IARIA

ICONS 2015

The Tenth International Conference on Systems

ISBN: 978-1-61208-399-5

EMBEDDED 2015

International Symposium on Advances in Embedded Systems and Applications

April 19 - 24, 2015

Barcelona, Spain

ICONS 2015 Editors

Leszek Koszalka, Wroclaw University of Technology, Poland
Pascal Lorenz, University of Haute Alsace, France

ICONS 2015

Foreword

The Tenth International Conference on Systems (ICONS 2015), held between April 19th-24th, 2015 in Barcelona, Spain, 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).

ICONS 2015 also featured the following Symposium:

- EMBEDDED 2015, The International Symposium on Advances in Embedded Systems and Applications

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

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ICONS 2015 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICONS 2015 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 also hope Barcelona provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

ICONS 2015 Advisory Committee:

Raimund Ege, Northern Illinois University, USA
Hermann Kaindl, Vienna University of Technology, Austria
Leszek Koszalka, Wroclaw University of Technology, Poland
Marko Jäntti, University of Eastern Finland, Finland

EMBEDDED 2015 Advisory Committee:
Sabina Jeschke, RWTH Aachen University, Germany
I-Cheng Chang, National Dong Hwa University, Taiwan
Ralf-D. Kutsche, TU Berlin / Fraunhofer FOKUS institute, Germany

ICONS 2015

Committee

ICONS Advisory Committee

Raimund Ege, Northern Illinois University, USA
Hermann Kaindl, Vienna University of Technology, Austria
Leszek Koszalka, Wroclaw University of Technology, Poland
Marko Jäntti, University of Eastern Finland, Finland

ICONS 2015 Technical Program Committee

Mehmet Aksit (Akşit), University of Twente - Enschede, The Netherlands
Marco Aiello, University of Groningen, The Netherlands
Abdallah Al Sabbagh, University of Technology - Sydney (UTS), Australia
Cristina Alcaraz, University of Malaga, Spain
Eduardo Alonso, City University London, UK
Giner Alor Hernández, Instituto Tecnológico de Orizaba - Veracruz, México
César Andrés, Universidad Complutense de Madrid, España
Luis Anido-Rifon, University of Vigo, Spain
Mark Austin, University of Maryland, College Park, USA
Javier Bajo Pérez, Universidad Politécnica de Madrid, Spain
Lubomir Bakule, Institute of Information Theory and Automation of the ASCR, Czech Republic
Zbigniew Banaszak, Warsaw University of Technology | Koszalin University of Technology, Poland
Jacob Barhen, Oak Ridge National Laboratory, USA
Molka Becher, CEA-Leti/DACLE/LIALP Lab | Verimag Lab | Grenoble University, France
Nicolas Belanger, Eurocopter Group, France
Ateet Bhalla, Oriental Institute of Science & Technology, Bhopal, India
Jun Bi, Tsinghua University - Beijing, China
Francesco Bianconi, University of Perugia, Italy
Freimut Bodendorf, University of Erlangen-Nuremberg, Germany
Mietek Brdys, University of Birmingham, UK
Alisson Brito, Universidade Federal da Paraiba (UFPB), Brazil
Mario Cannataro, University "Magna Græcia" of Catanzaro - Germaneto, Italy
M. Emre Celebi, Louisiana State University in Shreveport, USA
Chi-Hua Chen, National Chiao Tung University, Taiwan , R.O.C.
Albert M. K. Cheng, University of Houston, USA
Ding-Yuan Cheng, National Chiao Tung University, Taiwan , R.O.C.
Sunil Choenni, Research and Documentation Centre - Ministry of Security and Justice, Netherlands
Lawrence Chung, University of Texas at Dallas, USA
Carlos J. Costa, ISCTE - University Institute of Lisbon, Portugal
Nicolas Damiani, Eurocopter Group, France
Peter De Bruyn, Universiteit Antwerpen, Belgium
Lucio De Paolis, University of Salento, Italy

