011), p. 1210-1222.

[6] G. M. Weinberg, An Introduction to General Systems Thinking. Wiley-Interscience, 1975.

[7] C. Y. Baldwin and K. B. Clark, Design Rules: The Power of Modularity. Cambridge, MA, USA: MIT Press, 2000.

[8] K. Nakamatsu, and Jair M. Abe, “The Development of Paraconsistent Annotated Logic Programs”, Int. J. Reasoning-based Intelligent Systems, vol. 1, pp. 92–102, June 2009.

[9] A. Avizienis, J.-C. Laprie, B. Randell, and C. E. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," IEEE Trans. Dependable Sec. Comput., vol. 1, no. 1, pp. 1133, 2004.

 ISBN: 978-1-61208-184-7

Radial Basis Functions for High-Dimensional Visualization

Vaclav Skala
Department of Computer Science and Engineering
University of West Bohemia
CZ 306 14 Plzen, Czech Republic
skala@kiv.zcu.cz

***Abstract* —** High-dimensional visualization is usually connected with large data processing. Because of dimensionality, it is nearly impossible to make a tessellation, like the Delaunay tessellation in E^d, followed by data interpolation. One possibility of data interpolation is the use of the Radial Basis Functions (RBF) interpolation. The RBF interpolation supports the interpolation of scattered data in *d*-dimensional space. The computational cost of the RBF interpolation is higher but does not increase significantly with the data dimensionality. It increases with the number of values to be processed non-linearly. In this paper, the RBF interpolation properties will be discussed as well as how to process data incrementally. Incremental computation decreases computational complexity and decreases RBF computational cost for the given data set significantly, especially for the visualization purposes, when the interpolated/approximated data are used many times. As the proposed approach is based on a solution of a system of linear equations, the RBF interpolation is convenient especially for data sets processing using matrix-vector or GPU architectures.

Keywords - Visualization; computer graphics; interpolation; radial basis functions; RBF

I. INTRODUCTION

Visualization of potential (scalar) fields in a multi-dimensional space is a typical problem not only in physical sciences. The problem seems to be quite simple, but it is actually a quite complicated task. In the E^2 case the usual approach is to tessellate the domain (e.g. *x-y* space) and then to use linear interpolation or cubic interpolation. In general, the computational complexity of the Delaunay tessellation (DT) for *N* points in the *d*-dimensional case is of $O(d\,N^2)$ complexity. It needs to be noted that the DT is not easy to implement in the the *d*-dimensional space. There is also a severe problem how to smoothly interpolate scalar values in the *d*-dimensional space. The vast majority of interpolation techniques rely on "separable" interpolations, i.e. interpolation is made in each axis independently expecting that the selection of axes order is arbitrary. Unfortunately such approaches lead to some artifacts and caused errors are unpredictable.

Radial basis function (RBF) interpolation belongs to non-separable interpolations used for interpolation in *d*-dimensional space. The computational cost of RBF increases non-linearly with the number of data processed and linearly with the dimensionality of the data set. The RBF interpolation is based on a distance of two points, i.e. the distance of two points $r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ is computed. The great advantage of RBF interpolation is that it does not need any tessellation of the data domain and simply supports the data of any dimensionality. RBF applications are quite widespread and can be found in data visualization, solutions of partial differential equations (PDE), neural networks, reconstruction of corrupted images etc.

The computational cost of the RBF interpolation is higher as the cost of tessellation is inheritably covered into the RBF interpolation in principle. Two significant aspects are connected with the RBF:

- re-computation of the RBF interpolation and
- reduction of the data set.

It should be noted that the RBF interpolation leads to a solution of linear system of equations (LSE) $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. The proposed approaches are valid for the *d*-dimensional case, but in the following text, $d = 2$ will be used for explanation.

II. RADIAL BASIS FUNCTION INTERPOLATION

RBF interpolation is quite simple from a mathematical point of view. It is based on a distance computing of two points in the *d*-dimensional space. RBF interpolation is defined by the function:

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \, \varphi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) = \sum_{j=1}^{N} \lambda_j \, \varphi_j(r_j)$$
$$r_j = \|\boldsymbol{x} - \boldsymbol{x}_j\|$$

It means that for the given data set $\{< x_i, h_i >\}_{i=1}^{N}$, where h_i are associated values to be interpolated and $\boldsymbol{x}_i$ are domain coordinates, a linear system of equations is obtained:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{N} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) \qquad i = 1, \dots, N$$

where λ_j are weights to be computed. Due to stability issues, usually a polynomial $P_k(\boldsymbol{x})$ of a degree *k* is added to the form, i.e.:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{N} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) \; + P_k(\boldsymbol{x}_i) \qquad i = 1, \dots, N$$

 ISBN: 978-1-61208-184-7

For a practical reason in many applications, the polynomial of the 1st degree is used, i.e. linear polynomial $P_1(\boldsymbol{x}) = \boldsymbol{a}^T\boldsymbol{x} + a_0$. Then the RBF interpolation function has the following form:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{N} \lambda_j \, \varphi\left(\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|\right) + \boldsymbol{a}^T\boldsymbol{x}_i + a_0$$

$$h_i = f(\boldsymbol{x}_i) \qquad i = 1, \ldots, N$$

and additional conditions are applied:

$$\sum_{j=1}^{N} \lambda_i = 0 \qquad \sum_{j=1}^{N} \lambda_i \boldsymbol{x}_i = \boldsymbol{0}$$

For the *d*-dimensional case and *N* points given, a system of $(N + d + 1)$ linear equations has to be solved.

For *d=2* vectors $\boldsymbol{x}_i$ and $\boldsymbol{a}$ are given as $\boldsymbol{x}_i = [x_i, y_i]^T$ and $\boldsymbol{a} = [a_x, a_y]^T$. Using the matrix notation we can write:

$$\begin{bmatrix} \varphi_{1,1} & \cdots & \varphi_{1,N} & x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{N,1} & \cdots & \varphi_{N,N} & x_N & y_N & 1 \\ x_1 & \cdots & x_N & 0 & 0 & 0 \\ y_1 & \cdots & y_N & 0 & 0 & 0 \\ 1 & \cdots & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ a_x \\ a_y \\ a_0 \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_N \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

$$\boldsymbol{a}^T \boldsymbol{x_i} + a_0 = a_x \, x_i + a_y \, y_i + a_0$$

It can be seen that for the 2-dimensional case and *N* points given a system of $(N + 3)$ linear equations has to be solved. It can be seen that the RBF interpolations are not "separable" by the definition, i.e. an interpolation over x-axis and then over y-axis and vice versa cannot be made.

The radial basis functions interpolation was originally introduced using multiquadric method [5] in 1971 called Radial Basis Function method. Since then, many different RBF interpolation schemes have been developed with some specific properties, e.g. [4] uses $\varphi(r) = r^2 lg\, r$, which is called Thin-Plate Spline (TPS), a function $\varphi(r) = e^{-(\epsilon r)^2}$ that was proposed in [9]. Compactly Supported RBF (CSRBF) was introduced in [13] as

$$\varphi(r) = \begin{cases} (1-r)^q \, P(r), & 0 \le r \le 1 \\ 0\,, & r > 1 \end{cases},$$

where: $P(r)$ is a polynomial function and *q* is a parameter.

Theoretical problems with stability and solvability were resolved by [6] and [14]. Generally, there are two main groups of the RBFs:

- "global" – a typical example is TPS function
- "local" – Compactly supported RBF (CSRBF)

If the "global" functions are taken, the matrix $\boldsymbol{A}$ of the LSE is full, for large *N* is becoming ill-conditioned and problems with convergence can be expected. On the other hand if the CSRBFs are taken, the matrix $\boldsymbol{A}$ is becoming relatively sparse, i.e. computation of the LSE will be faster, but the scaling factor needs to be carefully selected due to a limited influence of the CSRBF and the final function tends to be "blobby" shaped.

TABLE I. TYPICAL EXAMPLE OF "GLOBAL" FUNCTIONS

"Global" functions	$\phi(r)$
Thin-Plate Spline (TPS)	$r^2 \log r$
Gauss function	$\exp\left(-(\varepsilon r)^2\right)$
Inverse Quadric (IQ)	$1/\left(1+(\varepsilon r)^2\right)$
Inverse multiquadric (IMQ)	$1/\sqrt{1+(\varepsilon r)^2}$
Multiquadric (MQ)	$\sqrt{1+(\varepsilon r)^2}$

TABLE II. TYPICAL EXAMPLE OF "LOCAL" CSRBF FUNCTIONS

ID	**Function**
1	$(1-r)_+$
2	$(1-r)_+^3(3r+1)$
3	$(1-r)_+^5(8r^2+5r+1)$
4	$(1-r)_+^2$
5	$(1-r)_+^4(4r+1)$
6	$(1-r)_+^6(35r^2+18r+3)$
7	$(1-r)_+^8(32r^3+25r^2+8r+1)$
8	$(1-r)_+^3$
9	$(1-r)_+^3(5r+1)$
10	$(1-r)_+^7(16r^2+7r+1)$

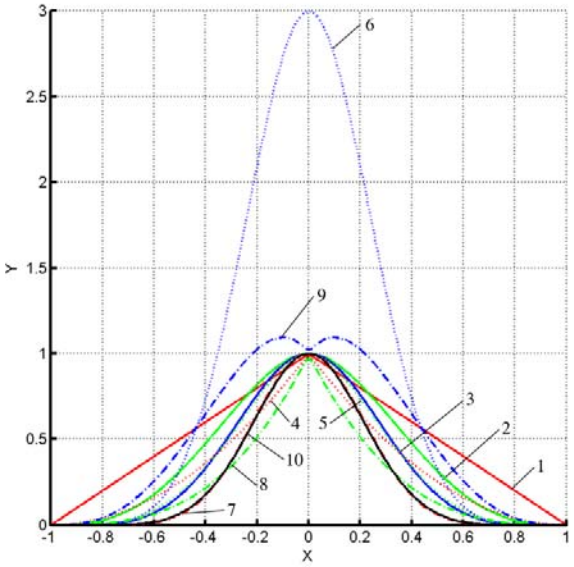


Figure 1. Geometrical properties of CSRBF

Tab. 2 presents typical examples of CSRBFs defined for the interval < 0 , 1 >, but for the practical use a scaling is used, i.e. the value *r* is multiplied by a scaling factor α, where 0 < α < 1. Fig. 1 presents the geometrical properties of typical CSRBFs.

 ISBN: 978-1-61208-184-7

III. INCREMENTAL RBF COMPUTATION

Some interesting problems can be solved using RBF interpolation quite effectively, e.g. surface reconstruction from scattered data [3][8][9][16], reconstruction of damaged images [11][15], inpainting removal [2][12] etc. All those applications based on RBFs interpolation have one significant disadvantage – the computational cost. This is especially severe in applications when the data are not static. Typical examples of non-static data are:

- Position of points has changed. It means that the whole system of linear equations has to be formed and recomputed which leads generally to $O(N^3)$ computational complexity and unacceptable time-consuming computation.
- Position of points remains fixed, but the value associated with a point has changed. In this case, iterative methods are usually faster than explicit computation of an inverse matrix.

In some applications a "sliding window" on data is required, especially in time-related applications when old data should not be used in the interpolation and new data are to be included. This is a typical situation in signal processing applications. Considering the above facts above there is a question how to compute RBF incrementally with a lower computational complexity.

The main question to be answered is:

Is it possible to use already computed RFB interpolation if a new point is to be included to the data set?

If the answer is positive it should lead to significant decrease of computational complexity. In the following, it will be presented how a new point can be inserted, how a selected point can be removed and also how to select the best candidate for a removal according to an error caused by this point removal.

Let us consider some operations with block matrices (assuming that all operations are correct and matrices are non-singular in general etc.).

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \\ -(\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & (\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \end{bmatrix}$$

Let us consider a matrix $\boldsymbol{M}$ of $(n+1) \times (n+1)$ and a matrix $\boldsymbol{A}$ of $n \times n$ in the following block form:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^T & c \end{bmatrix}$$

Then the inverse of the matrix $\boldsymbol{M}$ applying the rule above can be written as:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \left(\boldsymbol{A}-\frac{1}{c}\boldsymbol{b}\boldsymbol{b}^T\right)^{-1} & -\frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b} \\ -\frac{1}{k}\boldsymbol{b}^T\boldsymbol{A}^{-1} & \frac{1}{k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^{-1}+\frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b}\boldsymbol{b}^T\boldsymbol{A}^{-1} & -\frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b} \\ -\frac{1}{k}\boldsymbol{b}^T\boldsymbol{A}^{-1} & \frac{1}{k} \end{bmatrix}$$

where: $k = c - \boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b}$

We can easily simplify this equation if the matrix $\boldsymbol{A}$ is symmetrical as:

$$\boldsymbol{\xi} = \boldsymbol{A}^{-1}\boldsymbol{b} \qquad k = c - \boldsymbol{\xi}^T\boldsymbol{b}$$

$$\boldsymbol{M}^{-1} = \frac{1}{k}\begin{bmatrix} k\boldsymbol{A}^{-1}+\boldsymbol{\xi}\otimes\boldsymbol{\xi}^T & -\boldsymbol{\xi} \\ -\boldsymbol{\xi}^T & 1 \end{bmatrix}$$

where: $\boldsymbol{\xi}\otimes\boldsymbol{\xi}^T$ means the tensor multiplication of vectors and the result is a matrix.

All computations needed are of $O(N^2)$ computational complexity. It means that an inverse matrix can be computed incrementally with $O(N^2)$ complexity instead of $O(N^3)$ complexity required originally in this specific case. The structure of the matrix $\boldsymbol{M}$ is "similar" to the matrix of the RBF specification. The matrix $\mathbf{A}$ in the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is symmetrical and non-singular if appropriate rules for RBFs are kept.

Now, the question is how the incremental computation of an inverse matrix can be used for RBF interpolation?

A. Point Insertion

Let us assume a simple situation when the interpolation for N points has been computed and we need to include a new point into the given data set. A brute force approach of full RBF computation on the new data set can be used with $O(N^3)$ complexity computation.

If the RBF interpolation for $N+1$ points is considered, the following system of equations is obtained:

$$\begin{bmatrix} \varphi_{1,1} & \cdots & \varphi_{1,N} & \varphi_{1,N+1} & x_1 & y_1 & 1 \\ \vdots & \ddots & \cdots & \vdots & \vdots & \vdots & 1 \\ \varphi_{N,1} & \vdots & \varphi_{N,N} & \varphi_{N,N+1} & x_N & y_N & 1 \\ \varphi_{N+1,1} & & \varphi_{N+1,N} & \varphi_{N+1,N+1} & x_{N+1} & y_{N+1} & 1 \\ x_1 & \cdots & x_N & x_{N+1} & 0 & 0 & 0 \\ y_1 & \cdots & y_N & y_{N+1} & 0 & 0 & 0 \\ 1 & \cdots & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ \lambda_{N+1} \\ a_x \\ a_y \\ a_0 \end{bmatrix} = [h_1 \quad \cdots \quad h_N \quad h_{N+1} \quad 0 \quad 0 \quad 0]^T$$

where: $\varphi_{i,j} = \varphi_{j,i}$. Reordering the equations above we get:

$$\begin{bmatrix} 0 & 0 & 0 & x_1 & \cdots & x_N & x_{N+1} \\ 0 & 0 & 0 & y_1 & \cdots & y_N & y_{N+1} \\ 0 & 0 & 0 & 1 & \cdots & 1 & 1 \\ x_1 & y_1 & 1 & \varphi_{1,1} & \cdots & \varphi_{1,N} & \varphi_{1,N+1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_N & y_N & 1 & \varphi_{N,1} & \cdots & \varphi_{N,N} & \varphi_{N,N+1} \\ x_{N+1} & y_{N+1} & 1 & \varphi_{N+1,1} & \cdots & \varphi_{N+1,N} & \varphi_{N+1,N+1} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_0 \\ \lambda_1 \\ \vdots \\ \lambda_N \\ \lambda_{N+1} \end{bmatrix} = [0 \quad 0 \quad 0 \quad h_1 \quad \cdots \quad h_N \quad h_{N+1}]^T$$

The last row and the last column is "inserted". As RBF functions are symmetrical, the recently derived formula for iterative computation of the inverse function can be used directly. The RBF interpolation is given by the matrix $\boldsymbol{M}$ as:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^T & c \end{bmatrix}$$

where the matrix $\boldsymbol{A}$ is the RBF $(N+3) \times (N+3)$ matrix and the $(N+3)$ vector $\boldsymbol{b}$ and scalar value c are defined as:

$$\boldsymbol{b} = [x_{N+1} \quad y_{N+1} \quad 1 \quad \varphi_{1,N+1} \quad \cdots \quad \varphi_{N,N+1}]^T$$
$$c = \varphi_{N+1,N+1}$$

 ISBN: 978-1-61208-184-7

It means that it is possible to compute the *(N+1) × (N+1)* matrix $\boldsymbol{M}^{-1}$ if the $N \times N$ matrix $\boldsymbol{A}^{-1}$ is known with $O(N^2)$ complexity.

That is exactly what we wanted!

Now we have proved that the iterative computation of inverse function is of $O(N^2)$complexity offers a significant performance improvement for points insertion. It should be noted that some operations can be implemented more effectively, especially $\boldsymbol{\xi}\otimes\boldsymbol{\xi}^{\boldsymbol{T}} = \boldsymbol{A}^{-1}\boldsymbol{b}\boldsymbol{b}^{T}\boldsymbol{A}^{-1}$ as the matrix $\boldsymbol{A}^{-1}$ is symmetrical etc.

B. Point removal

In some cases it is necessary to remove a point from the given data set. It is actually an inverse operation to the insertion operation described above. Let us consider a matrix $\boldsymbol{M}$ of the size *(N+1) × (N+1)* as

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^{T} & c \end{bmatrix}$$

Now, the inverse matrix $\boldsymbol{M}^{-1}$ is known and we want to compute matrix $\boldsymbol{A}^{-1}$, which is of the size $N \times N$.

Recently, derived opposite rule:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^{T} & c \end{bmatrix}$$

$$\boldsymbol{\xi} = \boldsymbol{A}^{-1}\boldsymbol{b} \qquad k = c - \boldsymbol{\xi}^{\boldsymbol{T}}\boldsymbol{b}$$

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} + \frac{1}{k}\boldsymbol{\xi}\otimes\boldsymbol{\xi}^{\boldsymbol{T}} & -\frac{1}{k}\boldsymbol{\xi} \\ -\frac{1}{k}\boldsymbol{\xi}^{\boldsymbol{T}} & \frac{1}{k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{bmatrix}$$

It can be seen that:

$$\boldsymbol{Q}_{11} = \boldsymbol{A}^{-1} + \frac{1}{k}\boldsymbol{\xi}\otimes\boldsymbol{\xi}^{\boldsymbol{T}}$$

and, therefore,:

$$\boldsymbol{A}^{-1} = \boldsymbol{Q}_{11} - \frac{1}{k}\boldsymbol{\xi}\otimes\boldsymbol{\xi}^{\boldsymbol{T}}$$

Now there are known both operations, i.e. insertion and removal, with effective computation of $O(N^2)$ computational complexity instead of $O(N^3)$. It should be noted that vectors related to the point assigned for a removal must be in the last row and last column of the matrix $\boldsymbol{M}^{-1}$.

C. Point selection

As the number of points within a given data set could be high, the point removal might be driven by a requirement of removing a point causing a *minimal interpolation error*. This is a tricky requirement as there is probably no general answer. The requirement should include additional information which interval of $\boldsymbol{x}$ is to be considered.

Generally, we have a function:

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \varphi_j(\boldsymbol{x}) + P_k(\boldsymbol{x})$$

And we want to remove a point $\boldsymbol{x}_j$ which causes a minimal interpolation error ε_j, i.e.

$$f_i(\boldsymbol{x}) = \sum_{j=1, i\neq j}^{N} \lambda_j \varphi_j(\boldsymbol{x}) + P_k(\boldsymbol{x})$$

and the following should be minimized:

$$\varepsilon_i = \int_{\Omega} \left| f(\boldsymbol{x}) - f_j(\boldsymbol{x}) \right| d\boldsymbol{x}$$

where: Ω is the interval on which the interpolation is to be made. It means that if the point $\boldsymbol{x}_j$ is removed the error ε_i is determined as:

$$\varepsilon_i = \lambda_i \int_{\Omega} |\varphi(\|\boldsymbol{x} - \boldsymbol{x}_i\|)| d\boldsymbol{x}$$

As the interval Ω on which the interpolation is known, we can compute or estimate the error ε_j for each point $\boldsymbol{x}_j$ in the given data set and select the best one. For many functions φ, the error ε_j can be computed or estimated analytically as the evaluation of ε_j is simple, e.g.

$$\int r^m \ln \mathrm{r}\ dr = r^{m+1}\left[\frac{ln\, r}{m+1} - \frac{1}{(m+1)^2}\right] \quad m \neq -1$$

In particular, it means that for TPS function $r^2 \ln r$ the error ε_k is easy to evaluate. In the case of CSRBFs, the estimation is even simpler as they have a limited influence, so generally λ_j determines the error ε_j.

It should be noted that a selection of a point with the lowest influence to the interpolation precision in the given interval Ω is of $O(N)$ complexity only.

The above has shown a new approach to RBF computation which is convenient for larger data sets. It is especially convenient for t-varying data and for applications, where a "sliding window" needs to be used. Additionally basic operations – point insertion and point removal – have been introduced. These operations have $O(N^2)$ computational complexity only, which makes a significant difference from the original approach used for RBFs computation having $O(N^3)$.

IV. SCATTERED DATA RBF INTERPOLATION

The RBF interpolation relies on solution of a LSE $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ of the size $N \times N$ in principle, where N is a number of the data processed. If the "global" functions are used, the matrix $\boldsymbol{A}$ is full, while if the "local" functions are used (CSRBF), the matrix $\boldsymbol{A}$ is sparse.

However, in visualization applications it is necessary to compute the final function $f(\boldsymbol{x})$ many many times and even for already computed λ_i values, the computation of $f(\boldsymbol{x})$ is too expensive. Therefore it is reasonable to significantly "reduce" the dimensionality of the LSE $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Of course, we are now changing the interpolation property of the RBF to approximation, i.e. the values computed do not pass the given values exactly.

Probably the best way is to formulate the problem using the Least Square Error approximation. Let us consider the formulation of the RBF interpolation again.

 ISBN: 978-1-61208-184-7

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi\left(\left\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\right\|\right) + \boldsymbol{a}^T \boldsymbol{x}_i + a_0$$

$$h_i = f(\boldsymbol{x}_i) \qquad i = 1, \ldots, N$$

where: $\boldsymbol{\xi}_j$ are not given points, but points in a pre-defined "virtual mesh" as only coordinates are needed (there is no tessellation needed). This "virtual mesh" can be irregular, orthogonal, regular, adaptive etc. For simplicity, let us consider 2-dimensional squared (orthogonal) mesh in the following example. Then the $\boldsymbol{\xi}_j$ coordinates are the corners of this mesh. It means that the given scattered data will be actually "re-sampled", e.g. to the squared mesh.

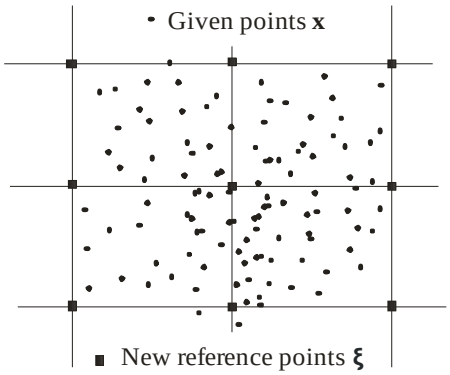


Figure 2. RBF approximation and points' reduction

In many applications the given data sets are heavily over sampled, or for the fast previews, e.g. for the WEB applications, we can afford to "down sample" the given data set. Therefore the question is how to reduce the resulting size of LSE.

Let us consider that for the visualization purposes we want to represent the final potential field in d-dimensional space by M values instead of N and $M \ll N$. The reason is very simple as if we need to compute the function $f(\boldsymbol{x})$ in many points, the formula above needs to be evaluated many times. We can expect that the number of evaluation Q can be easily requested at $10^2\,N$ of points (new points) used for visualization.

If we consider that $Q \geq 10^2\,N$ and $N \geq 10^2\,M$ then **the speed up factor in evaluation can be easily about 10^4** !

This formulation leads to a solution of a linear system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ where number of rows $N \gg M$, number of unknown $[\lambda_1, \ldots, \lambda_M]^T$. As the application of RBF is targeted to high dimensional visualization, it should be noted that the polynomial is not requested for all kernels of the RBF interpolation. But it is needed for $\varphi(r) = r^2 lg\, r$ kernel function (TPS). This reduces the size of the linear system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ significantly and can be solved by the Least Square Method (LSM) as $\boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{A}^T\boldsymbol{b}$ or Singular Value Decomposition (SVD) can be used.

$$\begin{bmatrix} \varphi_{1,1} & \cdots & \varphi_{1,M} \\ \vdots & \ddots & \vdots \\ \varphi_{i,1} & \cdots & \varphi_{i,M} \\ \vdots & \ddots & \vdots \\ \varphi_{N,1} & \cdots & \varphi_{N,M} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_M \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ \vdots \\ \vdots \\ h_N \end{bmatrix} \qquad \boldsymbol{Ax} = \boldsymbol{b}$$

The high dimensional data can be approximated for visualization by RBF efficiently with a high flexibility as it is possible to add additional points of an area of interest to the mesh. It means that a user can add some points to already given mesh and represent easily some details if requested. It should be noted that the use of LSM increases instability of the LSE in general.

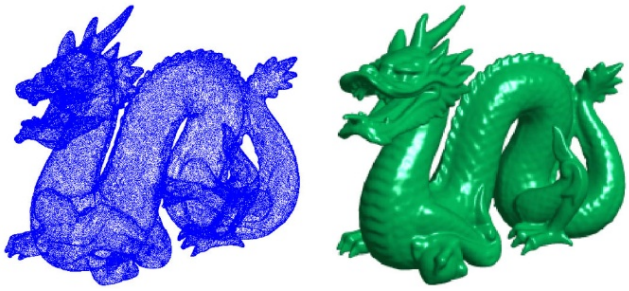
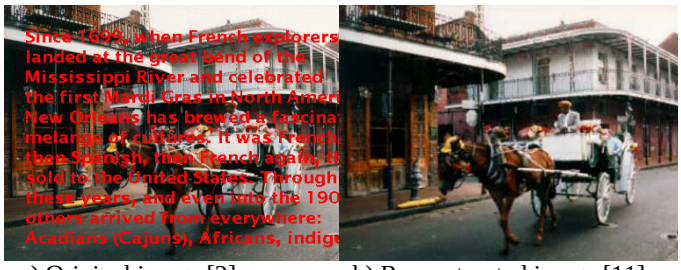


Figure 3. Surface reconstruction (438 000 points) [3]

Experimental evaluation

The RBF interpolation is a very powerful tool for interpolation of data in d-dimensional space in general. In order to demonstrate the functionality the RBF, we have recently used RBF for reconstruction of damaged images by a noise or by inpainting. Also a surface reconstruction has been solved by the RBF interpolation well. Fig. 3–5 illustrate the power of the RBF interpolation [2][3][8][15].



a) Original image [2] b) Reconstructed image [11]

Figure 4. Inpaited image reconstruction

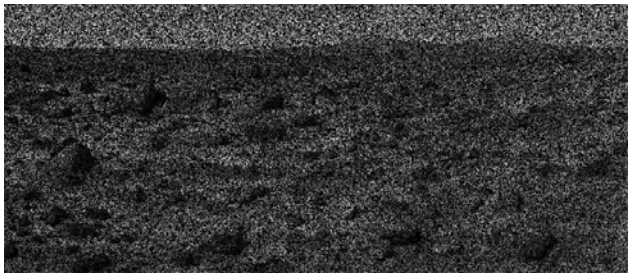


Figure 5a. Original image with 60% of damaged pixels [11]

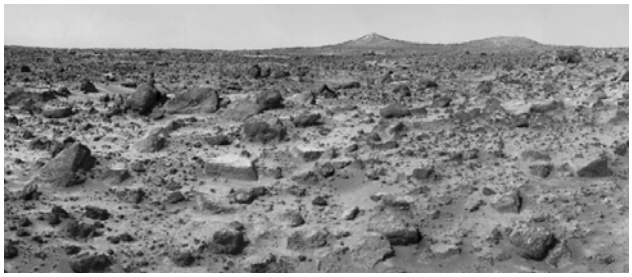


Figure 5b. Reconstructed image [11]

The RBF interpolation gives quite good results even if the images are heavily damaged. The advantages of RBF interpolation over the other interpolations have been proved even though that the RBF interpolation causes some additional computational cost as the RBF is primarily targeted for scattered data interpolation.

V. CONCLUSION

The radial basis functions (RBF) interpolation is a representative interpolation method for unordered scattered data sets. It is well suited approach for solving problems without meshing the data domain. RBF interpolations are used in many computational fields, e.g in solution of partial differential equations etc. RBF approach supports interpolation in the *d*-dimensional space naturally.

This paper describes an incremental computation of RBF and shows the decrease of the computational complexity from approx. $O(N^3)$ to $O(N^2)$ for a point insertion and a point removal.

It also presents a method for "resampling" the data processed as the approximation is acceptable in many applications, namely in visualization. The approach enables to increase details for visualization by adding new points to the "virtual mesh", if more details are needed. It is necessary to mention, that there is no mesh actually needed and only points of the "virtual mesh" need to be defined.

Future research should be devoted to the evaluation of computing precision and stability as the RBF interpolation generally leads to not well conditioned LSE. Also, there is a need to analyze, how the ratio $\nu = N/M$ can be controlled effectively and what can be expected in real and large data applications, e.g. from GIS fields, inverse engineering process in CAD/CAM etc.

ACKNOWLEDGMENT

The author thanks to colleagues at the University of West Bohemia (UWB) in Plzen and at the VSB-Technical University in Ostrava for their critical comments and constructive suggestions, to anonymous reviewers for their critical view and comments that helped to improve the manuscript. Special thanks belong to RongJiang Pan, Shandong University, China for recommendations during his stay at the University of West Bohemia (UWB), to former PhD and MSc. students at the UWB Vit Ondracka, Lukas Loukota, Jan Hobza, Karel Uhlir and Jiri Zapletal.

This research was supported by the Ministry of Education of the Czech Republic, projects ME10060 and LA10035.

REFERENCES

[1] B.J.Ch. Baxter, "The Interpolation Theory of Radial Basis Functions," PhD thesis, Trinity College, Cambridge University, U.K., 1992.

[2] M. Bertalmio, G. Sapiro, C. Ballester and V. Caselles, "Image Inpainting," Proceedings of SIGGRAPH'00, Computer Graphics, pp. 417-424, 2000.

[3] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum and T.R. Evans, "Reconstruction and Representation of 3D Objects with Radial Basis Functions," Computer Graphics (SIGGRAPH 2001 proceedings), pp. 67-76, 2001.

[4] J. Duchon, "Splines Minimizing Rotation-invariant Semi-norms in Sobolev space," in Constructive Theory of Functions of Several Variables, Springer Lecture Notes in Math, Vol. 21, pp. 85-100, 1977.

[5] L.R. Hardy, "Multiquadric Equations of Topography and other Irregular Surfaces", J. Geophysical. Res., Vol. 76, pp. 1905-1915, 1971.

[6] C.A. Micchelli, "Interpolation of Scattered Data: Distance Matrix and Conditionally Positive Definite Functions," Constr. Approx., No. 2, pp. 11-22, 1986.

[7] R. Pan and V. Skala, "Implicit Surface Modeling Suitable for Inside/Outside Tests with Radial Basis Functions," 10th International Conference on Computer Aided Design and Computer Graphics (CAD/Graphics), 2007.

[8] R. Pan and V. Skala, "A Two-Level Approach to Implicit Modeling with Compactly Supported Radial Basis Functions," Engineering and Computers, Springer Verlag, Vol. 27. No. 3., pp. 299-307, ISSN 0177-0667, 2011.

[9] R. Pan and V. Skala, "Continuous Global Optimization in Surface Reconstruction from an Oriented Point Cloud," Computer Aided Design, Vol. 43, No. 8, pp. 896-901, Elsevier, 2011.

[10] I.P. Schagen, "Interpolation in Two Dimension – A New Technique," IMA Journal of Applied Mathematics, Vol. 23, No. 1, pp. 53-59, 1977.

[11] K.Uhlir and V. Skala, "Radial Basis Function use for the Restoration of Damaged Images," in Computer Vision and Graphics, Dordrecht: Springer, pp. 839-844, 2006.

[12] Ch.C.L. Wang and T.-H. Kwok, "Interactive Image Inpainting using DCT Based Exemplar Matching," ISVC 2009, LNCS 5876, pp. 709-718, 2009.

[13] H. Wendland, "Computational Aspects of Radial Basis Function Approximation," in Topics in Multivariate Approximation and Interpolation (Ed.K. Jetter et al.), Elsevier B.V., pp. 231-256, 2005.

[14] G.B. Wright, "Radial Basis Function Interpolation: Numerical and Analytical Developments," University of Colorado, PhD Thesis, 2003.

[15] J. Zapletal, P. Vanecek and V. Skala, "RBF-based Image Restoration Utilising Auxiliary Points," CGI 2009 proceedings, ACM, pp. 39-44, 2009.

[16] Y. Ohtake, A. Belyaev and H.-P. Seidel, "3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs," Graphical Models, Vol. 67, No. 3, pp. 150-165, 2005.

WEB references

FastRBF: http://www.farfieldtechnology.com/. <retrieved: 2011-12-05>

 ISBN: 978-1-61208-184-7

Visual Data Mining Using the Point Distribution Tensor

Marcel Ritter
Graduate School for Scientific Computing
University of Innsbruck
Innsbruck, Austria
marcel.ritter@uibk.ac.at

Werner Benger
Center for Computation & Technology
Louisiana State University
Baton Rouge, USA
werner@cct.lsu.edu
Institute for Astro- and Particle Physics
University of Innsbruck
Innsbruck, Austria
werner.benger@uibk.ac.at

Biagio Cosenza
Distributed and Parallel Systems Group
University of Innsbruck
Innsbruck, Austria
cosenza@dps.uibk.ac.at

Keera Pullman
ESRI Australia
Darwin, NT, Australia
kpullman@esriaustralia.com.au

Hans Moritsch
Distributed and Parallel Systems Group
University of Innsbruck
Innsbruck, Austria
hans@dps.uibk.ac.at

Wolfgang Leimer
Distributed and Parallel Systems Group
University of Innsbruck
Innsbruck, Austria
wolfgang.leimer@student.uibk.ac.at

***Abstract*—We explore a novel algorithm to analyze arbitrary distributions of 3D-points. Using a direct tensor field visualization technique allows to easily identify regions of linear, planar or isotropic structure. This approach is very suitable for visual data mining and exemplified upon geoscience applications. It allows to distinguish, for example, power lines and flat terrains in LIDAR scans. We furthermore present the work on the optimization of the computationally intensive algorithm using OpenCL and potentially utilizing the Insieme optimizing compiler framework.**

Keywords-metric tensor; scientific visualization; point cloud; OpenCL.

I. Introduction

Point clouds occur as primary data sources in different scientific domains, e.g., stemming from simulations in computational fluid dynamics by smooth particle simulations or from observational methods, such as light detection and ranging (LIDAR) laser scanning [1]. Classification of point clouds is still ongoing research for LIDAR laser scan data [2]. Geometric information about the local point distribution can be used for classification, for constructing surfaces, or as basis for other algorithms. An algorithm to compute Gaussian and mean curvature on polygon meshes was presented in [3], based on the tensorial product of the polygon's normal vectors. A product with additional weights was used to compute the co-variance matrix of point neighborhoods describing tangential frames for surfaces in [4]. This co-variance matrix provides us with a type of smooth transition between lines, surfaces, and volumes [5].

In this article, we utilize the direct tensor visualization technique [6] to illustrate the co-variance matrix resulting from arbitrary point clouds. Section II introduces the distribution tensor and the utilized visualization technique, presented on simple geometric point distributions. Two algorithms for the tensor computation are described: One for central processing units (CPUs) and one for graphics processing units (GPUs). Optimizations are presented and the Insieme compiler optimization framework [7] is introduced. Our visualization method is demonstrated on two geo-scientific applications in Section III: On the analysis of LIDAR laser scan data and the analysis of coastlines. The paper concludes and describes future work in Section IV.

II. Computing the Point Distribution Tensor

A. Mathematical Background

We define the "point distribution tensor" as a measure constructed from of a set of N points $\{P_i : i = 1...N\}$ similar to the co-variance matrices in [4] [8] [5]:

$$S(P_i) = \frac{1}{N} \sum_{k=1}^{N} \omega_{ik} (t_{ik} \otimes t_{ik}^{\tau}) \qquad (1)$$

whereby $t_{ik} = P_i - P_k$ and an optional weighting function $\omega_{ik} := f(||P_i - P_k||, r, i)$. Here, r is an user specified distance or radius defining the neighboorhood of point P_i. The weighting function ω_{ik} is zero outside this radius. The distribution tensor is symmetric and positive definite such as the metric tensor [9] and, thus, yields three eigen-values when doing an eigen-analysis: $\lambda_3 \geq \lambda_2 \geq \lambda_1$. These are used to classify the tensor via three shape factors [10], characterizing the shape of a fitting ellipsoid of the point

 ISBN: 978-1-61208-184-7

neighborhood in barycentric coordinates, see Figure 1:

$$\begin{aligned} c_{linear} &= (\lambda_3 - \lambda_2) \quad /(\lambda_1 + \lambda_2 + \lambda_3) \\ c_{planar} &= 2(\lambda_2 - \lambda_1) \quad /(\lambda_1 + \lambda_2 + \lambda_3) \\ c_{spherical} &= 3\lambda_1 \quad /(\lambda_1 + \lambda_2 + \lambda_3) \end{aligned} \tag{2}$$

with $c_{linear} + c_{planar} + c_{spherical} = 1$. A tensor field visualization method more suitable for large data than drawing tensor ellipsoids is utilized. Instead of ellipsoids textured splats are rendered with smooth transitions in color, orientation, texture and transparency, as shown in Figure 1.

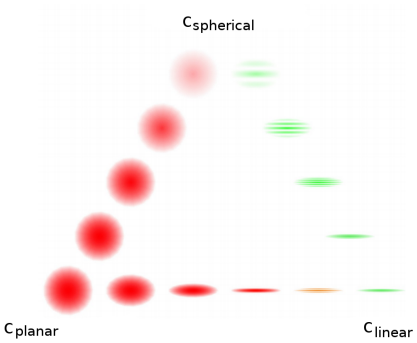
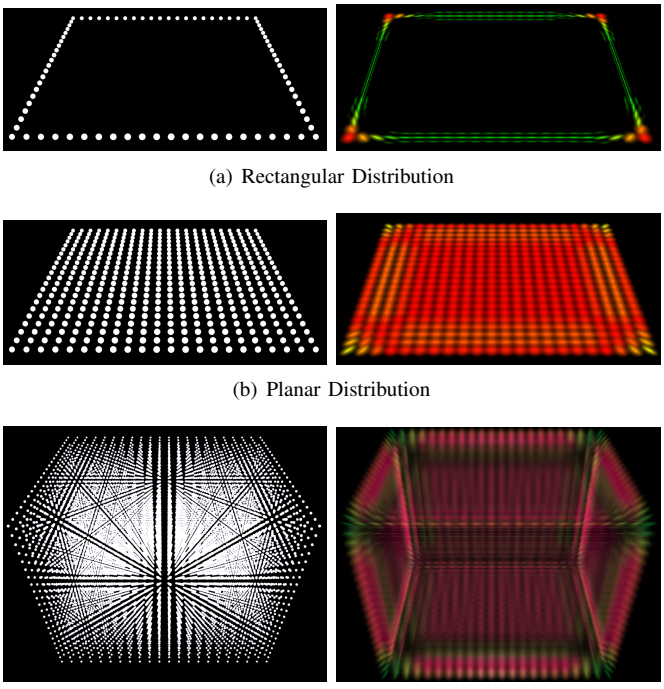


Figure 1. Tensors are visualized as textured oriented disks. The three shape factors, Equation 2, are used for smooth transitions between linear (right), planar (left), and spherical (top) shape [9]. In this context, the disks enhance the visualization of points predominantly distributed on a line, on a surface, or in a volumetric distribution.



(a) Rectangular Distribution

(b) Planar Distribution

(c) Volumetric Distribution

Figure 2. *Left:* Analytic point distributions illustrated by simple point rendering. *Right:* Corresponding distribution tensor fields. Linear 1D, planar 2D and isotropic 3D tensors are visualized using tensor splats, having a dominant linear, planar and spherical shape factor, respectively.

B. Test Cases

Simple analytic test cases were used to verify and study the properties of the distribution tensor, as illustrated in Figure 2. The point distributions have an extent of 1.0 in spatial dimensions and have been computed using a neighborhood radius of $r = 0.2$. Figure 2 (a) shows linear tensor splats textured and oriented in one direction. At the corners of the rectangle tensors become planar caused by two equally dominant directions in the neighborhood. Homogeneous distributions are fully transparent and become invisible, as demonstrated in Figure 2 (c). Here, the inner region is transparent, the border surfaces become more planar and are colored red while corner points become linear (green).

C. Algorithm

A first serial algorithm was implemented in the visualization shell VISH [11] utilizing C++ and OpenGL. Computation and visualization tasks were split in different modules. The computation module searches for neighbors in a 3D KD-Tree [12] within a user specified radius (where $\omega_{ik} > 0.0$) to limit the number of considered points. Alternatively to setting the radius, also the number of neighbors can be specified. Furthermore, a scalar field given on the points can be utilized to set the radius or number of points for each point individually. Eqn. 1 is utilized to compute the distribution tensor for each point. Different weighting functions have been implemented inside the neighborhood: $\omega_{ik} := (r - ||P_i - P_k||)/r$, $\omega_{ik} := 1/r$ and $\omega_{ik} := 1/r^2$.

Data is represented in an unified data model [9] [13] which allows support of different types of grid geometries and topologies. The computation algorithm operates on the vertices of any grid type. In the following applications point clouds and sets of lines are used for analysis.

D. GPU implementation

An alternative implementation of the algorithm was done in OpenCL [14], a framework for multicores and parallel hardware being able to execute programs also across heterogeneous platforms. The neighborhood is controlled by a fixed radius. Instead of the KD-Tree, an uniform grid was preferred as data structure to speed-up the neighborhood search. Here, the *loose grid* approach was adapted, where each particle is assigned to one cell based on its position. The grid's cell size depends on the influence radius (i.e., radius$\geq$cell size). Therefore, each particle can affect the closest 27 cells while calculating the tensor. This method allowed to bin the particles into the cells and to sort them by their grid index. The algorithm comprises four steps:

1) for each particle a hash value is computed, i.e., the cell index where it is located;
2) particles are sorted by hash; for this step NVidia's optimized bitonic sorting [15] is utilized;

 ISBN: 978-1-61208-184-7

3) the sorted list is used to compute the starting cell where the particle is located, running a thread for each particle, and performing scattered memory writes;
4) tensor calculation: Each particle searches the closest 27 grid cells from its location and it computes the tensor with each of the particles in these cells.