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
Sergio Escalera, University of Barcelona, Spain
Andras Farago, The University of Texas at Dallas, USA
Miguel Franklin de Castro, Federal University of Ceará, Brazil
Marta Franova, CNRS & LRI, France
Matthias Galster, University of Canterbury, New Zealand
Subhashini Ganapathy, Wright State University, USA
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
Mohanad Halaweh, University of Dubai, UAE
José Luis Herrero Agustin, University of Extremadura, Spain
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Yo-Ping Huang, National Taipei University of Technology - Taipei, Taiwan
Wen-Jyi Hwang, National Taiwan Normal University - Taipei, Taiwan
Tomasz Hyla, West Pomeranian University of Technology, Poland
Marko Jäntti, University of Eastern Finland, Finland
Jiri Jaros, Australian National University, Australia
Jaroslav Kadlec, Brno University of Technology, Czech Republic
Hermann Kaindl, Vienna University of Technology, Austria
Krishna Kant, George Mason University, USA
Fu-Chien Kao, Da-Yeh University, Taiwan
Andrzej Kasprzak, Wroclaw University of Technology, Poland
Abdelmajid Khelil, Huawei Research, Germany
Leszek Koszalka, Wroclaw University of Technology, Poland
Eiji Kawai, National Institute of Information and Communications Technology, Japan
Radek Kuchta, Brno University of Technology, Czech Republic
Wim Laurier, Université Saint-Louis / Ghent University, Belgium
Frédéric Le Mouël, INRIA/INSA Lyon, France
Lenka Lhotska, Czech Technical University in Prague, Czech Republic
David Lizcano, Open University of Madrid (UDIMA), Spain
Andrei Lobov, Tampere University of Technology, Finland
Pascal Lorenz, University of Haute Alsace, France
Ivan Lukovic, University of Novi Sad, Serbia
Jia-Ning Luo (羅嘉寧) Ming Chuan University, Taiwan
José Manuel F. Machado, University of Minho, Portugal
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
D. Manivannan, University of Kentucky, USA
Juan Martínez-Miranda, Universidad Politecnica de Valencia, Spain
Patrick Meumeu Yomsi, CISTER Research Center, IPP Hurray-ISEP, Portugal
Roberto Montemanni, University of Applied Sciences of Southern Switzerland (SUPSI), Switzerland
Fernando Moreira, Universidade Portucalense, Portugal
Daoudi Mourad, University of Sciences and Technology, Algeria

Tito Valencia, University of Vigo, Spain
Dirk van der Linden, University of Antwerp, Belgium
Lorenzo Verdoscia, ICAR - CNR - Napoli, Italy
Dario Vieira, EFREI, France
Hironori Washizaki, Waseda University, Japan
Wei Wei, Xi'an University of Technology, P.R. China
Yair Wiseman, Bar-Ilan University, Israel
Kuan Yew Wong, Universiti Teknologi Malaysia (UTM), Malaysia
Heinz-Dietrich Wuttke, Ilmenau University of Technology, Germany
Xiaodong Xu, Beijing University of Posts and Telecommunications, China
Linda Yang, University of Portsmouth, UK
Sameh Yassin, Cairo University, Egypt
Chang Wu Yu (James), Chung Hua University, Taiwan
Wai YuenSzeto, University of Hong Kong, Hong Kong
Sherali Zeadally, University of Kentucky, USA
Xiangmin Zhang, Wayne State University, USA
Wenjie Zhang, University of New South Wales - Sydney, Australia
Ying Zhang, University of New South Wales - Sydney, Australia
Ty Znati, University of Pittsburgh, USA
Dawid Zydek, Idaho State University, USA

EMBEDDED 2015 Advisory Committee

Sabina Jeschke, RWTH Aachen University, Germany
I-Cheng Chang, National Dong Hwa University, Taiwan
Ralf-D. Kutsche, TU Berlin / Fraunhofer FOKUS institute, Germany