Steps $1-3$ are related to the build process of the grid data structure. The sorting algorithm is highly effective because it improves the memory access coherency when calculating the tensor, and reduces thread divergence (particles in the same thread group tend to be close together in space).

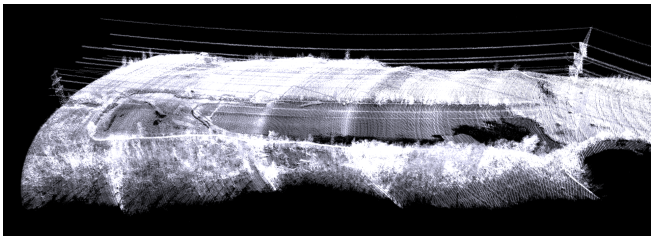
E. Optimization

The CPU algorithm was parallelized using OpenMP [16], adding a minimal overhead in development. Furthermore, OpenMP, as also OpenCL is supported by Insieme [7].

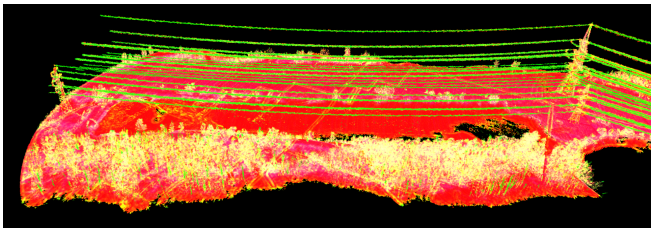
The Insieme compiler, under development at the University of Innsbruck, is a source-to-source compiler for C/C++ aiming at the automatic optimization of parallel programs implemented with MPI, OpenMP or OpenCL. It optimizes the source code for a specific platform (e.g., NVidia Fermi architecture), and applies transformations such as loop enrolling and collapsing, thread merge and data pre-fetching. Insieme aims at supporting programmers in effectively optimizing programs across different architectures, including shifting of computations from CPU to GPU cores. Optimizations are performed at compile-time through code analysis and transformations for sequential and parallel code regions. An intermediate representation is facilitated which explicitly describes parallelism, synchronization, and communication. The program's behavior is optimized and customized to the available hardware resources at runtime by utilizing statistical machine learning techniques based on a performance analysis database. Performance measures are, e.g., execution time, energy consumptions, and computing costs. Preliminary tests will be done with this code which is now part of the Insieme test cases.

III. Application Results

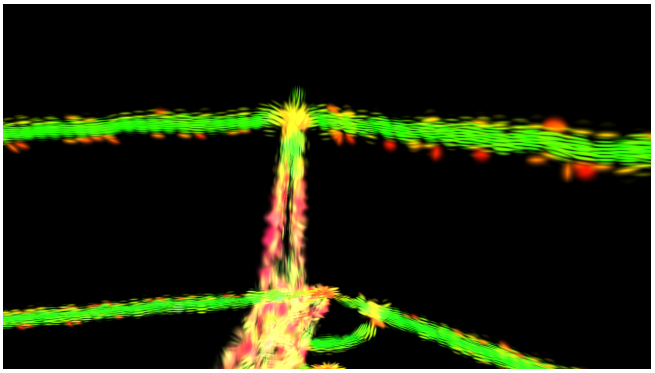
The method was applied to two different geoscience applications. Figure 3 shows a scan of a water basin close to the Danube in Austria captured with the Riegl hydrographic laser scanner VQ-820G [17]. In the example, a fixed neighborhood radius of two meters and a constant weighting function $\omega_{ik} = 1$ showed good results. Other parameters for r and ω_{ik} have been tested as well. Figure 3 (a) illustrates the received and processed laser echoes as a cloud of points; (b) and (c) show the distribution tensor field. Linear structures, such as the power cables, are well identified (green). The ground is dominantly planar (red). Some regions of the ground fade to magenta indicating less planarity. Here, grass influences the planar tensor to become more isotropic. Bushes and trees are isotropic or of an interpolated intermediate shape, mostly appearing



(a) LIDAR Echos



(b) LIDAR Tensors $r = 2.0m$



(c) LIDAR Tensors $r = 2.0m$ Detail

Figure 3. Distribution tensor field of returned laser echoes from an airborne laser scan. Linear distributions such as cables and planar distributions such as ground are emphasized. Vegetation is fading to spherical.

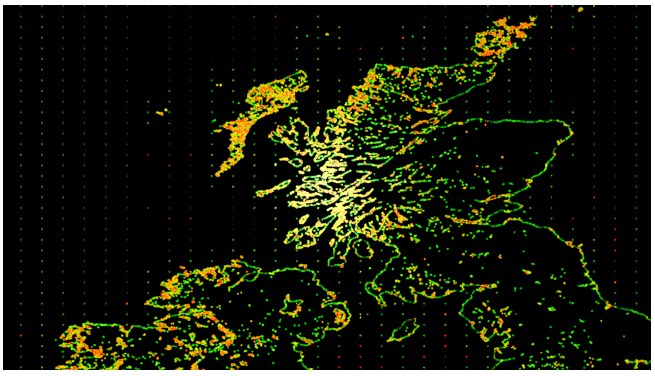


Figure 4. Distribution tensor field of an ESRI shapefile of the earth's water bodies and coastlines. The distribution tensor field with 12 fixed neighbors of the northern part of the United Kingdom is illustrated.

 ISBN: 978-1-61208-184-7

yellow. The computation with the OpenMP version utilizing 4 threads of the 4.81mio points with approximately 600 neighbors per point ($r = 2.0m$) took 752 seconds on a i7 M640 2.8GHz with 7.7GB RAM and NVidia Quadro FX3800M using Linux64bit, gcc 4.4.5, and Vish SVN 3854.

Another application was the analysis of coast and contour lines. Shapefiles [18] of water bodies and coastlines were investigated. Figure 4 (b) shows the distribution tensor field of the coast of the United Kingdom. Unstructured coastlines are highlighted in green whereas cliffy coast lines are shown in red, for example when looking at the northern coast of Scotland. Here, a rather small neighborhood of fixed 12 points turned out to emphasize cliffy coasts.

IV. Conclusion and Future Work

A new method of enhancing the visualization of point distributions was introduced, described, and demonstrated. Two different implementations and parallelization approaches were presented. Using an unified data model opened the possibility to apply the technique to data sets stemming from two different scientific applications: The visual extraction of power cables in LIDAR data and the visual enhancement of cliffy coastlines. We will further use the tensor analysis on LIDAR data to enhance point classification and the creation of digital terrain models. Different weighting functions and parameter studies will be investigated on more datasets. We ultimately will use the Insieme framework to optimize our parallel GPU and OpenMP codes.

Acknowledgment

Thanks to Frank Steinbacher for providing the LIDAR data sets. This work was supported by the Austrian Research Promotion Agency (FFG) *Airborne Hydromapping*, the Austrian Science Foundation FWF DK+ project *Computational Interdisciplinary Modeling* (W1227), and grant P19300. This research employed resources of the Center for Computation and Technology at Louisiana State University, which is supported by funding from the Louisiana legislatures Information Technology Initiative. This work was supported by the Austrian Ministry of Science BMWF as part of the UniInfrastrukturprogramm of the Forschungsplattform Scientic Computing at LFU Innsbruck.

References

[1] E. P. Baltsavias, "Airborne laser scanning: existing systems and firms and other resources," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 54, pp. 164–198, 1999.

[2] P. Dorninger, B. S. A. Zamolyi, and A. Roncat, "Automated Detection and Interpretation of Geomorphic Features in LiDAR Point Clouds," no. 99, pp. 60–69, 2011.

[3] G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," in *Proceedings of the Fifth International Conference on Computer Vision*, ser. ICCV '95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 902–.

[4] M. Alexa, S. Rusinkiewicz, M. Alexa, and A. Adamson, "On normals and projection operators for surfaces defined by point sets," in *In Eurographics Symp. on Point-Based Graphics*, 2004, pp. 149–155.

[5] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 11, pp. 1114 –1116, nov 1994.

[6] W. Benger and H.-C. Hege, "Tensor splats," in *Conference on Visualization and Data Analysis 2004*, vol. 5295. Proceedings of SPIE Vol. #5295, 2004, pp. 151–162.

[7] DPS Group at Universität Innsbruck, "The insieme compiler project." [Online]. Available: http://www.dps.uibk.ac.at/insieme/

[8] A. Adamson, "Computing curves and surfaces from points," Ph.D. dissertation, TU Damrstadt, 2008.

[9] W. Benger, "Visualization of general relativistic tensor fields via a fiber bundle data model," Ph.D. dissertation, FU Berlin, 2004.

[10] C. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. Jolesz, "Geometrical diffusion measures for mri from tensor basis analysis," in *Proceedings of ISMRM, Fifth Meeting, Vancouver, Canada*, Apr. 1997, p. 1742.

[11] W. Benger, G. Ritter, and R. Heinzl, "The Concepts of VISH," in 4^{th} *High-End Visualization Workshop, Obergurgl, Tyrol, Austria, June 18-21, 2007*. Berlin, Lehmanns Media-LOB.de, 2007, pp. 26–39.

[12] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematics Software*, vol. 3, no. 3, pp. 209–226, September 1977.

[13] M. Ritter, "Introduction to HDF5 and F5," Center for Computation and Technology, Louisiana State University, Tech. Rep. CCT-TR-2009-13, 2009.

[14] KHRONOS Group, "OpenCL," 2011. [Online]. Available: http://www.khronos.org/opencl

[15] K. E. Batcher, "Sorting networks and their applications," in *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, ser. AFIPS '68 (Spring). New York, NY, USA: ACM, 1968, pp. 307–314.

[16] OpenMP Architecture Review Board, "OpenMP," 2011. [Online]. Available: http://openmp.org

[17] F. Steinbacher, M. Pfennigbauer, A. Ulrich, and M. Aufleger, "Vermessung der Gewässersohle - aus der Luft - durch das Wasser," in *Wasserbau in Bewegung ... von der Statik zur Dynamik. Beitrge zum 15. Gemeinschaftssymposium der Wasserbau Institute TU München, TU Graz und ETH Zürich*, 2010.

[18] ESRI, "ESRI Shapefile Technical Description," Environmental Systems Research Institute, Inc, White Paper, July 1998.

 ISBN: 978-1-61208-184-7

3D Visualizations for Supporting Social Awareness in Learning Communities

Ekaterina Prasolova-Førland
Norwegian University of Science and Technology
Trondheim, Norway
e-mail: ekaterip@idi.ntnu.no

***Abstract* - Establishing and nurturing vibrant learning communities is seen as a highly complex process. An important concept in this context is that of social awareness. In this paper, we discuss supporting social awareness by 3D visualization of social networks, activities and resources. We compare social awareness support provided by virtual environments of Active Worlds and Second Life, outlining directions for future work.**

Keywords-3D visualizations; social awareness; 3D virtual worlds; learning communities

I. INTRODUCTION

The goal of this research is exploring the possibilities of 3D visualizations for supporting social awareness in learning communities. Communities are fluid and emergent [1], as opposed to well-defined groups. It might therefore be difficult for community members to get an overview of the existing social structures, activities and available resources in a learning community. This is a problem because awareness of, e.g., experience distribution and social ties creates occasions for knowledge sharing, facilitating search for cooperation partners. On the opposite, lack of this awareness creates continuous breakdowns in the flow of knowledge and, as a consequence, impacts negatively on learning. We use the term social awareness to indicate awareness of the social situation in a group or community in a shared environment, which can be physical, virtual or both. This awareness includes knowledge on learners' resources, activities and social connections. We distinguish between short-term and long-term, synchronous and asynchronous social awareness [2].

There are various mechanisms for promoting social awareness in everyday life, like chance encounters, message boards, verbal and non-verbal cues [3]. These techniques are not always sufficient due to a number of reasons, such as physical distances between the students, social fears and inhibitions and available spaces that are not optimal for meeting, working and information sharing. Various groupware tools have been used to promote awareness, overcoming the limitations of everyday modalities of interactions, including Twitter, Facebook, Skype and other [4]. Still, these tools mainly focus on supporting already established groups and networks rather than fluid communities [5].

3D virtual worlds have promising potential for supporting social awareness in learning communities. There are a number of reasons for that. First, 3D visualization is a powerful tool for supporting understanding of complex concepts, including social aspects, and is widely used in educational context [2, 6]. Second, 3D virtual worlds provide a constructivist and flexible learning environment where learners can collaboratively construct their understanding by exploring, building and sharing their experiences with peers and forming the environment according to the current needs of the learning community. Third, 3D virtual worlds provide a social arena where students, teachers and other stakeholders can meet and interact overcoming distances and different time zones, allowing supporting social awareness in a synchronous manner [7].

On the longer term, virtual spaces become a container of artifacts used by the users for their daily social and educational activities, and traces left by community members as a result of their participation. The places that serve as triggers and repositories of community memory in real life cannot serve as a permanent reference for community members since such places cannot be accessed any time and because the memories they keep have to be replaced regularly due to the limited amount of available space. The use of virtual places, on the other hand, allows such places with traces to be saved before they are cleaned up or removed, in this way supporting long-term/asynchronous social awareness [8, 9].

In this paper, we will discuss how to 3D visualizations can be used to support social awareness. The discussion will be illustrated by 2 case studies performed by the author in the virtual environments of Active Worlds [10] and Second Life [11]. By comparing social awareness support provided by these 2 environments, as well as alternative approaches, we outline directions for future work.

II. SUPPORTING SOCIAL AWARENESS WITH 3D VISUALIZATION IN ACTIVE WORLDS AND SECOND LIFE

Example 1 (Active Worlds). In order to support social awareness, we have created a virtual world called Viras in the virtual world of Active Worlds [10]. Viras is based on the metaphor of 'Archipelago': a virtual world consisting of sea and islands and groups of islands (Fig. 1). We have arrived at this metaphor after analyzing various spatial metaphors used in educational virtual worlds, trying to combine different features in one system in order to achieve sufficient flexibility [2, 8]. One of the goals behind this metaphor was to re-create the way in which communities and groups naturally are created and developed. Islands represent groups

 ISBN: 978-1-61208-184-7

and individuals, their constellations into archipelagos are communities, and the links, bridges and roads serve as connections between them. Also, we wanted to create a landscape with a high degree of overview, especially from the 'bird's-eye view', of the existing structures by clear distinction of borders and units of community building against the 'sea' background, thus promoting awareness of the community development. In addition to visualizing the social structures, the resources and activities of the community members are visualized in the virtual places where these activities take place (Fig. 2)

Example 2 (Second Life). The second example is a Virtual Research Arena being constructed at the Virtual Campus of Norwegian university of Science and Technology in Second Life [11], presenting a number of research projects performed at our university as well as student projects (Fig. 3). The virtual research arena serves as a meeting point for the communities of students, researchers and general public, sustaining awareness of research activities in these communities. It has been used as a venue for a number of community events, such as a science fair, international seminars and project presentations (Fig. 4). While developing the Virtual Research Arena, we have explored innovative ways of capturing, storing and mediating community knowledge through 3D creative visualizations and role-plays. The 3D constructions capturing the knowledge and experiences acquired by different generations of students and researchers will be stored in a virtual 'project gallery' constituting the community repository [9].

This repository and the community meeting areas contain a number of boundary objects that have been collaboratively created in order to facilitate the exchange of ideas between communities of students, researchers and practitioners. These boundary objects contribute to establishing a common ground, shared understanding and vocabulary among community members by to a significant degree taking advantage of visual symbols, interactive elements and aesthetics elements [9].

In Table 1, we provide an overview and compare social awareness mechanisms in Active Worlds and Second Life (based on the presented 2 studies), along the dimensions of *learner*, *place*, *artifacts*, following a characterization framework developed earlier by the author [2].

Figure 1. 3D visualization of social structure of a learning community as 'Archipelago'

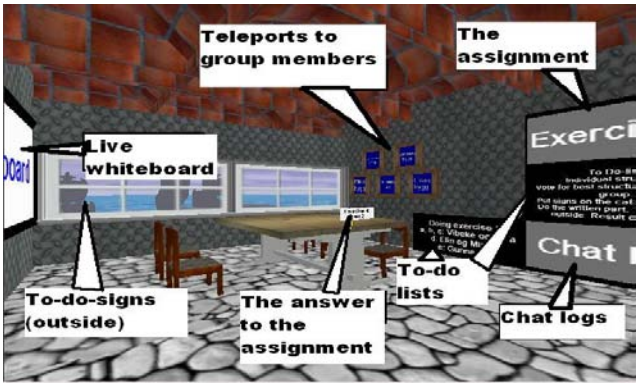


Figure 2. A virtual place with visualizations and traces of students' discussions and activities.

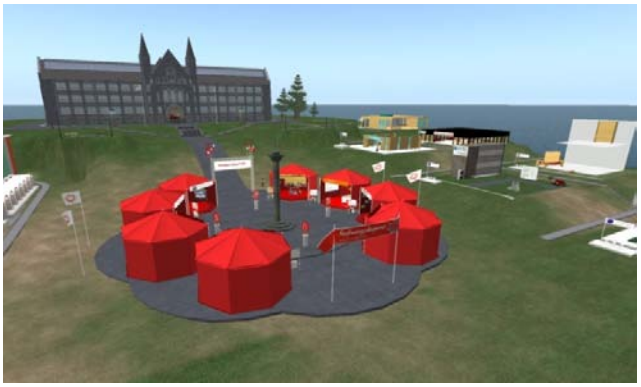


Figure 3. Virtual Research Arena as 3D visualization of a learning community's activities and research projects

Figure 4. Examples of 3D visualizations of community's activities: Virtual Science Fair, exhibition booths presenting research projects and seminars

III. DISCUSSION

The overview of social awareness mechanisms in Table 1 shows that Second Life in general provides equivalent or better support. For both virtual environments, usefulness for long-term social awareness support depends on the extent of usage and applies to active users only. Short-term awareness can only be supported with a relatively large amount of users online, and to a limited degree. This discussion raises a number of issues in connection with the suitability of 3D

 ISBN: 978-1-61208-184-7

TABLE 1. COMPARING SOCIAL AWARENESS SUPPORT IN ACTIVE WORLDS AND SECOND LIFE

Active Worlds (Viras/Archipelago)	**Second Life (Virtual Campus/Virtual Research Arena)**
Learner	
• Virtual environment provides rich possibilities for conveying of awareness information and identity expression through creation of places and artifacts • Virtual environment provides alternative means of communication and conveying of awareness through chat and 3D content brainstorming, 3D movement and manipulation of 3D artifacts and places	
• Limited selection and rotation of avatars makes identification of users and conveying of short-term social awareness difficult. From AW version 4.2, some avatar customization features are provided • Complexity of movement and communication in 3D space limits the use of existing awareness mechanisms • Search for collaboration partners with needed resources is complicated, especially due to the fact that Active Worlds consists of a number of un-connected universes	• Extensive possibilities for avatar (appearance and clothing) modifications, such as in in-built avatar editor and acquiring avatar features on the Linden market makes possibilities for social awareness support more prominent • Second Life features such as advanced camera movement, Skype-like, mini-map and teleportation features facilitate exchange of awareness information, but it is still restricted • Second Life's in-built search possibilities makes search for collaboration partners, i.e., members of different communities, easier. Additional search elements can be implemented by scripting, i.e., 'virtual project gallery' [9]
Place	
• Flexibility of building is beneficial for awareness and allowed fast development of the world • Uncritical use of provided facilities for flexible building may limit creativity and thus awareness expression (especially uncritical buying of pre-made objects in Second Life) • A repertoire of places of different types is beneficial for social awareness, allows a quick creation of an initial set of places and mediation of a range of activities (especially pre-made objects in Second life, commercially available and provided in the virtual campus and virtual research arena [9]) • 3D places can convey awareness of activities and learners by keeping traces and by mediating activities in a way, similar to real-life (though leaving traces is restricted due to rights and property managements) • Place structures are useful for visualizing social structure on the group level	
• Existing flexibility of building is still not fully sufficient for reflecting social structures on the community level with the increased size of the world • The additional complexity associated with leaving traces explicitly and still limited repertoire of places makes other tools more appropriate in a number of tasks • The appearance (Archipelago) can serve as a distraction for learning activities	• In-built map features provide a comprehensive overview of community development. Still, such development and its flexibility is strictly limited by land ownership and building rights issues • Leaving of traces is restricted by rights management mechanisms, but adjustments are possible through programming. Repertoire of places is rather extensive, with a variety of free and for-sale objects in the universe • The appearance of a science fair and campus provides an appropriate atmosphere for learning and research activities, but at the same time serves as an informal socializing place
Artifact	
• 3D artifacts provide rich possibilities for conveying awareness information about learners and their activities, communication and activity mediation in a visual, symbolic and 'real life' way • The effort associated with expressing awareness information in 3D artifacts makes other tools more preferable in some cases	
• The functionality associated with 3D artifacts does not cover the whole range of user needs	• Second Life has extensive functionality for programming and designing artifacts, including integration with external tools such as Sloodle, but the associated complexity makes it rather difficult to use in some cases

 ISBN: 978-1-61208-184-7

virtual worlds in general for social awareness support and learning facilitation, compared to other tools.

For example, support for finding collaboration partners in 3D virtual worlds may for the moment appear less straightforward compared to the more traditional 'yellow pages'/database approach [12] and social media such as Facebook and Twitter. Generally, simpler text-based communityware applications can in some cases be more efficient for exchange of information and 'matchmaking' of community members [13]. At the same time, as shown by our results and in the related literature, 3D virtual worlds can provide possibilities the 'traditional' tools cannot give. This includes 3D visualization of both awareness-related and educational information and an alternative arena for social and learning activities [2, 8, 9]. The facilities provided by the 'traditional' tools, can with some effort be integrated with the virtual worlds, e.g., more effective operations on documents. An example is the Sloodle initiative that focuses on integrating functionalities of Moodle learning management system and Second Life [14]. Therefore, 3D virtual worlds can potentially provide nearly all functionalities necessary for performance of central social and learning tasks. However, these functionalities must be provided in a more user-friendly and effective way than is possible with most existing virtual worlds at the moment.

We believe that appropriate support for social awareness, especially in an educational and research context, could be obtained in the context of virtual research arenas like the one described in Section II, where students and researchers can participate in sharing and collaborative elaboration of scientific content. Such virtual research arenas contribute to increasing awareness of ongoing research projects by visualizing activities and resources available at the corresponding research groups, thus facilitating search for collaboration partners. By presenting research results in a visualized and interactive way, it could be popularized and presented to a wider audience in a more appealing manner. In addition, such virtual research arenas can be accessed by the students, researchers and general public at no cost across distances and different time zones.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we have focused on 3D visualizations of learning communities and associated resources, activities and social networks, comparing support for social awareness provided by virtual environments of Active Worlds and Second Life. We have also shortly discussed the advantages and disadvantages of 3D visualizations for social awareness support compared to alternative approaches. In order to provide adequate support for social awareness, there is a need to overcome the identified limitations of 3D virtual worlds. To fully exploit the strengths of different technologies, a hybrid solution will probably be optimal. Future work will therefore include developing a comprehensive framework and guidelines for 3D dynamic visualization of social structures and processes in a learning community. Additional issues to consider will include augmenting 3D virtual worlds with other awareness tools (e.g., mobile devices, social media etc.) and combining 3D visualizations for educational and social purposes.

ACKNOWLEDGMENT

Many thanks to Mikhail Fominykh, Monica Divitini and Leif Martin Hokstad for their contributions to research presented in this paper.

REFERENCES

[1] E. Wenger, Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press, 1999.

[2] E. Prasolova-Førland, "Virtual Spaces as Artifacts: Implications for the Design of Educational CVEs", Special issue on "Cyberworlds and Education" in The International Journal of Distance Education Technologies (JDET), 2(4):94-115, Oct-Dec. 2004.

[3] A. Huxor, "The Role of the Personal in Social Workspaces: Reflection on Working in AlphaWorld". Collaborative Virtual Environments., Springer-Verlag London Ltd, 2001.

[4] C. Gutwin, S. Greenberg, & M. Roseman, "Workspace Awareness in Real–Time Distributed Groupware: Framework, Widgets and Evaluation". Proc. HCI 1996, London, UK, Springer–Verlag, 1996, pp. 281–298.

[5] E. Prasolova-Førland and L. M. Hokstad, "Supporting Learning Communities through a Lifecycle in a Serious Games Context: Requirements for Social Tools", Proc. Computers and Advanced Technology in Education (CATE 2009), November 22-24, 2009, St. Thomas, US Virgin Islands, ACTA Press, 2009

[6] M. Czerwinski, M. van Dantzich, G. G. Robertson, & H. Hoffman, "The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3D". Proc. Interact 1999, Edinburgh, Scotland, 1999, pp. 163-170.

[7] S. Clark & M. L. Maher, "The Role of Place in Designing a Learner Centered Virtual Learning Environment", Proc. CAAD Futures 2001, Eindhoven, The Netherlands, 2001.

[8] E. Prasolova-Førland, "A Repository of Virtual Places as Community Memory: an Experience of Use", Proc. ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI 2004), Singapore, 16-18 June 2004, pp. 225-228.

[9] M. Fominykh and E. Prasolova-Førland: "Virtual Research Arena: Presenting Research in 3D Virtual Environments," Proc. the 2nd Global Conference on Learning and Technology (Global Learn Asia Pacific), Melbourne, Australia, March 28-April 1, 2011, Chesapeake, VA: AACE, 2011, pp. 1558-1567.

[10] ActiveWorlds: www.activeworlds.com (retrieved: Dec 2011)

[11] Second Life, www.secondlife.com (retrieved: Dec 2011)

[12] M. S. Ackerman & C. Halverson, "Considering an Organization's Memory", Proc. CSCW 1998, November 14-18, Seattle, Washington USA, pp. 39-48. ACM Press, 1998.

[13] M. Koch, "Community Support in Universities – The Drehscheibe Project", Proc. Communities and Technologies (C&T 2003), Kluwer Publishers, Amsterdam, The Netherlands, 2003, pp. 445-464.

[14] Sloodle, http://www.sloodle.org/moodle/ (retrieved: Dec 2011)

 ISBN: 978-1-61208-184-7

A new Robust Method of Line Detection in a Structured Light System

Hussam Yousef, Regis Huez, Laurent Hussenet and Michel Herbin
CReSTIC
University of Reims Champagne Ardenne
Reims, France
hussam.yousef@etudiant.univ-reims.fr, {regis.huez, laurent.hussenet, michel.herbin} @univ-reims.fr

***Abstract*— This paper presents an integrated 3D face scanning system using the structured light technique. A new pattern consisting of horizontal colored strips using the De_Bruijn sequence is designed in a manner that can ensure a minimal distance between any two strips of the same color. After illuminating the scene with this pattern, a first image is taken and used to obtain 3D information. To detect the strip centers in the captured image we use a smoothing Gaussian filter with a large kernel applied to the V component in the HSV color space. The size of this kernel is computed separately for each column. Then, a connection algorithm is used to connect the isolated points in the detected strips. The next step is to determine the colors of these detected strips. For this purpose we exploit the fact that the resulting strips are connected. Firstly, we use the H component of the HSV color space to determine the color of the easy sets of pixels in these strips. Then, we apply a region growing algorithm to assign the colors to the remaining pixels. Finally, each connected and colored part of strip is matched to a strip in the projected pattern and triangulated with its corresponding one in the projected pattern to generate the 3D model. Using the concept of connected strips we exploit the information from several columns to take the decision in each column, which enhances the robustness of the method used here versus the existed ones. A second image of the scene without illumination is obtained and used to add the texture to the reconstructed 3D model. Experiment results show an accurate 3D resolution with this technique.**

Keywords – Image Processing; Computer vision; 3D visualization; Structured light system.

I. INTRODUCTION

Recently 3D models are widely used. One of the first techniques in this field was the stereo vision. In this technique several images of the scene are taken from different points of view using a calibrated set of cameras. Each point of the scene is searched in the different images. Finally, triangulation step is applied to the detected points in the image to calculate their 3D position. The main drawback of stereo vision is the necessity of having landmarks [2].

Structured light systems are based on the association of a projector illuminating a scene with a coded pattern and a camera is used to capture one image of the illuminated scene. Each point of the coded pattern is determined in the captured image, and a triangulation step allows the attribution of the 3D positions. The first structured light systems used laser planes or laser dots scanning the object to be triangulated [7, 8]. These laser-based techniques allow a high resolution; nevertheless, mechanical operations will be required each time the laser changes its position.

Recently, the techniques based on a laser were replaced by a projector used to project a coded pattern. The image of the illuminated scene is captured and a 3D model is generated [9]. This technique can be separated into two basic groups [5]. In the first one, which is called time-multiplexing, a sequence of black and white patterns is projected on the scene; the drawback of this technique is that only static objects can be measured. The second one called one-shot technique uses only one colored pattern, and the unique position of each point is determined using its local neighborhood. In this context, it was shown [5] that a pattern using the De_Bruijn sequence achieves good results in terms of accuracy and resolution. A De_Bruijn sequence allows attributing a specific color to each line.

The generation of a coded color pattern using a De_Bruijn sequence can be achieved in two ways [2], with or without black gaps introduced between the colored lines (Fig. 1). The colored line projection generates colored stripes on the illuminated scene, therefore on the captured image. Both techniques require the detection of the strip centers to obtain a good 3D resolution after a triangulation step [7, 8, 9]. Using black gaps between strips gives the pattern more robustness against the variation of the ambient illumination.

In this paper, a structured light system using black gaps and applied to 3D face modeling is proposed (Fig. 1).

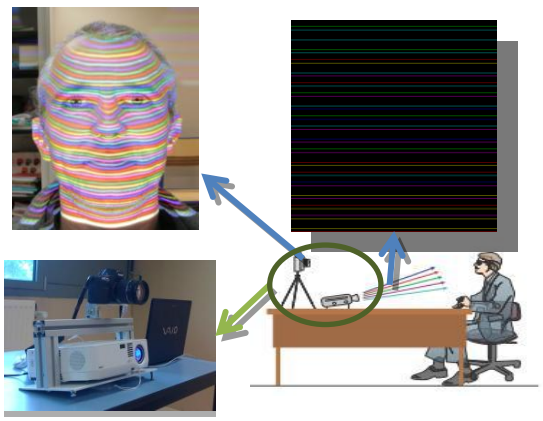


Figure 1. Layout of the system.

 ISBN: 978-1-61208-184-7

The part "Framework" will be firstly described, with a description of the used material. The second part named "location strip centers" concerns the determination of the center of the projected strips, this point is fundamental to obtain a sufficiently accurate 3D reconstruction. The following part talk about the line construction, where a global approach is taken contrary to a point to point approach. The last part "3D face model" presents the obtained results. A conclusion ends this paper.

II. FRAMEWORK

A. Overview

In a pattern consisting of horizontal colored lines, each column of the image can be seen as a 1D signal, its characteristics are: flat with bumps representing the projected lines. The determination of the projected lines requires information about the exact position of the bump centers. It also requires information on the colors of the determined centers, and finally a method to triangulate the centers with their corresponding points in the projected pattern.

The most popular methods to determine the pattern strips in structured light systems are based on the same principle. It consists in estimating on each column the possible candidates to be the centers of these lines and assigning them three probabilities:

- To be valid,
- To have a projected color,
- To be in a defined position in the used pattern.

The assignment of these points to specific points in the projected pattern [1, 2, 3] is carried out by methods of minimization of an objective function by dynamic programming. In this method, each column is treated separately without any consideration of the adjacent columns. It is obvious that the information from these columns could enhance the quality of the decision.

B. Method

We propose a global method to detect the centers of the strips and assign them to their corresponding points in the projected pattern. This method can be decomposed in four successive steps. In the first step, we determine the best position of the centers using a robust method, contrary to classic methods where a probability is given for all points. Secondly, the points are connected within their local neighborhoods. This procedure yields the entirely connected lines or at least long connected parts of the lines. The third and fourth parts are successively the color and lines assignments. In the last two steps, the detected strips are treated as an integrated unit instead of treating each one of its points alone.

C. Material

Fig. 1 shows the structured light system used, it consists of a « NEC NP410 LCD projector » with a full resolution of 1200*1600. It projects the colored multi-slit colored pattern on the face of the person at a distance of 0,4m. Then, the image of the illuminated scene is captured using a « Canon EOS 5D camera » with a resolution of 4368 by 2912 pixels. A polarized filter located before the camera can limit the problems of optical reflection and allows measurement taken in ambient light (Fig. 1). This system is controlled by a classic computer.

An offline calibration procedure is applied to synchronize the system and calculate the necessary projection matrices. The calibration procedure uses a dozen images of projected checkerboard on a plan containing a printed one. After the extraction of the corners in the projected and printed checkerboards, Zhang's method [10] is used to minimize the least square function over the needed parameters.

III. LOCATING STRIP CENTERS

A. Pattern generating

The patterns used in 3D reconstruction systems must be able to provide easy assignment of each point in the captured image to a point in the projected pattern. A minimal distance between any two strips having the same color must be kept to avoid unexpected connection of these similar strips in the captured image.

This application is dedicated to face reconstruction. Usually, the height of a face size is about 0,3m, so a pattern of 64 horizontal lines leads to a resolution of 4,6mm which is coherent with a face reconstruction. The 64-line pattern is achieved by a De_Bruijn binary sequence with a subsequence of six elements. This one is the starting point to construct a vector of 64 elements based on 6 different colors where the order of the colors is never repetitive. The colors are chosen to maximize the RGB color space. Two lines having the same color are separated by "at least" two lines of different color. The captured image is exploited in a RGB form.

B. Problem

The most famous way to locate strip centers is to search the peaks of strips in each separate column of each component RGB by fitting a 1D Gaussian profile in the neighborhood of the local maxima and searching its peak using the sub-pixel resolution [1, 2, 3]. With this method, two problems arise:

- In a color image, there is always an offset between the sub-pixel locations of the peaks in each RGB component, this problem is well known and is called RGB component misalignment [2, 4].
- The strips do not always have the same width, so two closed strips can be fitted by a unique Gaussian. On the other hand, a wide strip can be fitted by two Gaussians. The choice of the size of the Gaussian profile width should be carefully studied in order to ensure the result of this method.

In order to obtain a robust and performing solution, a global approach based on filtering is proposed. First of all a face shape detection step is applied to the image with the pattern to determine the region of interest, then we use a global approach to determine the strip centers using the V

 ISBN: 978-1-61208-184-7

component in the HSV color space. The V component is defined as:

$$V=\max(r,g,b) \quad (1)$$

This component has the biggest value of the three components in the RGB color space. In this way, we can get the center of the strip without considering color channels. Due to the RGB misalignment component, the V component signal can have some disturbed values at the strip centers.

In order to locate the strip centers, we use a smoothing filter with a big kernel; this filter smoothes the signal and focuses its peak in the middle. A common and powerful way to smooth a 1D signal is to convolute the signal with the Gaussian kernel.

Notice that a performing smoothing filter is similar to a low-pass filter with a low cut off frequency; it requires a big size of kernel, so a high value of the standard deviation. The size of the kernel k_s is defined as follows: the number of the nonzero values of the Gaussian kernel, the kernel is normalized and a truncation is used for all values lower to 0,01, k_s is proportional to the standard deviation. The influence of applying the Gaussian filter with different kernel sizes is depicted in Fig. 2. (Top) shows the intensities of the V component over a selected column, (middle) shows the same column after applying a Gaussian filter with a kernel of size 11, (below) shows the original column smoothed by a 21kernel size Gaussian filter.

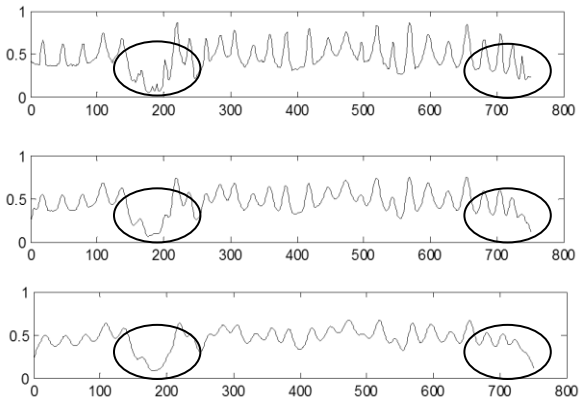


Figure 2. Application of kernels (size 11 and 21) on V component.

The larger the used kernel size the smoother the resulting signal, but a special attention should be paid to not confusing two successive strips and losing the relevant information as shown in Fig. 2 (below) when applying the kernel of size 21. To avoid this information lost, the size of the kernel must always be adjusted and controlled by the gaps between the strips in each scanned column.

Using this smoothing filter with a large kernel has two major advantages. The first one is that no small peaks correspond to the noise or disturbed strips could be found, while the second advantage is that this filter focuses the peaks of each strip exactly in the middle.

C. Determination of the Kernel size k_s

1) Algorithm

Due to the non-linearity of the scanned faces, the width of the gaps between the strips in the captured image is not always the same along the whole image. An important difference in size of these gaps can be found between the different columns. To yield the best results, a different kernel must be defined at each column according to the average of the gap sizes in this column.

Our contribution in this field is an algorithm dedicated to the determination of the kernel size for each separated column. On each column, the values of the pixels in the rising slope are summed up. The resulting values are associated to the positions of the relative peaks in the column. These values are normalized using the maximum of each column. A threshold is used so as not to take into account the low level values that are not representative of the strips. Fig. 3 describes two columns of the V component (dashed line), and the relative positions of the approximate peaks detected using this algorithm (solid line). The distances between each two successive peaks are calculated. To avoid the extreme abnormal values, the mean value of these distances is calculated and set as the size of the 1D kernel of the Gaussian filter that will be applied to the column.

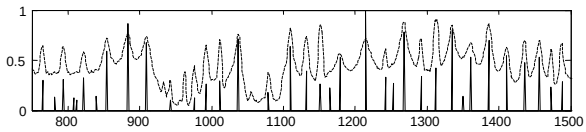


Figure 3. Approximate positions of the peaks in a column of V component.

The results present a variability of the gap between the peaks coming from several parameters, like the projector-subject distance or the surface inclination.

2) Algorithm validity

To test the validity of the algorithm, for several columns, the k_s value obtained by the algorithm is compared to the range of k_s values; which can give an exact restitution of the number of strips.

Over several images, a set of columns (10% of all) have been chosen to evaluate the validity of the proposed solution. This gives, for each image, a set of 71 columns covering all the areas of interest across the face.

The numbers of the projected colored lines on the face over these columns are counted using the human eye. Then for each of these columns we have applied a Gaussian filter with kernel size varying between 1 and 100, and compared

 ISBN: 978-1-61208-184-7

the resulted peak number (found using the peak locating algorithm presented later) with the real number of lines. At the end of this step the range of k_s that yields the exact number of peaks is found for each one of the columns.

Fig. 4 shows an example for the image used, where the x-axis represents the index for the columns, the y-axis represents the values of k_s, the dashed and solid curves represent successively the minimum and the maximum of k_s that yields the exact number. The dotted curve shows the value of k_s found using the algorithm, and its related position to the range that yields the good results.

After this step of the determination of k_s, it is necessary to apply the Gaussian filter to the images.

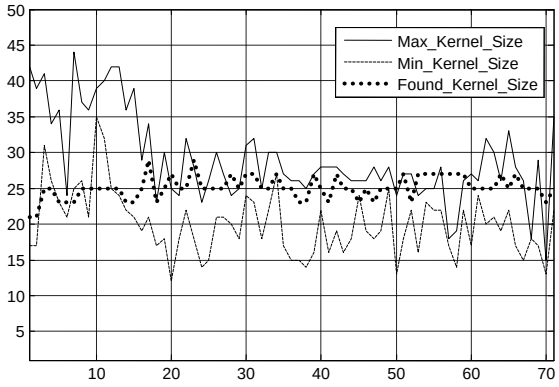


Figure 4. Ranges of values of k_s obtained on 71 columns

D. Locating peaks

1) Smoothing filter application

The Gaussian filter designed uses the previously described algorithm on each column of the image. As early mentioned this procedure smooths the signal and focuses the peaks of the strips in the middle. This procedure enables to recover the strip centers easily and with robustness. And it also makes the RGB component misalignment problem trivial. Fig. 5 shows several strips of a column signal, the solid curve represents the original signal while the dashed one represents the filtered signal.

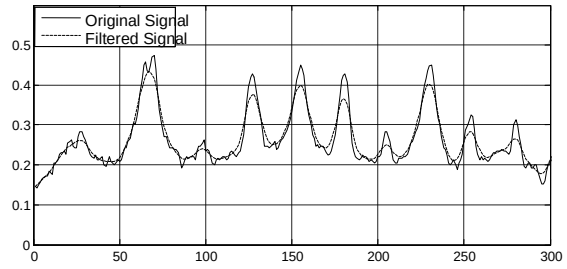


Figure 5. Original and filtered signal of a column signal.

On easy strips, the procedure does not improve the quality of the signal, in order to detect the local maxima, but on a bad quality signal (misalignment colors, etc.), this procedure allows to find a satisfactory solution.

2) Detection of the maximum

To determine the strip centers, it is necessary to find the local maxima along the smoothed signal of each column. The exact position of these local maxima is determined by applying the following simple algorithm to the smoothed signal; the difference between a point and the previous one must be positive, and the difference between a point and the following one must be negative, therefore the point is at the maximum of the curve.

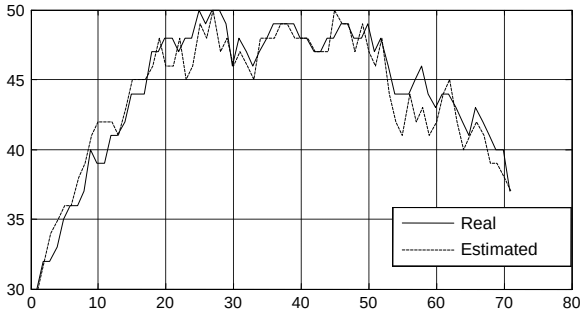


Figure 6. Peak detection error obtained on 71 columns.

In the special case where several points have the same value, the middle position (x-axis) is taken as maximum. This procedure is applied to all columns as previously. An evaluation of the detection quality is achieved by using the difference between the number of detected stripes and the real number of lines counted using the human eye. Fig. 6 shows the numbers of detected stripes, the solid and dashed curves are respectively the number of stripes detected by the eyes (real) and estimated by the algorithm. On several real images, the maximum of detection error is about 3 stripes. The mean of the error detection, number of real strips minus number of estimated strips, is about 1,5 strip/column.