EMBEDDED 2015 Program Committee Members

Arnulfo Alanis, Instituto Tecnológico de Tijuana, Mexico
Cristina Alcaraz, Universidad de Malaga, Spain
Mohamed Bakhouya, International University of Rabat, Morocco
Fadila Bentayeb, University of Lyon 2, France
Patrick Brezillon, LIP6 - University Pierre and Marie Curie (UPMC), France
Juan Carlos Cano Escribá, Universitat Politècnica de València, Spain
I-Cheng Chang, National Dong Hwa University, Taiwan
Jiann-Liang Chen, National Taiwan University of Science and Technology, Taiwan
Li-Der Chou, National Central University Taoyuan, Taiwan
Jianguo Ding, University of Skövde, Sweden
Luis Javier García Villalba, Universidad Complutense de Madrid, Spain
Chia Hung Yeh, National Sun Yat-Sen University, Taiwan
Sabina Jeschke, RWTH Aachen University, Germany
Alexandra Kees, Hochschule Bonn-Rhein-Sieg, Germany
Mehdi Khouja, University of Gabes, Tunisia
Brian (Byung-Gyu) Kim, SunMoon University, South Korea
Jeongchang Kim, Korea Maritime and Ocean University (KMOU), South Korea

Copyright Information

Table of Contents

Towards a Formal Semantics for System Calls in terms of Information Flow

Laurent Georget Guillaume Piolle
Frédéric Tronel Valérie Viêt Triem Tong
EPC CIDRE SUPELEC/INRIA/CNRS/University of Rennes 1
Rennes, France
Email: *first_name.last_name*@supelec.fr

Mathieu Jaume
Team MoVe, UPMC LIP6
Paris, France
Email: mathieu.jaume@lip6.fr

***Abstract*—We propose a new semantics for system calls, which focuses on the information flows they generate in a UNIX OS. We built a prototypal model of an OS and system calls using the concurrent transaction logic along with its interpreter. We have yet a few results and applications that show the usefulness of our semantics to model an OS from a kernel point of view. Once completed, we expect our semantics to enable us to extensively test security software implemented inside the kernel, among other use cases.**

***Keywords*—Operating Systems; Security; System Calls; Information Flow.**

I. Introduction

In an operating system (OS), system calls define a clear boundary between the kernel-land, where lives the core part of the OS and the userland, which contains all the end-user applications. While the kernel runs with all privileges on the hardware, the user applications are granted only few rights to prevent them from interfering with each other. When such applications want to perform tasks requiring an access to the hardware or to communicate with each other, they need to ask the kernel for this service by the mean of *system calls*. Those are the only code interfaces between kernel-land and userland.

System calls are necessary because they enable the userland processes to achieve complex tasks. By allowing processes to communicate with each other and storing data, they let information flow in the system, which is necessary for all kinds of tasks. However, this can also be problematic because OS security mechanisms mostly rely on access control which can only prevent access to containers of information, and not to the information itself. Therefore, once a piece of information has left its original container, it is difficult to control the way it is accessed. One way to keep being able to know where each piece of information is in the system is to monitor the information flows using meta-information attached to each container of information called *taints*. Each time information flow from one container to another, the receiver gets *tainted* with the meta-information from the source. This way, each container is tainted accordingly to the origin of the data it contains. This monitoring can be performed at several levels of granularity depending on what is considered an elementary container of information. In our case, we are interested in OS-level containers, such as processes, files, sockets, etc. As the entire OS security relies on the correct interpretation of the information flows, information flows monitors are critical, and it is important to know how far they can be trusted.

Our main problem is that it is hard to know if information flows monitors actually interpret the flows correctly and if their view of the system is consistent with the actual state of the system. A formal proof through static analysis for example is infeasible if they are implemented on top of a preexisting OS which was not designed for proof. To tackle this issue, we need to know what are the information flows caused in an OS. Our work is based on the two following hypothesis. First of all, only processes cause information flows in a system because they are the active entities that execute codes whereas the other components are passive. Second, system calls are necessary for all information flows deliberately caused by processes. This leads us to define more precisely what the information flows generated by each system call are. Our goal is a formalized semantics of system calls in terms of information flows for a preexisting UNIX-like OS. This semantics will define clearly and unambiguously what information flows can be caused in the system and by which means. This would give us a reference to state on the correctness of a given information flow monitor. The rest of this paper is organized as follows. In Section I, we present the state of the art. In Section II, we discuss our working hypothesis and we present the main elements of our modelization. In Section III, a commented example of a system call gives the intuition of our syntax. Then, in Section IV, we present our work on a use case of our semantics: testing an existing information flow monitor. We conclude and give perspectives in Section VI.