IV. LINE CONSTRUCTION

A. From peaks to line

An image is constructed where each strip is represented by a simple line. Fig. 7 depicts the lines detected on a face; the one on the left represents the detected strips for the entire face superposed with the V component image. This image shows the exact position of the detected peaks in the middle of the strips. While the center and right parts of Fig. 7 successively represent a detail of the mouth and a detail of the nose. It is easy to understand that parts like the mouth will be easy to recognize and parts like the nose will be difficult to attribute.

 ISBN: 978-1-61208-184-7

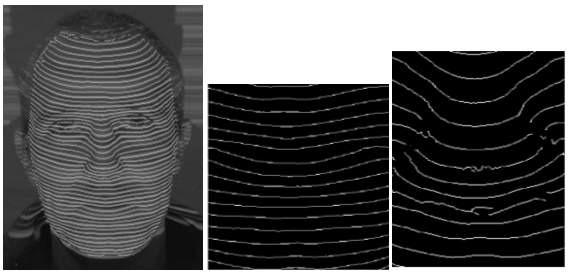


Figure 7. Line construction (face, mouth and nose).

B. Line connection

In the classic methods of structured light systems, the points resulting from the previous step are treated without connection. In order to increase the quality of the color attribution step and pattern affectation, a step of line connection is implemented to connect the isolated detected peaks within their local neighborhood. The horizontal projected lines in the captured image becoming curves have more or less accentuated slopes. As a result of searching the strip centers per column, a strip with a vertical slope generates discontinuity problems. Fig. 8 (top) shows the maximum allowed slope before getting this discontinuity. To overcome this problem the following connection algorithm is applied:

The points in the image are scanned from left to right and a criterion is tested in each one to decide whether it is well connected to the adjacent points in the line. If this point is found as an end of a connected line, a searching procedure is applied in a 7*10 pixel window to find the nearest beginning of another connected part of line. The two points are connected and the algorithm keeps scanning the other points in the image. An example of this problem and a resulting part of the image are shown in Fig. 8 (below).

The neighborhood used is determined empirically and found to be sufficient and yields good results in the face reconstruction application. At the end of this step, the majority of the detected peaks are connecting in long enough lines.

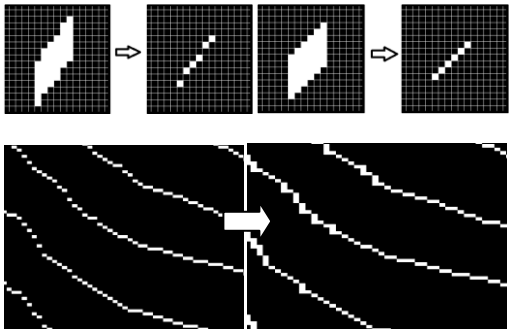


Figure 8. Top: maximum allowed slope, Down: connection algorithm.

C. Color of the line detection

1) Overview

It is well known that colors encounter a lot of distortions, starting with the influence of the projector, during its reflection on the surface of the measured object, and finally in the sensors of the camera. These distortions may be affected by either the absorption and reflecting proprieties of the measured object or by the proprieties of the projector and camera used which can lead to a cross-talk between the projector and the camera sensors. A lot of work has been done in this field; Zhang, Curless and Seitz [4] assumed that the surface is spectrally uniform and only calculated the projector-camera color cross-talk matrix, while in [3] an adaptive color classification is proposed to detect the colors of the projected lines using the RGB color space.

Several difficulties appear while using the RGB color space: the RGB components are highly correlated, the RGB space is not perceptually uniform, and it is highly sensible to color variations [6]. In our application, we chose to detect the color of the detected lines using the HSV color space and more precisely its H component which corresponds to the hue and refer to the dominant color. This color space is more related to the way in which human beings perceive colors and it is usually less affected by color variations.

In RGB color space, the colors of the pattern are chosen among the eight full saturated colors.

2) The H component histogram

Using the information of the H component and the connecting lines resulting from the previous step, the algorithm we used to assign the colors to these lines is described as the following:

The original captured image is converted to HSV color space and the H component is used. The ideal histogram of the six colors used in the H component is shown on Fig. 9a.

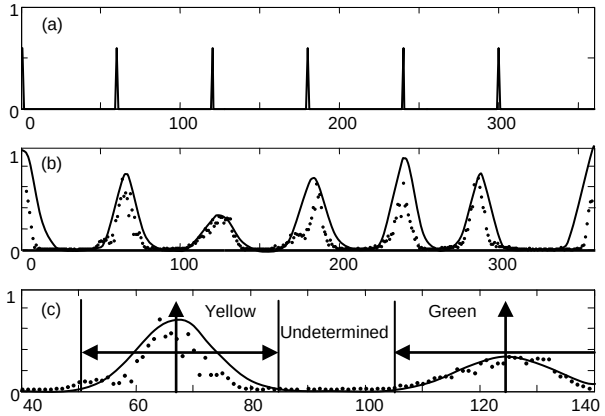


Figure 9. Histogram theoretical (a), experimental (b), enlargement (c).

The x-axis is graduated in degrees, so two successive colors are separated by 60°. The histogram of the detected lines in the H component image is shown in Fig. 9b. Due to

the distortion mentioned earlier, the distribution of each color is less regular than the ideal one. In most cases, the shape of the distribution is similar to a normal distribution around the mean value of each color area. The parameters of this distribution depend on the behavior of the colors on the skin, in other words, the interaction of light with skin. The modified parameters are the mean value, the shape and the width of the bumps. The bumps can be assimilated to Gaussian curves, so the width can be replaced by a standard deviation.

Due to these distortions, especially when these shapes have a small peak and large standard deviation, it is difficult to define a rule that can ensure a safe division of the H component into six regions and determine the boundaries between these regions.

The solution of this problem is divided into two steps: the first step is to find the pixels that can ensure a good color assignment, while the second one is to use the region growing in the connected lines to assign the colors of the remaining pixels.

3) Color attribution

The first step is achieved by dividing the distance between each two successive mean value positions in each color area into three equal regions.

Each one of the border regions has a safe color assignment to the related color; while the central region is labeled as undetermined (Fig. 9c) and set to the color white. In the second step, the indetermination will be lifted using the connected lines and the region growing.

The regions with the undetermined color are searched inside a connected line, and its color is found using the color of its adjacent pixels on both sides. These regions could be found in three possible cases:

- The regions can be within the connected line and surrounded by the colored regions from both sides and the surrounding regions have the same color on both sides, the undetermined color of the region is set as the same color.
- The regions can be within the connected line but the surrounding regions do not have the same color. This case is related to the connection of two lines with different colors. The regions with the undetermined color are deleted to avoid an error.
- Undetermined regions can be at the beginning or the end of the line, and then the color is determined by the color of the pixels on one side.

All the points in the connected lines resulting from this step have the same color. An example of the resulting image is shown in Fig. 10.

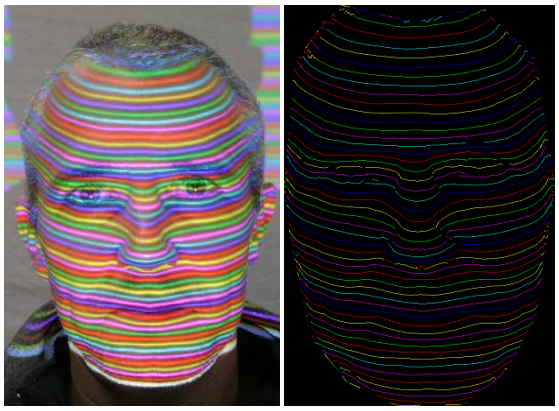


Figure 10. The resulting colored lines.

V. LINE MATCHING

As mentioned in the introduction of the paper, a De_Bruijn sequence is a cyclic sequence which consists of a certain number of subsequences. Each of these subsequences appears only once. This concept is used to match the detected strips to their corresponding ones in the projected pattern. The real scenes are not usually uniform ones. In many cases several parts of the projected lines will not be found in the captured image because of mislabeling, occlusions, shadows and other properties of each scanned object [4]. In this case, several colored lines will be missed in some columns. This means that the line matching using the classic dynamic programming is not always the best way. Multi-pass Dynamic Programming is proposed in [4]. In this method, the authors compute the monotonic components of the optimal path in multiple passes. To solve this problem, as mentioned earlier, each group of connected points is treated together as a connected line.

The information from all the points along the line is used to match this line to a line in the projected pattern. A subsequence of the six adjacent lines is formed and compared to De_Bruijn sequence used to create the projected pattern. In the case where some parts of the line have the problems mentioned earlier, such as occlusion, shadows and other problems will be matched using the information of the remaining points in the lines. For each of these lines the previous and following lines were found and used to assign it to a line in the projected pattern. The resulting points were triangulated with the corresponding points in the projected pattern by calculating the intersection of the two lines of sight defined by these points and the focal points of the camera and projector.

VI. 3D FACE MODEL

Several experiments have been performed. Fig. 11 shows typical example of these experiments; the first image with the pattern is used to generate the 3D wireframe model, while a second one without the pattern is used to add the texture.

 ISBN: 978-1-61208-184-7

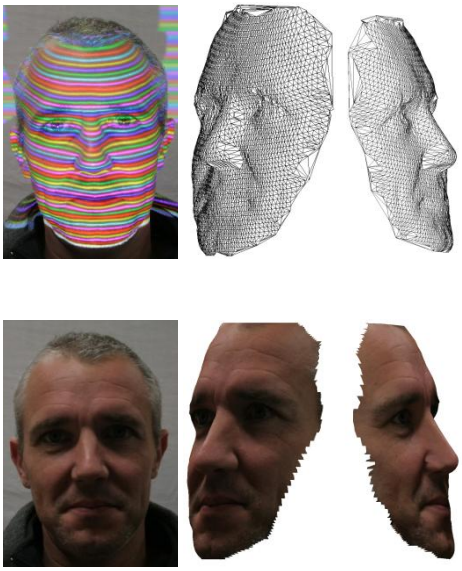


Figure 11. Images used and resulting wireframe model and textured one.

The step of peak detection yields some points which correspond to false peaks between the real peaks. Other eventual errors arise where the false peaks are found in the same local neighborhood. They can be connected during the connection step. In the step of color detection, a specific color will be assigned to these resulting false lines (the color red in most cases).

In the step of lines matching, the subsequence will be formed by this false line and the other five surrounded colored lines. As a result of use of the De_Bruijn sequence with big period (subsequence of six elements), there is no chance that the resulting subsequence will be matched to a subsequence in the De_Bruijn sequence, and these false lines or points will be excluded.

VII. CONCLUSION

We have presented in this paper our 3D scanner using the structured light system technique. Our robust method is applied on a human face; it can be also used in many other fields, either in computer vision or industry applications. A specially designed color pattern using the De_Bruijn sequence is proposed to enhance the robustness of the system.

A new method to locate the strip centers in the captured image uses a smoothing filter with a large kernel applied to the V component of the HSV color space. The validation step of the algorithm used to choose the size of the kernel has been presented, and satisfactory results have been obtained. Usually, each strip center is treated alone, and more precisely, each point of the picture is treated alone. We use an innovative and global technique in order to construct usable lines. The method is based on a growing algorithm using an appropriate window and it is applied to V component image. We treat each set of connected points together while determining their color and assigning them to a line in the projected pattern. The last paragraph presents the results obtained on 3D face modeling. A texture has been applied to be closer to reality.

Due to the robustness of the proposed 3D face acquisition method it has been used to scan faces for web applications.

The week point of this work is the absence of comparative studies with the existing approaches. Such studies form the major part of our future work.

REFERENCES

[1] P. Fechteler, P. Eisert, and J. Rurainsky, “Fast and High Resolution 3D Face Scanning” In Proc. of the International Conference on Image Processing (ICIP 2007), vol. 3, pp. 81-84, San Antonio, Texas, USA, doi:10.1109/ICIP.2007.4379251.

[2] J. Pages, J. Salvi, C. Collewet, and J. Forest, “Optimised De Bruijn patterns for one-shot shape acquisition” Image and Vision Computing 2005, vol. 23, Issue 8, pp. 827-849, doi:10.1016/j.imavis.2005.05.007.

[3] P. Fechteler, P. Eisert, “Adaptive Color Classification for Structured Light Systems”, in Computer Vision and Pattern Recognition Workshops (CVPRW 2008), pp. 1-7, doi:10.1109/CVPRW.2008.4563048.

[4] L. Zhang, B. Curless, and S.M. Seitz, “Rapid shape acquisition using color structured light and multipass dynamic programming”, 1st IEEE International Symposium on 3D Data Processing 2002, Visualization, and Transmission, pp. 24–36, Padova, Italy, doi:10.1109/TDPVT.2002.1024035.

[5] J. Salvi, J. Pagès, and J. Batlle, “Pattern codification strategies in structured light systems”, Pattern Recognition 2004, vol. 37, Issue 4, pp. 827-849, doi:10.1016/j.patcog.2003.10.002.

[6] Jiebo Luo, D. Crandall, “Color object detection using spatial–color joint probability functions”, IEEE Transactions on Image Processing 2006, vol. 15, Issue 6, pp. 1443-1453, doi:10.1109/TIP.2006.871081.

[7] D. Scharstein, R. Szeliski, R. Zabih, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, In Proc. of IEEE Workshop on Stereo and Multi-Baseline Vision, SMBV 2001, pp. 131-140, doi:10.1023/A:1014573219977.

[8] A. Dipanda, S. Woo, “Towards a real-time 3D shape reconstruction using a structured light system”, Pattern Recognition, Oct. 2005, vol. 38, Issue 10, pp. 1632-1650, doi:10.1016/j.patcog.2005.01.006.

[9] Y. F. Li and B. Zhang, “A Method for 3D measurement and reconstruction for active vision”, Measurement Science and Technology, 2004, vol. 15, Issue 11, pp. 2224-2232, doi:10.1088/0957-0233/15/11/007.

[10] Z. Zhang, “A flexible new technique for camera calibration”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, vol. 22, Issue 11, pp. 1330-1334, doi:10.1234/12345678.

 ISBN: 978-1-61208-184-7

Finding 3D Positions from 2D Images Feasibility Analysis

H. G. Lochana Prematunga
University of Colombo School of Computing,
35, Reid Avenue, Colombo 7, Sri Lanka
lochana.prematunga@ifsworld.com

Anuja T Dharmaratne
University of Colombo School of Computing,
35, Reid Avenue, Colombo 7, Sri Lanka
atd@ucsc.lk

***Abstract*—- In this paper, we prove that it is possible to recover the position (or coordinates) of an object using a single 2D image, given the size and shape of the image. Here we employ a purely mathematical proof to enable guaranteed accuracy. These theories and their derivations can be employed in recovering illumination patterns defined on the actual object using their images.**

Keywords-3D position, 2D image, uniqueness, mathematical modeling.

I. INTRODUCTION

There are situations in computer vision where it is required to determine the position of an object using a single 2D image. In this paper, these positions are expressed as coordinates in a system of coordinates defined by the camera. When an image of a certain object is given, there is a doubt whether there can be multiple object positions that may create this same image. But this paper takes a mathematical approach to prove that, when the shape and the size of an object are given with its image from a certain camera, the coordinates of the object can be determined uniquely. Therefore it is proved that there can be one and only one position for an object with respect to a camera, for a given image.

Also the coordinates of the object with respect to the camera is disclosed in the process as mathematical formulas. Here a rectangular object is taken as the example to prove the uniqueness. But any planer object satisfies the uniqueness condition, since a rectangle drawn on such an object has a unique image inside a given image of that original object.

These discoveries will be of importance in distance detection applications. As an example, an image of a satellite may be used to determine the distance between the satellite and the camera from which the image is taken. Similar approach may be employed to determine the position of a person in an airplane using a photograph of a wing (of the same flight) taken by him in an air plane crash situation.

Distance information may be used as data itself as explained above or it may be an intermediate data used to derive some other important information. For an example, brightness of an image pixel will not yield any brightness information (of the object) if the distance between the camera and the object is unknown.

Also the angle between the object and the reviewer may provide important information in some legal procedures. Another application of the angle matching will be to position the antenna in the most favorable direction for the receiver. This will play an important role in satellite communication and in communicating with space vehicles.

Positioning of 3D objects (surgical equipments) and recovering the position and orientation data of objects (organs) accurately is also a vital step in computer assisted surgeries.

Recovering 3D models from 2D images is also an important step in virtual reality applications.

Finally let us look into the structure of the paper. Next section is devoted to identify the previous work, which is related to the work in this article, done by other researchers. The section after that introduces the camera model used in the rest of the article. Section that follows is devoted to disclose the methodology used to recover the object from the image. And the last section in the paper is allocated to list the conclusions.

II. PREVIOUS WORK

There are attempts to recover 3D images using a series (2 or more) of 2D images of certain objects. One is the research done at the University of Ottawa [2]. This will use 2 images taken from 2 cameras simultaneously and the object is being recovered using the images of some feature points on the 3D model.

In another research there was a successful attempt of recovering a non-rigid 3D model [3] from a sequence of images created by a video stream. Also there are some tools developed to recover the 3D models from images with wide baselines [4]. These tools employ a method that uses a universal camera intrinsic matrix estimation technique to eliminate the need for camera calibration experiments.

In another research [5], there is an attempt to recover smooth objects using image contours that approximate the image with an octree spline structure. Some research work has also been carried out on recovering moving 3D objects [6]. This method consists of integrating the measured 2D motion of the object to recover its 2D-position in the image.

 ISBN: 978-1-61208-184-7

There is an interesting research [7] on recovering 3D object models from a single 2D image. In this method, the matching of corresponding features is employed to recover 3D data. Some research was done to recover the pose of a head [8] including the motion to be mainly used in virtual reality aviators.

An interesting approach to the problem has been presented in [9]. In this paper certain pre-analyzed object classes have been used. Objects belonging to these classes were then extracted from an image. Some work was also done using voting techniques [10] but without feature extraction. This will enable the method to be employed also for smooth objects. Another approach used was to synthesize 3D objects by comparing the image against the data in a pre-stored object library. In one of the studies described in [11] this approach was used in 3D model synthesis to even recover the data in the back (invisible) side of the objects. Some research [12] is also done on the difficult problem of decoupling the relative position recovery and relative orientation recovery.

III. CAMERA MODELING

The camera model used to establish the mathematical formulas is shown in the figure bellow:

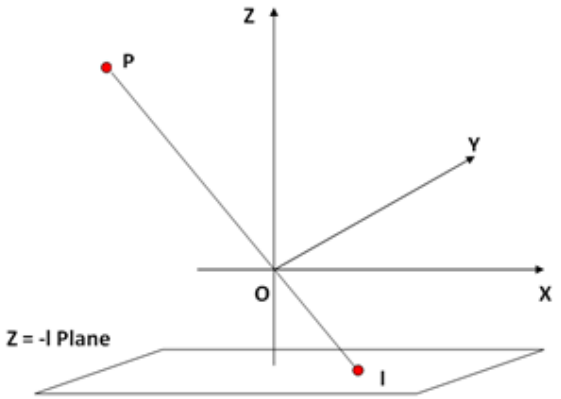


Figure 1: Camera model used in the derivation of mathematical formulas.

Let **O** $(0, 0, 0)^T$ be the center of the camera and the plane **Z = -l** be the imaging surface.

Let **P** $(x, y, z)^T$ be any point that is capable of generating an image on the camera and **I** $(x_I, y_I, -l)^T$ be its image.

Then for a given scalar t observe the relationship:

$$\overrightarrow{OP} = t\,\overrightarrow{IO}$$
$$\boldsymbol{P} = -t\,\boldsymbol{I}$$

That is

$$(x, y, z)^T = -t(x_I, y_I, -l)^T.$$

That is for a given image **I** $(x_I, y_I, -l)^T$ for a point **P**, **P** can be given by $(-tx_I, -ty_I, tl)^T$.

$$P = (-tx_I, -ty_I, tl)^T \quad \text{------ (1)}$$

In this case the parameter t should be positive (t>0) in the real world.

Otherwise the point P will be inside the camera or behind it.

IV. RECOVERING THE OBJECT FROM THE IMAGE

Let $P_0P_1P_3P_2$ be a rectangular object with P_0P_2 = m, in the above 3D coordinate system.

And let $I_0I_1I_3I_2$ be its image on the imaging surface z = -l of the camera. Refer to Figure 2.

Let $I_0 = (x_0, y_0, -l)^T$, $I_1 = (x_1, y_1, -l)^T$, $I_2 = (x_2, y_2, -l)^T$ and $I_3 = (x_3, y_3, -l)^T$.

Then, equation (1) implies:

$$P_0 = (-t_0x_0, -t_0y_0, t_0l)^T,$$
$$P_1 = (-t_1x_1, -t_1y_1, t_1l)^T,$$
$$P_2 = (-t_2x_2, -t_2y_2, t_2l)^T,$$
$$P_3 = (-t_3x_3, -t_3y_3, t_3l)^T.$$

Where t0, t1, t2 and t3 are positive scalar (parametric) values.

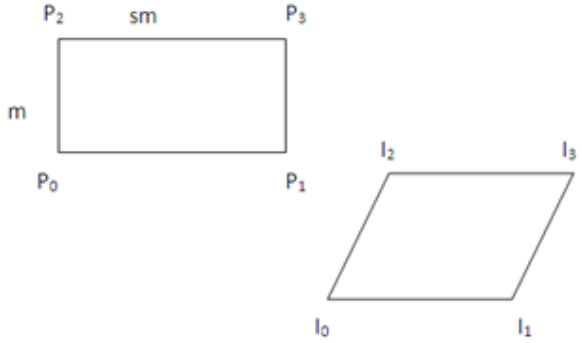


Figure 2: Actual object $P_0P_1P_3P_2$ and its image $I_0I_1I_3I_2$.

Since $P_0P_1P_3P_2$ is a rectangle: $\overrightarrow{P_0P_1} = \overrightarrow{P_2P_3}$.

$$\boldsymbol{P_1 - P_0 = P_3 - P_2}$$

$$(-\boldsymbol{t_1x_1}, -\boldsymbol{t_1y_1}, \boldsymbol{t_1l})^T - (-\boldsymbol{t_0x_0}, -\boldsymbol{t_0y_0}, \boldsymbol{t_0l})^T = (-\boldsymbol{t_3x_3}, -\boldsymbol{t_3y_3}, \boldsymbol{t_3l})^T - (-\boldsymbol{t_2x_2}, -\boldsymbol{t_2y_2}, \boldsymbol{t_2l})^T$$

Comparing z components: $t_1l - t_0l = t_3l - t_2l$

$$t_1 - t_0 = t_3 - t_2.$$

Thus,

$$t_1 + t_2 - t_3 = t_0 \quad \text{-- (2)}$$

Comparing x components:

$$-t_1x_1 - (-t_0x_0) = -t_3x_3 - (-t_2x_2).$$

Therefore,

$$t_1x_1 + t_2x_2 - t_3x_3 = t_0x_0 \quad \text{-- (3)}$$

Similarly, by comparing y components:

$$t_1y_1 + t_2y_2 - t_3y_3 = t_0y_0 \quad \text{--(4)}$$

By solving these equations:

$$t_3 = t_0 \frac{(x_{10}y_{21} - x_{21}y_{10})}{(x_{31}y_{21} - x_{21}y_{31})} \quad \text{--- (5)}$$

where $x_{ij} = x_i - x_j$.

And

$$t_2 = t_0 \frac{(x_{10}y_{31} - x_{31}y_{10})}{(x_{31}y_{21} - x_{21}y_{31})} \quad \text{--- (6)}$$

That is:

$$t_2 = at_0 \quad \text{---- (7)}$$

And

$$t_3 = bt_0 \quad \text{--- (8)}$$

Where

$$a = \frac{(x_{10}y_{31} - x_{31}y_{10})}{(x_{31}y_{21} - x_{21}y_{31})}$$

and

$$b = \frac{(x_{10}y_{21} - x_{21}y_{10})}{(x_{31}y_{21} - x_{21}y_{31})}.$$

Considering the fact that $P_0P_2 = m$.

Therefore $|P_2 - P_0| = m$

$$|(-t_2x_2, -t_2y_2, t_2l)^T - (-t_0x_0, -t_0y_0, t_0l)^T| = m$$

$$(t_2x_2 - t_0x_0)^2 + (t_2y_2 - t_0y_0)^2 + l^2(t_2 - t_0)^2 = m^2.$$

Using the equation (7):

$$(at_0x_2 - t_0x_0)^2 + (at_0y_2 - t_0y_0)^2 + l^2(at_0 - t_0)^2 = m^2$$

$$t_0^2\{(ax_2 - x_0)^2 + (ay_2 - y_0)^2 + l^2(a-1)^2\} = m^2 \quad \text{---(9)}$$

Therefore we have

$$t_0 = \frac{\pm m}{\sqrt{\{(ax_2 - x_0)^2 + (ay_2 - y_0)^2 + l^2(a-1)^2\}}}$$

But by definition of t_0, we have $t_0 > 0$.

Therefore:

$$t_0 = \frac{m}{\sqrt{\{(ax_2 - x_0)^2 + (ay_2 - y_0)^2 + l^2(a-1)^2\}}} \quad \text{---- (10)}$$

That means t_0 is a unique value for a given m.

Therefore by considering equations (5) and (6) we can uniquely determine t_2 and t_3.

- For a given image $I_0I_1I_3I_2$, we can uniquely determine the points P_0, P_3 and P_2.
- By geometry we can uniquely determine the point P_1.
- For a given image $I_0I_1I_3I_2$ we can uniquely determine the rectangle $P_0P_1P_3P_2$ that created the image.

Now consider the image shown in Figure 3.

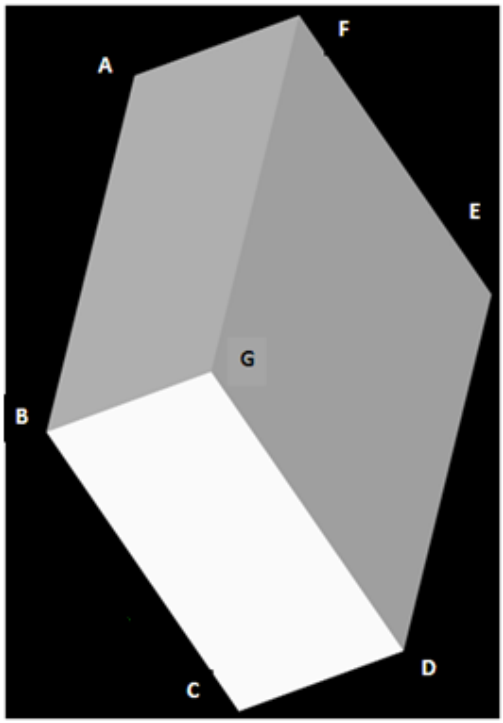


Figure 3: Image of an actual object to be measured.

Point	Pixel X	Pixel Y	Actual X	Actual Y
A	103	58	2.682292	1.510417
B	33	340	0.859375	8.854167
C	186	562	4.84375	14.63542
D	316	515	8.229167	13.41146
E	387	231	10.07813	6.015625
F	233	10	6.067708	0.260417
G	163	293	4.244792	7.630208

 ISBN: 978-1-61208-184-7

Here the ratio 38.4 was taken to convert pixel values to centimeters. Considering the geometry, it is safe to take l = 1, without losing the generality. Considering the rectangle BCDG, and with BG = 1cm, we got $t_0 = 0.0865161$.

That means the actual coordinates of B is given by (-0.07435, -0.76603, 0.0865161) where all the coordinates are given in cm. Similarly other coordinates can also be calculated.

V. CONCLUSION

Given the shape and size of an object with an image of it, there can be one and only one position for the object with respect to the camera from which the image is taken. This implies that for the given class of objects, (planar objects with an identifiable rectangular shape on them) when an image is given, and the 3D positions are calculated, there is no need to find out whether there are any more possible object instances that we need to consider. This will greatly reduce the complexity of successive steps in a system where object extraction is an intermediate step. Otherwise it would be required to apply the same algorithm to multiple possible objects and validate each other to select the best appropriate.

This simple observation not only reduces the time and complexity of the resulting systems, but also reduces the possibility of errors.

References

[1] Eric Marchand. Control Camera and Light Source Positions using Image Gradient Information. IEEE Int. Conf. on Robotics and Automation, ICRA'07, Roma, Italia, April 2007.

[2] Lavoie P., Lonescu D., Petriu E.M.. 3D object model recovery from 2D images using structured light. Instrumentation and Measurement, IEEE Transactions.

[3] Christoph Bregler, Aaron Hertzmann, Henning Biermann. Recovering Non-Rigid 3D Shape from Image Streams.

[4] Yuzhu Lu, Shana Smith. A Comprehensive Tool for Recovering 3D Models From 2D Photos With Wide Baselines. J. Comput. Inf. Sci. Eng. December 2006.

[5] Lavallee S, Szeliski R. Recovering the position and orientation of free-form objects from image contours using 3D distance maps. IEEE Pattern Analysis and Machine Intelligence, IEEE Transactions on Apr 1995.

[6] Cretual A, Chaumette F, Bouthemy P. Complex object tracking by visual servoing based on 2D image motion. IEEE Pattern Recognition, 1998, Proceedings, Fourteenth International Conference on 16-20 Aug 1998.

[7] D Danial Sheu, Alan H Bond. A Generalized Method for 3D Object Location from Single 2D Images. Pergamon Press Ltd, 1992 Pattern Recognition Society.

[8] M D Cordea, E M Petriu, N D Georganas, D C Petriu, T E Whalen. 3D Head Pose Recovery for Interactive Virtual Reality Avatars. IEEE Instrumentation and Measurement Technology Conference, Budapest, Hungary, May 21-23, 2001.

[9] Han-Pang Chiu, Huan Liu, Leslie Pack Kaelbling, Tom´as Lozano-P´erez. Class-Specific Grasping of 3D Objects from a Single 2D Image. The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems October 18-22, 2010, Taipei, Taiwan.

[10] Kenpo Tsuchiya, Shuji Hashimoto, Toshiaki Matsushima. A Pixel Voting Method to Recover 3D Object Shape from 2D Images. MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-1 5, 1994, Kawasaki.

[11] Tal Hassner, Ronen Basri. Example Based 3D Reconstruction from Single 2D Images. Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06) IEEE.

[12] Rodrigo L Carceroni, Christopher M Brown. Decoupling Orientation Recovery from Position Recovery with 3D-2D Point Correspondences. University of Rochester, Computer Science Department, Rochester, NY-14627-USA.

 ISBN: 978-1-61208-184-7

Interpolation and Intersection Algorithms and GPU

Vaclav Skala
Department of Computer Science
VSB-Technical University
Ostrava, Czech Republic
Vaclav.Skala@vsb.cz

Abstract — Interpolation and intersection methods are closely related and used in computer graphics, visualization, computer vision etc. The Euclidean representation is used nearly exclusively not only in computational methods, but also in education despite it might lead to instability in computation in many cases. The projective geometry, resp. projective extension of the Euclidean space, offers many positive features from the computational and educational points of view with higher robustness and stability of computation. This paper presents simple examples of projective representation advantages, especially from the educational point of view. In particular, how interpolation and intersection can be applied to fundamental algorithms, which are becoming more robust, stable and faster due to compact formulation. Another advantage of the proposed approach is a simple implementation on vector-vector architectures, e.g. GPU, as it is based on matrix-vector operations.

Keywords - Interpolation; intersection; principle of duality; barycentric coordinates; cross-product; linear systems of equations.

I. INTRODUCTION

Algorithm efficiency and robustness are key points of research activities in computer graphics [7], [1], computer vision [6], [4], texture mapping [19] etc. Due to many items to be processed, a strong requirement for speed arises; also hardware architecture needs to be considered. However, speed and robustness requirements are usually in contradiction, especially if the Euclidean representation is used.

Nevertheless, some other approaches like projective or conformal geometries can be used to overcome selected problems. As the projective representation is widely used in computer graphics, a simple modification of interpolation and intersection algorithms will be introduced and simple examples presented for demonstration.

II. PROJECTIVE REPRESENTATION

Projective representation uses homogeneous coordinates for computations and geometric transformations. A point $\boldsymbol{X} = (X, Y)$ in E^2 (the Euclidean space) can be represented as $\boldsymbol{x} = [x, y: w]^T$ in P^2 (the projective extension of the Euclidean space). Mutual conversion is defined as:

$$X = {}^{x}/_{w} \quad Y = {}^{y}/_{w} \quad w \neq 0 \quad \text{(origin excluded)}$$

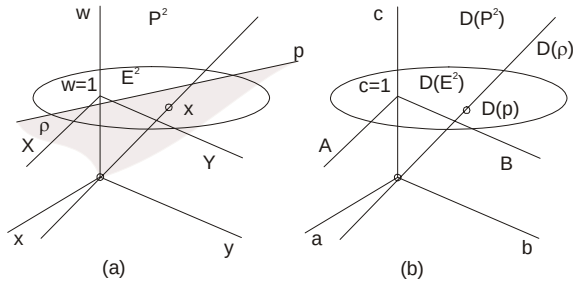


Figure 1. Geometric interpretation of a dual space

One parametric set of points in P^2 representing a unique point in E^2, see [14]. A significant advantage of the projective representation is a possibility to use principle of duality. In the above case, a point is dual to a line and vice versa, etc. which might lead to new algorithms [2], [13], [18].

Similarly, the concept of the projective extension of the Euclidean space can be extended to n-dimensional space, especially to E^3 used in computer graphics and vision.

This simple formulation shows, that many computations, not necessarily only geometric transformations, can be made using homogeneous coordinates.

It means that "*projective scalar*", i.e. $\boldsymbol{x} = [\bar{x}: w]^T$ or "*projective vector*" $\boldsymbol{x} = [\bar{\boldsymbol{x}}: w]^T = [x, y: w]^T$, where $\bar{\boldsymbol{x}}$ is a vector, can form an input or output of the processing pipeline, e.g. interpolation and intersection computation.

Projective representation can also help to explain many geometrical problems in a simple way, e.g. line intersection [13], area or volume computation [14], solution of linear homogeneous systems and computation of barycentric coordinates [12].

III. PRINCIPLE OF DUALITY

Principle of duality is an essential principle and especially in computer graphics and vision can bring quite a new way how to handle and solve non-trivial problems. The principle states that any theorem remains true when we interchange the words "point" and "line", "lie on" and "pass through", "join" and "intersection" and so on. Once the theorem has been established, the dual theorem is obtained as described above, see [5].

 ISBN: 978-1-61208-184-7

The advantages of the projective geometry and principle of duality use can be demonstrated on very simple examples, e.g. a point as an intersection of two lines and a line as a join of two points.

Let two points $\mathbf{x}_1$ and $\mathbf{x}_2$ be given in the projective space. Then the coefficients of the line $\mathbf{p}$, which is defined by those two points, are determined as the of their homogeneous coordinates.

$$\mathbf{p} = \mathbf{x}_1 \times \mathbf{x}_2 \qquad \text{i.e.} \qquad \mathbf{p} = \det\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{bmatrix}$$

where: $\mathbf{p} = [a,b:c]^T$

If the principle of duality is used, it is possible to write computation of an intersection of two lines as:

$$\mathbf{x} = \mathbf{p}_1 \times \mathbf{p}_2 \qquad \text{i.e.} \qquad \mathbf{x} = \mathbf{p}_1 \times \mathbf{p}_2 = \det\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix}$$

where: $\mathbf{x} = [x,y:w]^T$

A computation of an intersection point of two lines or a computation of a line if two points are given is made by the same sequence using the principle of duality and no division operation is needed.

In the case of E^3 point is dual to a plane and vice versa, i.e. if three points are given a computation of a plane is the same, in the sense of duality, as intersection computation of three planes, i.e. a plane is computed by the generalized cross-product as:

$$\mathbf{x}_1 \times \mathbf{x}_2 \times \mathbf{x}_3 = \det\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{bmatrix}$$

and an intersection of three planes is computed as:

$$\boldsymbol{\rho}_1 \times \boldsymbol{\rho}_2 \times \boldsymbol{\rho}_3 = \det\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix}$$

It should noted that a division operation is not needed and the computational sequence is the same for both the cases.

IV. LINEAR INTERPOLATION

Linear interpolation is frequently used method not only in computer graphics. Let us consider a simple case of linear interpolation, when we want to interpolate on a line $\boldsymbol{p}$ or on a surface $\boldsymbol{\rho}$, i.e.

$$\boldsymbol{p}: \boldsymbol{X}(t) = \boldsymbol{X}_A + \boldsymbol{S}_1 t \quad \text{or}$$

$$\boldsymbol{\rho}: \boldsymbol{X}(u,v) = \boldsymbol{X}_A + \boldsymbol{S}_1 u + \boldsymbol{S}_2 v$$

where: $\boldsymbol{S}_1 = \boldsymbol{X}_B - \boldsymbol{X}_A$ and $\boldsymbol{S}_2 = \boldsymbol{X}_C - \boldsymbol{X}_A$

These are well-known formulas, of course. But what happens if points are given in homogeneous coordinates?

From the teaching experience, the approach is a conversion of points to the Euclidean space followed by the "standard" linear interpolation. It means that in the first case 6 divisions and in the second case 9 divisions are needed with all consequences, including precision and stability issues.

However, there is a possibility to make a linear interpolation **directly** in homogeneous coordinates as:

$$\boldsymbol{p}: \boldsymbol{x}(t) = \boldsymbol{x}_A + \boldsymbol{s}_1 t$$

where: $\boldsymbol{s_1} = \boldsymbol{x}_B - \boldsymbol{x}_A$
$= [x_B - x_A,\ y_B - y_A : w_B - w_A]^T$

or

$$\boldsymbol{\rho}: \boldsymbol{x}(\xi,\eta) = \boldsymbol{x}_A + \boldsymbol{s}_1\xi + \boldsymbol{s}_2\eta$$

where: $\boldsymbol{s_1} = \boldsymbol{x}_B - \boldsymbol{x}_A$
$= [x_B - x_A,\ y_B - y_A, z_B - z_A : w_B - w_A]^T$

and

$\boldsymbol{s_2} = \boldsymbol{x}_C - \boldsymbol{x}_A = [x_C - x_A,\ y_C - y_A,\ z_C - z_A : w_C - w_A]^T$

In both cases, the following conditions apply:

$$w_A > 0, \qquad w_B > 0, \qquad w_C > 0$$

As in the projective space the metric is not generally defined, there must be some different behavior of such interpolation. Note that there is a direct connection to interpolation and projection operation in the graphical pipeline.

Basic property of the interpolation in the projective space is a **non-linear monotonic parameterization**, i.e. for $\tau = 1/2$ the center of the segment $\boldsymbol{X}_A\boldsymbol{X}_B$ in the Euclidean space is not obtained in general. It is well known problem of determining z-coordinate after projection operation. It means that we have a linear interpolation with:

- Linear parameterization in the Euclidean space
- Non-linear parameterization, but with a **monotonic** parameterization, in the projective space. This fundamental property is needed when comparison of $t_1 < t_2$, resp. $\tau_1 < \tau_2$, is required for a decision, e.g. which object is closer etc.

In both cases, division operation can be avoided by "hiding" denominator to the homogeneous coordinate, i.e.

$$\boldsymbol{x}(t) = [w_B\overline{\boldsymbol{x}}_A + (w_A\overline{\boldsymbol{x}}_B - w_B\overline{\boldsymbol{x}}_A)t : w_A w_B]^T$$

In this case, the parameterization is linear, of course.

It should be noted that barycentric coordinates can be computed directly in homogeneous coordinates without division operations as well, see [12], using generalized cross-product.

The above presented approach is quite simple for understanding projective space principles.

V. BARYCENTRIC COORDINATES

Barycentric coordinates are very often used not only in computer graphics, computer graphics and visualization. It is known that computation of barycentric coordinates leads to solution of linear system of equations (LSE). A solution of LSE is equivalent to the generalized cross-product. This

 ISBN: 978-1-61208-184-7

result to computation of the barycentric coordinates directly using generalized cross-product without use of division operation and therefore the computation is more robust in general. The barycentric coordinates are computed as

$$\mathbf{b} = \boldsymbol{\xi} \times \boldsymbol{\eta} \times \mathbf{w} \qquad \boldsymbol{\tau}^T \mathbf{b} = 0$$

$$\mathbf{b} = [b_1, b_2, b_3, b_4]^T \quad \boldsymbol{\xi} = [x_1, x_2, x_3, x]^T \quad \mathbf{w} = [1, 1, 1, 1]^T$$

$$\boldsymbol{\eta} = [y_1, y_2, y_3, y]^T \quad det \begin{vmatrix} \tau_1 & \tau_2 & \tau_3 & \tau \\ x_1 & x_2 & x_3 & x \\ y_1 & y_2 & y_3 & y \\ 1 & 1 & 1 & 1 \end{vmatrix} = 0$$

The barycentric coordinates of the point **x** are then given as

$$a_1 = -\frac{b_1}{b_4},\ a_2 = -\frac{b_2}{b_4},\ a_3 = -\frac{b_3}{b_4}$$

The above formulas shows that the computation of the barycentric coordinates is quite simple. If hardware acceleration using matrix-vector operation is used, the computation is very fast. It is important to note that the similar scheme for the barycentric coordinates computation is valid for homogeneous coordinates as the determinant is multi-linear.

VI. INTERSECTION COMPUTATIONS

A. Line-plane intersection

There is lot of algorithms based on line intersections, like ray-tracing, line clipping etc. Let us consider a simple case of intersection of a line in a parametric form with a plane, which is the fundamental principle of many algorithms, e.g. Cyrus-Beck's (CB) line clipping, see Fig. 2 for E^2 analogy (planes are "degenerated" to edges).

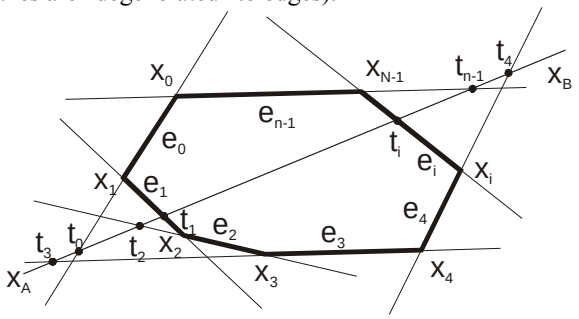


Figure 2. Line clipping by a convex polygon

Let us consider two planes $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ and a line $\boldsymbol{p}$ in a parametric form given in homogeneous coordinates by two points $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$.

In many algorithms including CB algorithm the relation $t_1 < t_2$, needs to be evaluated, e.g. to get an order of intersection points. For this *only* ***monotonic parameterization*** on the line $\boldsymbol{p}$ is needed. It means that the linear interpolation with non-linear parameterization presented above can be used efficiently.