II. Related Work

Information flow control systems is a long-studied topic in security. Some are implemented on top of preexisting OSes, which makes them more likely to be used in practical applications than *ad hoc* solutions. Blare [1] is built for the mainstream Linux kernel. Contrary to other tools such as Flume [2], it monitors automatically all the processes and does not rely on a user-land daemon to intercept system calls. Blare is based on the Linux Security Modules (LSM) framework [3]. LSM adds security fields in existing data structures inside the kernel and provides security developers with hooks. Those are placed before any access to a kernel object, which can represent a file for example. Functions can be set up on any hook so that when it is triggered the function is executed to mediate the access to the object. Blare plugs in functions in charge of calculating the propagation of taints on the kernel objects. The correctness of security softwares based on LSM relies on the correct placement of the hooks, otherwise, the access control is flawed. Several approaches using static analysis have been proposed so far to verify respectively that (1) all security-sensitive operations are mediated by a hook [4], and (2) to check that the necessary set of hooks is called before each security-sensitive operation on a kernel object [5]. Although those works are necessary to ensure that security mechanisms based on LSM are correct, they are not sufficient because they cannot be used to verify those mechanisms themselves. Our work aims at ensuring that information flows are correctly interpreted by security mechanisms, in order to give stronger guarantees on the validity of their model and the correctness of their implementation.

III. Modeling information flows in an OS

A. Discussion on our Working Hypothesis

Our working hypothesis is that no information flow can occur in an OS without at least the execution of one system call. Therefore, the kernel that receives the system calls and acts upon them is aware of all the information flows in the system. This hypothesis arises from the fact that system calls are the only way for a process to ask the kernel to run code on its behalf, and only code run by the kernel, that has high privileges, can make information flow between processes or from processes to other containers like files or network sockets. Of course, processes are free to manipulate the information they have inside their own memory space and to perform arbitrary manipulation on it. Hence, we have to consider containers of information at a coarser grain than individual memory locations. The file descriptor abstraction used under UNIX is useful here. On a first approximation, for our prototype, we consider as a container a process or anything that can be handled by a file descriptor in a UNIX abstraction. This covers files, network sockets, message queues, etc.

Our hypothesis has some drawbacks, too. For example, when two processes want to share a memory area, they need only a few system calls to set up and map the memory area and then, they can communicate and exchange information freely without any additional system call. So, to fully capture the information flows in this case, we have to overapproximate them and consider that the processes have exchanged *all* the information they had while the memory remained shared. A few malicious processes could then abuse this overapproximation by setting up shared memory areas and opening a lot of files to mask their real objective. Typically, a virus that would infect a lot of core processes could lower the accuracy of the information flow monitor so much that it would become practically ineffective. However, this would only result in valid flows being denied, not illegal flows being granted.

B. Scope and Object of our Modelization

To describe the information flows caused by each individual system call, we need to consider their environment. Information flows are caused by processes and occur between containers. The internal state of the OS also matters because the results of some system calls depend on specific conditions which are out of control from the process. For example, there is a limit on the total number of files that can be open simultaneously in an OS. Therefore, the same system call with the same parameters issued by a process asking for a given file to be opened can succeed or fail in two contexts indistinguishable from the process's point of view. This fact leads us to model the entire OS, with its internal state along with the system calls. Unlike many models focusing on processes like the process algebras which describe communication between them, we consider the kernel's point of view.

For our work, we first focused on the MINIX OS [6]. This OS is a fully-featured UNIX environment with a very clear and simple design. The methodology used to build our prototype can be extended to others kernels with a larger codebase such as Linux. In our model, the kernel is a database of containers of information. These containers may be of various types: files, network sockets, processes, memory segments, etc. For each of these containers, we maintain various pieces of information such as the size in case of a file, the program being executed in case of a process, etc. Separately, we have a list of observed information flows. We model system calls as transactions impacting the database asynchronously. Each time a system call is triggered, it is executed, objects in the database are created, altered or even deleted, and new information flows may be added to the list. This is where our semantics of system calls is necessary; without it, our model could not be proved accurate and we may miss important side effects in the internal state of the kernel (represented in our database), or even information flows. We also made our semantics and model executable. This is important because it allows us to build runnable test cases.