The CB algorithm is based on an intersection solution of a line $\boldsymbol{p}$ given in a parametric form and a plane $\boldsymbol{\rho}$ in E^3 (or a line in E^2) given in the implicit form as follows:

$$\boldsymbol{\rho}: \boldsymbol{a}^T \boldsymbol{x} = 0 \qquad \boldsymbol{p}: \boldsymbol{x}(\tau) = \boldsymbol{x}_A + \boldsymbol{s}\tau$$

It should be noted that all vectors are vectors of the projective space, i.e. they have homogeneous coordinates. Therefore, it is easy to compute the intersection point as

$$\boldsymbol{a}^T \boldsymbol{x}_A + \boldsymbol{a}^T \boldsymbol{s}\tau = 0 \qquad \text{then} \qquad \tau = -\boldsymbol{a}^T \boldsymbol{x}_A / \boldsymbol{a}^T \boldsymbol{s}$$

The parameter can be represented by a "*projective scalar*" as

$$\tau = [-\boldsymbol{a}^T \boldsymbol{x}_A : \boldsymbol{a}^T \boldsymbol{s}]^T = [\bar{\tau} : \tau_w]^T$$

Then the CB's algorithm can be modified as follows:

$\tau_{min} = [-\infty : 1]^T; \qquad \tau_{max} = [\infty : 1]^T$

for i:=1 **to** N_planes **do**

{ $\boldsymbol{\tau} := [-\boldsymbol{a}^T \boldsymbol{x}_A : \boldsymbol{a}^T \boldsymbol{s}]^T$; # $\boldsymbol{\tau} = [\bar{\tau} : \tau_w]^T$ #

if $\tau_w < 0$ **then** $\boldsymbol{\tau} := -\boldsymbol{\tau}$;

τ_w coordinate needs to be non-negative

if $\bar{\tau} < 0$ **then** $\boldsymbol{\tau}_{min} = \max(\tau_{min}, \tau)$

else $\boldsymbol{\tau}_{max} = \min(\tau_{max}, \tau)$

}

if NON-EMPTY ($\boldsymbol{\tau}_{min}$, $\boldsymbol{\tau}_{max}$) = **true then**

#equivalent test to $t_1 < t_2$ #

{ $\boldsymbol{x}_{A\_new} = \cdots\ldots; \boldsymbol{x}_{B\_new} = \cdots$ }

Algorithm 1

The above shows that no division operation is needed. Experiments made proved a slight speed-up for the case when the points are given in the Euclidean space (the algorithm has been simplified as w_A, $w_B = 1$ of course) and significant speed-up for the case when the points of the clipped line are given in the homogeneous coordinates. As N_planes, the number of planes, is usually higher, the speed-up will grow with the number of planes of the given convex polyhedron.

B. Intersection of two planes

Intersection of two planes is another case very often solved in computer graphics and vision. Unfortunately in many cases available solutions are not robust or formula are neither simple nor convenient for GPU use.

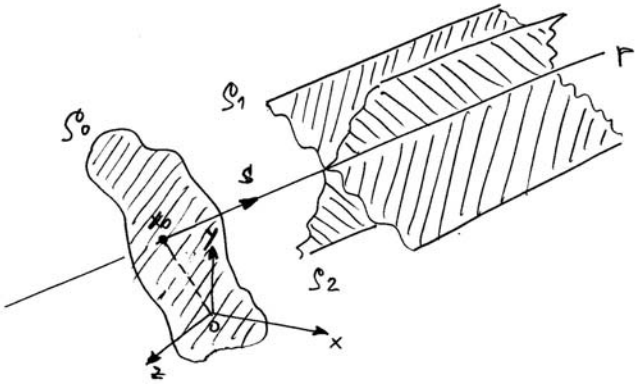


Figure 3. Intersection of two planes

 ISBN: 978-1-61208-184-7

If the projective space is used, the solution is quite simple. Let us consider two planes $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_1$ given as

$$\boldsymbol{\rho}_1 = [a_1, b_1, c_1 : d_1]^T \qquad \boldsymbol{\rho}_2 = [a_2, b_2, c_2 : d_2]^T$$

It means that normal vectors of those planes are

$$\boldsymbol{n}_1 = [a_1, b_1, c_1]^T \; \boldsymbol{n}_2 = [a_2, b_2, c_2]^T$$

It is obvious that a directional vector of a line is determined as an intersection of two planes $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_1$ given as

$$\boldsymbol{s} = \boldsymbol{n}_1 \times \boldsymbol{n}_2$$

However, the "starting" point $\boldsymbol{x}_0$ of the line is determined in quite complicated ways, sometimes even not robustly enough and based on a user choice of some value, or proposes solution of a system of linear equations [Gol90], [20].

The following "standard" formula can typically be found:

$$\boldsymbol{n}_3 = \boldsymbol{n}_1 \times \boldsymbol{n}_2$$

$$x_0 = \frac{d_2 \begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} - d_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix}}{DET}$$

$$y_0 = \frac{d_2 \begin{vmatrix} a_3 & c_3 \\ a_1 & c_1 \end{vmatrix} - d_1 \begin{vmatrix} a_3 & c_3 \\ a_2 & c_2 \end{vmatrix}}{DET}$$

$$z_0 = \frac{d_2 \begin{vmatrix} a_1 & b_1 \\ a_3 & b_3 \end{vmatrix} - d_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}}{DET}$$

$$DET = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$$

The formula is quite "horrible" one and for students not acceptable as it is too complex and they do not see from the formula comes from.

As outline below, there is a quite simple geometrical explanation and solution. So the first question is how to find the "starting" point $\boldsymbol{x}_0$ of the line $\boldsymbol{p}$ given by two planes $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$. If a robust solution is required a user should be prevented from a selection of some "parameters".

Let us imagine that there exists a plane $\boldsymbol{\rho}_0$, whose normal vector is given as $\boldsymbol{s} = \boldsymbol{n}_1 \times \boldsymbol{n}_2$.

It means that its position needs to be "fixed" in the space. As there is no other requirement on this plane, we can "fix" it so it passes through the origin of the coordinate system, i.e. the plane $\boldsymbol{\rho}_0$ is given as

$$\boldsymbol{\rho}_0 = [a_0, b_0, c_0 : 0]^T$$

and the line $\boldsymbol{p}$ is orthogonal to the plane $\boldsymbol{\rho}_0$ – resulting in a robust geometric position.

Now, the intersection point of three planes is the point $\boldsymbol{x}_0$ we are looking for. Coordinates of the point $\boldsymbol{x}_0$ are determined by generalized cross-product as

$$\boldsymbol{x}_0 = \boldsymbol{\rho}_1 \times \boldsymbol{\rho}_2 \times \boldsymbol{\rho}_0$$

As this formula is very compact and the cross-product is a GPU instruction, it is suitable for GPU use. See the Appendix for the extended cross-product GPU implementation.

From the formulation presented above, it can be seen that it is not only very simple, easy to understand and remember, but also easy to implement. It is obvious that the point $\boldsymbol{x}_0$ is also the closest point on the line to the origin, too. As a result the Plücker coordinates formulation of this problem solution is not needed when looking for such properties.

VII. WINDOW CLIPPING

Line or line segment clipping against rectangular window or convex polygon in E^2 is a basic operation in computer graphics. There are well-known Cohen-Sutherland algorithm and many other algorithms. Some of these well-known algorithms are not easy to implement due to their complexity. However, there is a simple and effective solution based on projective representation and the line clipping algorithm can be described using 7 lines only.

The algorithm is based on classification of the window vertices resulting in a binary code which is the address to TAB_1 and TAB_2 tables, where indices of intersected edges are stored. Coordinates of intersection points are computed as a cross-product of the given line and intersected edges.

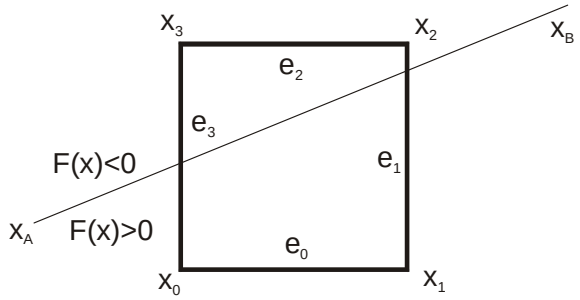


Figure 4. Line clipping by a rectangular window

procedure CLIP_L; { **input**: $\boldsymbol{x}_A$, $\boldsymbol{x}_B$ }
$\boldsymbol{x}_A=[x_A,y_A:w_A]^T$, $\boldsymbol{x}_B=[x_B,y_B:w_B]^T$
$\boldsymbol{x}_A$, $\boldsymbol{x}_B$ –in homogeneous coordinates
the **EXIT** statement ends the procedure
{ $\boldsymbol{p}$:= $\boldsymbol{x}_A$ x $\boldsymbol{x}_B$; { ax+by+c = 0; $\boldsymbol{p}$ = $[a,b,c]^T$ }
for k:=0 **to** N-1 do # $x_k=[x_k,y_k:w_k]^T$ #
if $\boldsymbol{p}^T\boldsymbol{x}_k \geq 0$ **then** $\mathbf{c}_k$:=1 **else** c_k:=0;
if $\mathbf{c} = [0000]^T$ **or** $\mathbf{c} = [1111]^T$ **then EXIT**;
i:= TAB_1[**c**]; j:= TAB_2[**c**];
$\boldsymbol{x}_A$:= $\boldsymbol{p}$ x $\boldsymbol{e}_i$; $\boldsymbol{x}_B$:= $\boldsymbol{p}$ x $\boldsymbol{e}_j$;
DRAW ($\boldsymbol{x}_A$; $\boldsymbol{x}_B$)
}

Algorithm 2

Where N is a number of edges of the clipping window TAB_1 and TAB_2 are constant tables with window edges classifications, for details see [15].

 ISBN: 978-1-61208-184-7

For the situation at Fig. 4, CODE=[0011]=3, TAB_1=1 and TAB_2=3. The algorithm itself is easy to explain and implement, too.

The line segment clipping algorithm is a little bit longer, but still easy to implement, see [15], and to modify for line or line segment clipping by a convex polygon as well. It should be noted that due to the principle of duality, line clipping by a convex polygon is dual to a point-in-polygon test, which is of $O(lgN)$ complexity. Line clipping algorithm of $O(lgN)$ complexity is more complex, see [16].

VIII. CONCLUSION

Application of projective geometry principles to the computational pipeline, especially in the field of geometry and computer graphics can bring new algorithms that are more robust and faster even for the Euclidean space. Due to the formulation, is very convenient for vector-vector or matrix-vector architectures, like GPU and a significant speed-up can be expected as well. Projective geometry can easily explain several methods in a more simple way and also provide new formula and geometric representation which contributes to students' better understanding.

ACKNOWLEDGMENT

The author would like to thank students and colleagues at the VSB-Technical University and University of West Bohemia for discussions and suggestions, reviewers for their critical comments and constructive recommendations.

This work was partially supported by SGS, VSB-Technical University of Ostrava, Czech Republic, under the grant No. SP2011/163.

REFERENCES

[1] D. van Arsdale, "Homogeneous Transformation Matrices for Computer Graphics," Computers & Graphics, Vol.18, No. 2, March-April 1994, pp. 177-191, 1994.

[2] M.M.S. Coxeter, "Projective Geometry," Toronto: University of Toronto, 2nd edition, 1974.

[3] R. Goldman, "Intersection of Three Planes," Graphics Gems (Ed. A.Glassner-), Academic Press, pp. 305-310, 1990.

[4] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2000.

[5] M. Johnson, "Proof by Duality: or the Discovery of "New" Theorems," Mathematics Today, December, 1996.

[6] M.E. Loaiza, A.B. Raposo and M. Gattass, "Multi-camera Calibration Based on an Invariant Pattern," Computers & Graphics, Vol. 35, Issue 2, pp. 198-207, 2011.

[7] J.R. Miller, "Vector Geometry for Computer Graphics," IEEE Computer Graphics and Applications, Vol. 19, No. 3., 1999.

[8] V. Skala, "Geometric Computation, Duality and Projective Space," IW-LGK workshop proceedings, pp.105-111, Dresden University of Technology, 2011.

[9] V. Skala and V. Ondracka, "A Precision of Computation in the Projective Space," Recent Researchers in Computer Science, WSEAS, pp. 35-40, 2011.

[10] V. Skala, "Duality and Intersection Computation in Projective Space with GPU support," ASM 2010 Conf., pp. 66-71, NAUN, 2010.

[11] V. Skala, "Computation in Projective Space," MAMETICS 2009 Conf., pp. 152-157, WSEAS, 2009.

[12] V. Skala, "Barycentric Coordinates Computation in Homogeneous Coordinates," Computers & Graphics, Elsevier, Vol. 32, No. 1, pp. 120-127, 2008.

[13] V. Skala, "Intersection Computation in Projective Space using Homogeneous Coordinates," International Journal on Image and Graphics, Vol. 8, No. 4, pp. 615-628, 2008.

[14] V. Skala, "Length, Area and Volume Computation in Homogeneous Coordinates," International Journal of Image and Graphics, Vol. 6., No. 4, pp. 625-639, 2006.

[15] V. Skala, "A New Approach to Line and Line Segment Clipping in Homogeneous Coordinates," The Visual Computer, Vol.21, No.11, pp.905-914, Springer Verlag, 2005.

[16] V. Skala, "O(lg N) Line Clipping Algorithm in E^2," Computers & Graphics, Pergamon Press, Vol.18, No.4, 1994.

[17] J. Stolfi, "Oriented Projective Geometry," Academic Press, 2001.

[18] F. Yamaguchi, "Computer-Aided Geometric Design," Springer Verlag, 2002.

[19] Y. Yu, "Efficient Visibility Processing for Projective Texture Mapping," Computers & Graphics, Vol. 23, No. 2, pp. 245-253, 1999.

WEB references

[20] Softsurfer http://softsurfer.com/ <retrieved on 2011-12-05>

APPENDIX

The cross-product in 4D defined as

$$\boldsymbol{x_1} \times \boldsymbol{x_2} \times \boldsymbol{x_3} = det \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} & \boldsymbol{l} \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{vmatrix}$$

can be implemented in Cg/HLSL on GPU as follows:

```
float4 cross_4D(float4 x1, float4 x2, float4 x3)
{
        float4 a;
        a.x=dot(x1.yzw, cross(x2.yzw, x3.yzw));
        a.y=-dot(x1.xzw, cross(x2.xzw, x3.xzw));
        // or a.y=dot(x1.xzw, cross(x3.xzw, x2.xzw));
        a.z=dot(x1.xyw, cross(x2.xyw, x3.xyw));
        a.w=-dot(x1.xyz, cross(x2.xyz, x3.xyz));
        // or a.w=dot(x1.xyz, cross(x3.xyz, x2.xyz));

        return a;
}
```

or more compactly

```
float4 cross_4D(float4 x1, float4 x2, float4 x3)
{
        return ( dot(x1.yzw, cross(x2.yzw, x3.yzw)),
        -dot(x1.xzw, cross(x2.xzw, x3.xzw)),
        dot(x1.xyw, cross(x2.xyw, x3.xyw)),
        -dot(x1.xyz, cross(x2.xyz, x3.xyz)) );
}
```

 ISBN: 978-1-61208-184-7

Quaternion Lifting Scheme for Multi-resolution Wavelet-based Motion Analysis

Agnieszka Szczesna
The Silesian University of Technology
Institute of Informatics
Gliwice, Poland
agnieszka.szczesna@polsl.pl

Janusz Slupik
The Silesian University of Technology
Institute of Mathematics
Gliwice, Poland
janusz.slupik@polsl.pl

Mateusz Janiak
The Silesian University of Technology
Institute of Informatics
Gliwice, Poland
mateusz.janiak@polsl.pl

***Abstract*—This paper considers human body motion analysis as local changes of orientations in hierarchical skeleton parts over time. Possible approaches by applying multiresolution analysis in form of a second generation wavelet transform directly on quaternion signal are shown. Quaternions in terms of motion analysis are a more efficient representation of rotation than Euler angles. This paper presents that the lifting scheme can be efficiently applied directly to quaternions. Lifting scheme building blocks for the quaternion Haar and linear transformation are presented.**

Keywords-quaternions; multi-resolution analysis; wavelet transform; lifting scheme; quaternion interpolation; motion analysis.

I. INTRODUCTION

Human body motion synthesis and analysis are very challenging tasks and a very popular research domain. The most precise measurements of motion data are obtained by motion capture systems. We cooperate with a high tech motion capture laboratory having dedicated hardware capable of performing motion acquisition. It can acquire motion data through simultaneous and synchronous measurement and recording of motion kinematics, muscle potentials by electromyography, ground reaction forces and video streams in high definition format and the HML supporting system which allows for storing, playing and browsing data. The data from the above mentioned subsystems are large and accurate, allowing for a thorough analysis of motion. Techniques of analysis of such data can be potentially applied in:

- Medicine - diagnosis and verification of a medical treatment;
- Entertainment - realistic animations;
- Sport - new training techniques;
- Security - people recognition based on their body movement.

In this paper, we present our approaches in performing motion analysis with multi-resolution techniques based on rotations of joints over time written in the form of quaternion signal. For this reason, we are trying to use a second generation wavelet transform constructed by the lifting scheme for quaternion rotation representation. Using the quaternion lifting scheme based on the quaternion algebra we can work directly on correlated motion data. This is in opposition to the methods presented in the literature where the filters work on Euler angles as three non-correlated components. Also, the example application of multi-resolution for denoising data is presented.

Section II describes the main assumptions of multi-resolution wavelet analysis of motion data and presents a short review of solutions presented in literature. Section III presents general information about the second generation wavelets and the lifting scheme as a simple construction tool of such wavelets. Section IV focuses on quaternion interpolating methods. Section V describes the construction of the lifting schema blocks for Haar and linear quaternion transformation and presents some results. Section VI presents example application of result quaternion multi-resolution representation. The last section is a conclusion.

II. MULTI-RESOLUTION WAVELET ANALYSIS OF MOTION DATA

The main idea of the multi-resolution transformation is to represent a signal coarse to fine hierarchy. The input signal is decomposed into coarse base data (global pattern of signal) and a hierarchy of detail coefficients. The result multi-resolution representation can be based of many algorithms such as [10], [14]: denoising, filtering (smoothing, enhancement), compression, feature detection and multi-resolution editing.

Most of the solutions are based on processing orientation data as three non-correlated signals defined by Euler angles. In [1], spatial filters for orientation data are proposed. A similar solution based on a digital filter bank technique is in [2]. In [3], the cubic interpolating bi-orthogonal wavelet basis, implemented as lifting scheme blocks, are used to compression skeletal animation data. Temporal coherence is exploited by this wavelet transform. The B-spline wavelet for unit quaternion is used for smoothing motion data in [4].

Quaternion wavelets are also proposed for phase based stereo matching for uncalibrated images [5]. This solution is based on a bi-orthogonal filter bank, where the real valued image signal is convolved with an analytic quaternion wavelet filter, to construct the 2D analytic signal.

 ISBN: 978-1-61208-184-7

In [6] and [7], the quaternion multiplier of plane rotations, inspired by a factorization algorithm, is implemented. This proposition considers the factorization of a quaternion multiplication matrix into lifting scheme steps. Our work is different because we work directly on quaternion signal as representing orientation changes over time and we design lifting scheme blocks using the convention of second generation wavelets. Our main task is to analyze human motion.

III. Second generation wavelets and lifting scheme

The lifting scheme [8], [9] is a simple but powerful tool to construct a wavelet transform. The main advantage of this solution is the possibility of building wavelet analysis on non-standard structures of data (irregular samples, bounded domains, curves, surfaces) while keeping all powerful properties as speed and good ability of approximation [10], [11], [12], [13]. This generalization are called as second generation wavelets [14]. They are not necessarily translated and dilated of one function (mother function). In this meaning, the lifting scheme also considers non-linear and data-adaptive multi-resolution decompositions.

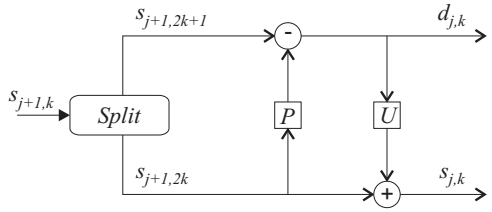


Figure 1. The forward lifting scheme

A general lifting scheme (Figure 1) consists of three types of operations:

- **Split**: splits input dataset into two disjoint sets of even and odd indexed samples. The definition of the lifting scheme does not impose any restriction on how the data should be split nor on the relative size of each subsets.
- **Predict**: predicts samples with odd indexes based on even indexed samples. Next the odd indexed input value is replaced by the offset (difference) between the odd value and its prediction.
- **Update**: updates the output, so that coarse-scale coefficients have the same average value as the input samples. This step is necessary for stable wavelet transform [14].

These calculations can be performed in-place. In all stages input samples can be overwritten by output samples of that step. The inverse transform (Figure 2) is easy to find by reversing the order of operations and flipping the signs.

IV. Interpolating quaternions

Quaternions [15]–[17] are structures allowing descriptions of vectors relations. They are commonly used (mainly in computer graphics) for performing rotations.

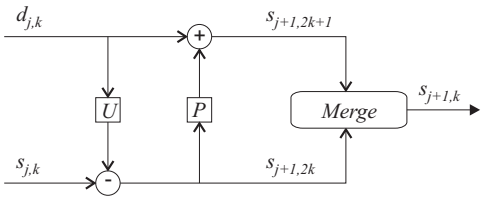


Figure 2. The reverse lifting scheme

Quaternion q is denoted as: $q = [s, v], s \in R, v \in R^3$. Here, s represents the *scalar part* and v is the *imaginary part* of the quaternion. In more details this representation can be given as:

$$q = xi + yj + zk + w,$$
$$x, y, z, w \in R, i^2 = j^2 = k^2 = ijk = -1.$$

Now, w is the scalar part and $[x, y, z]$ is the imaginary part. The space of quaternions is denoted as H.

We are working with unit quaternions; therefore, all interpolated quaternions are assumed to be unit quaternions. Interpolation step h is in range $[0; 1]$. Based on [18], the following quaternion interpolation methods can be distinguished:

- lerp - Computationally the most efficient from presented those but gives poor quality in generated rotation quaternions and does not guarantee unit quaternions as a result. Normalization is required. Generated movements have sharp ending motion so the movement of the body is not considered to be smooth.

$$lerp(q_i, q_{i+1}, h) = q_i * h + q_{i+1} * (1 - h)$$

- slerp - Ensures unit quaternion as a result. Unfortunately on the endings moves are still sharp.

$$slerp(q_i, q_{i+1}, h) = q_i(q_i^* q_{i+1})^h$$

- squad - Computational very demanding but gives very smooth movement after interpolation. No rapid changes in movement are noticeable at the interpolation range endings. This method is inspired by splines.

$$squad(q_i, q_{i+1} s_i, s_{i+1}, h) = slerp(slerp(q_i, q_{i+1}, h), slerp(s_i, s_{i+1}, h), 2h(1 - h))$$
$$s_i = q_i \exp\left(-\frac{\log(q_i^{-1} q_{i+1}) + \log(q_i^{-1} q_{i-1})}{4}\right)$$

- shoemake-bezier - Interpolation method is based on the De Castlejau algorithm. More details might be found in [19].

 ISBN: 978-1-61208-184-7

$$bezier(q_i, q_{i+1}, s_i, s_{i+1}, h) = slerp(slerp(q_{11}, q_{12}, h), slerp(q_{12}, q_{13}, h), h)$$
$$q_{11} = slerp(q_i, s_i, h)$$
$$q_{12} = slerp(s_i, s_{i+1}, h)$$
$$q_{13} = slerp(s_{i+1}, q_{i+1}, h)$$
$$s_i = q_i \exp\left(-\frac{\log(q_i^{-1}q_{i+1}) + \log(q_i^{-1}q_{i-1})}{4}\right)$$

V. Quaternion Lifting Scheme

Using the quaternion lifting scheme based on the quaternion algebra we can work directly on correlated motion data. This is in opposition to the methods presented in the literature where the filters work on Euler angles as three non-correlated components.

A short comment about the interpretation of details for the lifting scheme must be given here, as currently the coefficients are represented by quaternions. The goal of the prediction step is to produce values as close as possible to the given data. It means the smaller the difference (detail coefficient), the better the prediction step is. For quaternions this difference should tend to a zero rotation quaternion - $[1, [0, 0, 0]]$. If all coefficients are close to this value it means we have found a closed form description of the analyzed movement and this function might be used for such movement reproduction and analysis. Additionally small detail values suggest that the data is strongly correlated. On the other hand, if all coefficients have similar, but rather large values it is also possible that either the prediction step poorly describes data correlation or mean signal value is not maintained in the next resolutions.

Input motion signal with length 2^n is a set of normalized quaternions. In the split block this signal is divided into even and odd indexed samples: $..., o_i^j, e_i^j, o_{i+1}^j, e_{i+1}^j, ...$ The upper index j indicates the step scheme (the level of resolution).

In this paper, we give two propositions for the quaternion lifting schemes, which are fully and easy reversible and allow in-place computations. Those schemes also preserve the average signal at each resolution level.

A. Motion data

Data for analysis are obtained from the Human Motion Laboratory of the Polish-Japanese Institute of Information Technology in Bytom (Poland). The data (Figures 3 and 4) represent knee joint motion sampled at 100Hz. To better visualize the motion data, we have chosen the three Euler angles plots.

B. Quaternion Haar lifting schema

In literature the most basic lifting scheme is the Haar wavelet transformation. It predicts odd indexed samples with corresponding even indexed samples. The lifting scheme steps are the following:

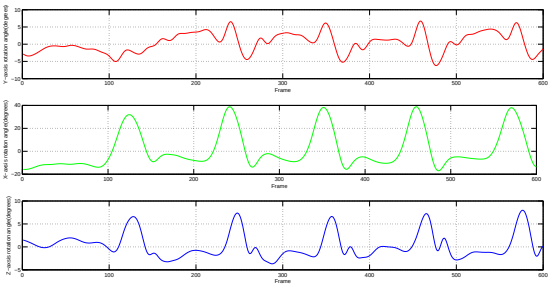


Figure 3. The Euler angles of knee joint motion data obtained from the Human Motion Laboratory of the Polish-Japanese Institute of Information Technology in Bytom (Poland).

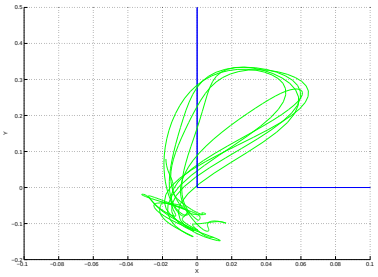


Figure 4. The quaternion curve of knee joint motion data obtained from the Human Motion Laboratory of the Polish-Japanese Institute of Information Technology in Bytom (Poland).

- Prediction step:

$$o_i^j = o_i^{j+1} \cdot SLERP(e_i^{j+1}, o_i^{j+1}, 0.5)^{-1}$$

- Update step:

$$e_i^j = o_i^j \cdot e_i^{j+1}$$

The reverse lifting scheme steps:

- Undo prediction step:

$$o_i^{j+1} = o_i^j \cdot e_i^j$$

- Undo update step:

$$e_i^{j+1} = SLERP(o_i^{j+1}, e_i^j, 2)$$

The results of the Haar transformation are presented in Figure 5 and 6. These are plots of Euler angles at the first and forth level of resolution (after first and forth step of the lifting schema) and details for such levels (differences between removed quaternions and predicted by the lifting schema).

 ISBN: 978-1-61208-184-7

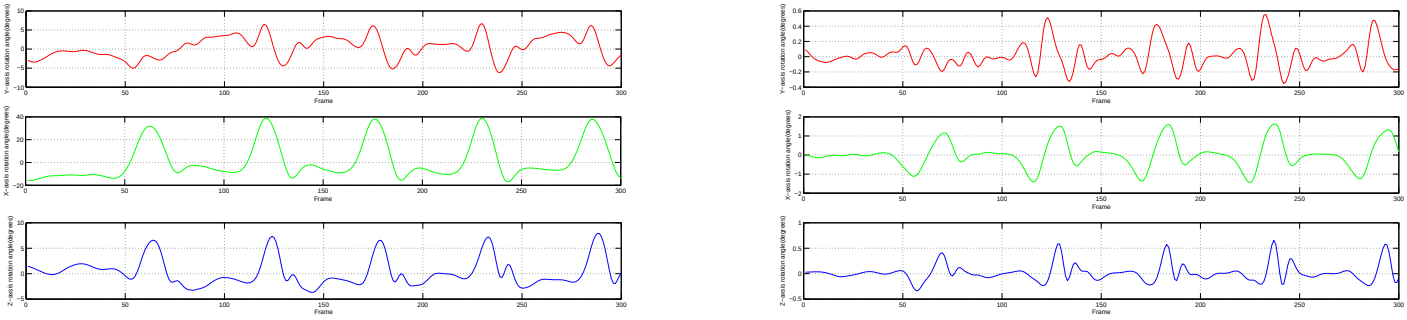


Figure 5. The first level of resolution computed by the Haar lifting schema: data (left plot) and details (right plot).

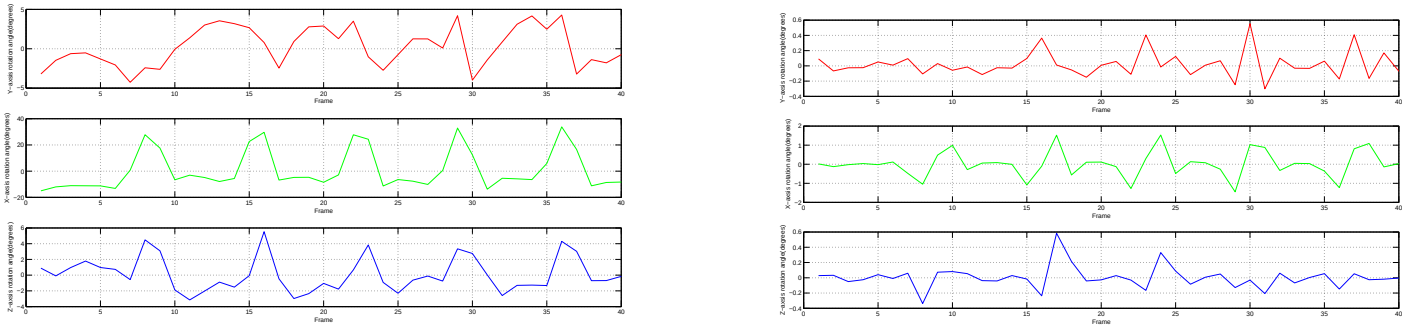


Figure 6. The forth level of resolution computed by the Haar lifting schema: data (left plot) and details (right plot).

C. Quaternion linear lifting schema

The next common prediction step is the linear interpolation between surrounding values. The lifting scheme steps are the following:

- Prediction step:

$$o_i^j = SLERP(e_i^{j+1}, e_{i+1}^{j+1}, 0.5)^{-1} \cdot o_i^{j+1}$$

- Update step:

$$e_i^j = e_i^{j+1} \cdot (SLERP(o_{i-1}^j, o_i^j, 0.5))^{0.5}$$

The reverse lifting scheme steps are:

- Undo update step:

$$e_i^{j+1} = e_i^j \cdot (SLERP(o_{i-1}^j, o_i^j, 0.5))^{0.5}$$

- Undo prediction step:

$$o_i^{j+1} = SLERP(e_i^{j+1}, e_{i+1}^{j+1}, 0.5) \cdot o_i^j$$

The results of the linear transformation are presented in Figure 7 and 8. These are plots of Euler angles of data at the first and forth level of resolution (after first and forth step of lifting schema) and details for such levels (differences between removed quaternions and predicted by the lifting schema).

VI. Example application - Denoising motion data

Denoising methods rely on removing the high frequency component of a signal, which consists of noise. The simplest method is to set to zero wavelet coefficients representing high frequencies from the first few levels of decomposition. In the quaternion lifting schema, details are set to unit quaternion. Another method is based on threshold methods, which change the wavelets coefficients selected on the basis of some threshold value. Determining the value of threshold for the quaternions domain requires further research.

Motion data with artificially added white Gaussian noise (Figure 9) was decomposed by a lifting schema into two levels of resolutions with detail quaternions coefficients. Coefficients from the first levels of transformation are small and mostly contain information about high frequency component. We can see this in Figure 10. In the reverse lifting scheme those coefficients were set to the unit quaternion. The results of the Haar and linear lifting schemes are presented in Figure 11. Because the prediction step in the linear lifting schema is based on two adjacent samples, the results of denoising for this schema are much better.

VII. Summary and future work

We have shown with results of our experiments that the lifting scheme can be efficiently applied to quaternions. This

 ISBN: 978-1-61208-184-7

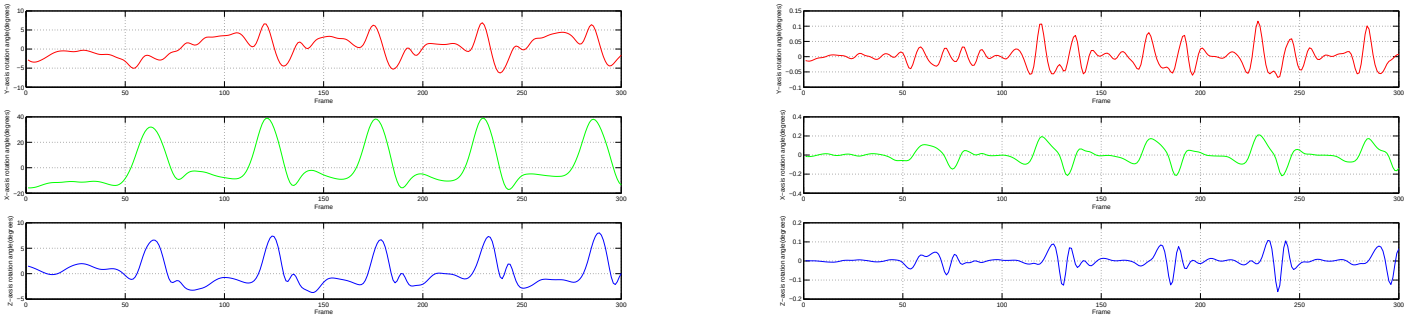


Figure 7. The first level of resolution computed by the linear lifting schema: data (left plot) and details (right plot).

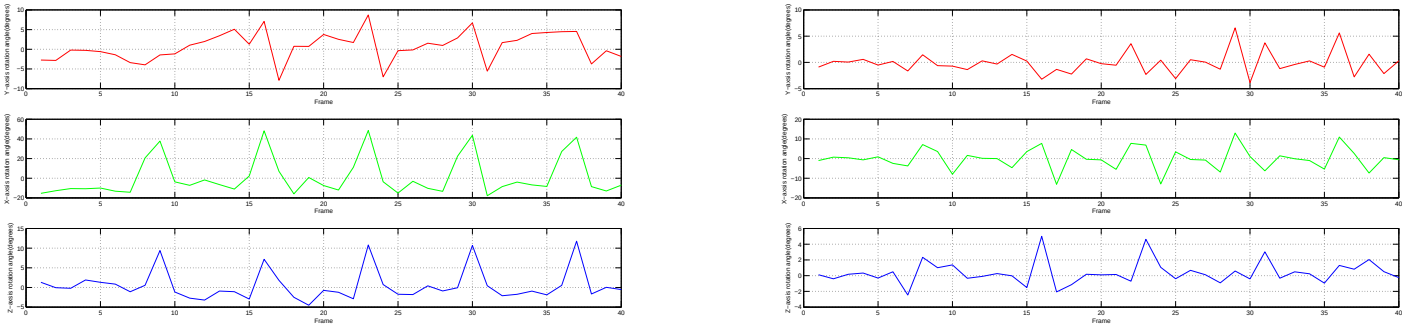


Figure 8. The forth level of resolution computed by the linear lifting schema: data (left plot) and details(right plot).

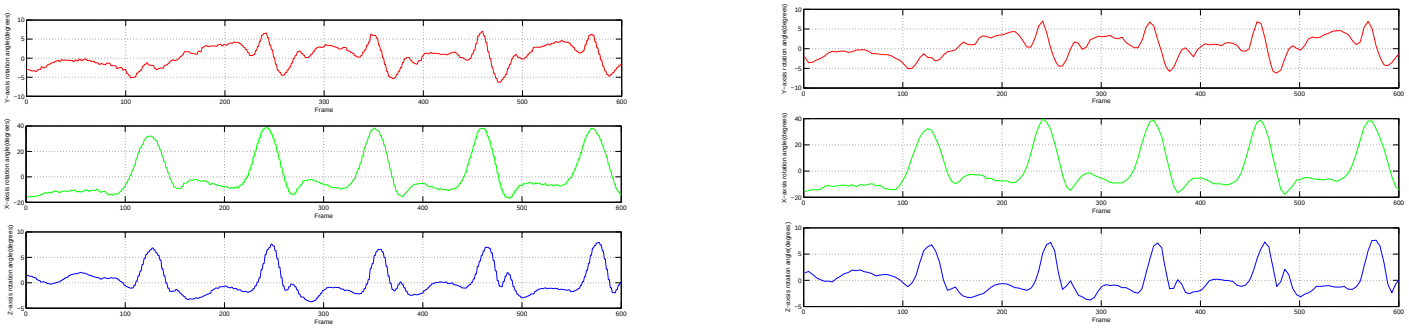


Figure 11. The motion data after removal of noise based on the Haar (left plot) and the linear lifting schema (right plot).

allows us to quickly and in place multi-resolution analysis applied directly on quaternion signal, which is a description of orientation changes over time. Using quaternion algebra properly, following quaternion space laws, data correlation would be captured at each level of resolution. This is more efficient than analyzing changes in time of each angle as a non-correlated signal.

Moreover, we are looking forward to creating proper update and predict steps for more complex quaternion interpolation methods mentioned in this paper, but not included in our research.

The results of multi-resolution representation can also be a base for different motion processing algorithms as a generalization of signal processing tools. Examples can be filtering, feature detection and compression. As an example, the very simple denoising algorithm was presented in this paper.

 ISBN: 978-1-61208-184-7

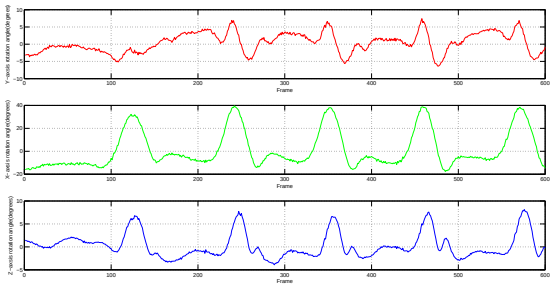


Figure 9. The signal with added white Gaussian noise.

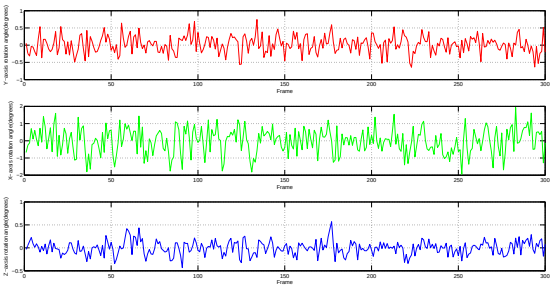


Figure 10. The Euler angles of the first level of details coefficients computed by the linear lifting scheme.

ACKNOWLEDGMENT

Used application is based on QuTEM [20] implemented by K. Lebek.

This work was partly supported by the European Community from the European Social Fund (*UDA-POKL.04.01.01-00-106/09*) and by grant (number *NN518289240*, "The development of a quantitative measurement of motion, rationalizing, on the basis of multimodal motion measurement, UPDRS subjective criteria in order to improve pre and post DBS implantation diagnostics for patients with Parkinson's disease.") awarded by the National Science Centre of the Polish Ministry of Science and Higher Education.

REFERENCES

[1] J. Lee and S.Y. Shin, Multiresolution Motion Analysis and Synthesis. Technical paper, 2000, (retrieved: 12, 2010)

[2] A. Bruderlin and L. Williams, Motion signal processing. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 1995, pp. 97-104

[3] P. Beaudoin, P. Poulin and M. Panne, Adapting wavelet compression to human motion capture clips. Proceedings of Graphics Interface, 2007, pp. 313-318

[4] Ch.-Ch. Hsieh, B-spline wavelet-based motion smoothing. Computers and Industrial Engineering, 41(1), 2001, pp. 59-76

[5] J. Zhou, Y. Xu, and X. Yang, Quaternion wavelet phase based stereo matching for uncalibrated images. Pattern Recogn. Lett. 28(12), 2007, pp. 1509-1522

[6] M. Parfieniuk and A. Petrovsky, Inherently lossless structures for eight- and six-channel linear-phase paraunitary filter banks based on quaternion multipliers. Signal Processing (Elsevier), 90(6), 2010, pp. 1755-1767

[7] M. Parfieniuk and A. Petrovsky, Quaternion Multiplier Inspired by the Lifting Implementation of Plane Rotations. IEEE Transactions on Circuits and Systems I, 57(10), 2010, pp. 2708-2717

[8] W. Sweldens, The lifting scheme: A construction of second generation wavelets. SIAM J. Math. Anal, 29(2), 1997, pp. 511-546

[9] W. Sweldens, The Lifting Scheme: A new philosophy in biorthogonal wavelet constructions. Wavelet Applications in Signal and Image Processing III, 1995, pp. 68-79

[10] E.J. Stollnitz, T. DeRose, and D. H. Salesin, Wavelets for Computer Graphics: Theory and Applications. Morgan Kaufmann, 1996

[11] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens, Wavelets on Irregular Point Sets. Royal Society, 357, 1999, pp. 2397-2413

[12] I. Guskov, W. Sweldens, and P. Schröder, Multiresolution Signal Processing for Meshes. Computer Graphics Proceedings, 1999, pp. 325-334

[13] A. Szczesna, The Multiresolution Analysis of Triangle Surface Meshes with Lifting Scheme. In Computer Vision/Computer Graphics Collaboration Techniques, Proceedings of MIRAGE, Gagalowicz A., Philips W. (editors), Springer, LNCS 4418, 2007, pp. 274-282

[14] M. Jansen and P. Oonincx, Second Generation Wavelets and Applications. Springer, 2005

[15] J.C. Hart, G.K. Francis, and L.H. Kauffman, Visualizing Quaternion Rotation. ACM Trans. Graph. 13(3), 1994, pp. 256-276

[16] K. Shoemaker, Animating Rotation with Quaternion Curves. SIGGRAPH Comput. Graph., 19(3), 1985, pp. 245-254

[17] A. Sudbery, Quaternionic analysis. Proc. Camb. Phil. Soc. 85, 1979, pp. 199-225

[18] E.B. Dam, M. Koch, and M. Lillholm, Quaternions, Interpolation and Animation. University of Copenhagen, Denmark. Technical Report, 1998, (retrieved: 12, 2010)

[19] C. Buckley, Bezier Curves for Camera Motion. Department of Computer Science, Trinity College, Dublin , 1994, (retrieved: 12, 2010)

[20] M.P. Johnson, Exploiting Quaternions to Support Expressive Interactive Character Motion. Massachusetts Institute of Technology, PhD Thesis, 2003

 ISBN: 978-1-61208-184-7