C. Model of the System Calls and the Database

As we previously said, system calls are modeled as transactions asynchronously reaching the database. Those transactions may alter the database in any possible way but they are executed as a whole, i.e., either the transaction is entirely executed or it is not at all. Furthermore, two system calls transactions cannot interfere with each other: their execution may be interleaved but not simultaneous and they cannot access the same objects if they are interleaved. These precautions are necessary to keep the database in a valid state, consistent with a real kernel. Of course, in a real multiprocesses system, two system calls may be executed at the same time but synchronization mechanisms, such as locks, prevent them from accessing the same objects and data structures in the kernel at the same time so our model remains valid if we make the assumption that synchronization is correctly enforced in the real system. If this were not the case, the real system would

TABLE I. FILES' META-ATTRIBUTES.

Name	Meaning
path	File system path
rd_locked	Read lock
wr_locked	Write lock
uid	Owner's id
gid	Owning group's id
mode	Access rights and flags `setuid`, `setgid`, `sticky bit`

be buggy and its internal state inconsistent, so no model would be accurate.

To model the system calls, we used the Concurrent TRansaction logic, CTR, by Bonner and Kifer [7]. This logic provides us with both a syntax and a semantics to express changes and consultations in a database as transactions. The database, in the case of the CTR, can have an arbitrary structure, provided that it can be accessed by the means of clearly defined logical primitives. In our case, those primitives are the consultation of the database, the atomic change of an attribute of a single object in the database, the creation of a new object in the database, the deletion of an object, and the registration of a newly detected information flow. As its name suggests, the CTR lets two or more transactions run concurrently, in an interleaved manner, which is desired in our case. Last but not least, the CTR provides us with an executional semantics: an interpreter can be built to simulate the execution of a transaction in a database. The documentation of our interpreter is available online [8].

Our database model contains four tables and a list. The tables are (1) the table of active processes in the system, (2) the table of existing files in the system, (3) the table of open file descriptors, and (4) the table of memory areas allocated to processes. The list contains the information flows the interpreter detects. It is chronologically sorted. This is of course a small model built only for the sake of prototyping because we consider only processes and files as containers of information but it can be extended. Each table contain a set of entries. An entry is indexed by a unique identifier and represents an object in the OS. For example, an entry in the processes table is a living process in the system. Each entry is described through several attributes. Tables I and II describe respectively the fields contained in the entries of the files table and the file descriptors tables. As one can see, we try to mimic as closely as possible the real OS and we replicate the semantics of `open` and `close` because obviously, information flows are not identical for open and closed files. This is why we made the distinction between files and file descriptors, which correspond to files opened by some process.

IV. EXAMPLE OF A SYSTEM CALL: `READ`

System calls are written as a set of logical clauses. Each clause is a transaction which corresponds to the system call. The clauses are mutually exclusive, only one of them will execute. They correspond to the different behaviors of the system call: there is a clause for each error case and each valid case. The way the interpreter chooses the clause to execute is simple: it tries them one after the other, until one can be entirely executed. Each clause begins with the header of `read` with some arguments. Arguments can be preceded by a $+$ sign or a $-$ sign. A $+$ sign means that the argument is an input, a $-$ sign that it is an output. A system call usually has a return value but can also return values through a pointer passed as an argument. Our notation captures this.

TABLE II. FILES DESCRIPTORS' META-ATTRIBUTES.

Name	Meaning
file	File corresponding to the file descriptor
procs	Processes owning this file descriptor
opts	Read-only, Write-only, Read-Write, Append, etc.
pos	Current position in the file

$$\texttt{read}(\text{+FileDescriptor, +Buffer, +Size, -Return}) \leftarrow \quad (1)$$
$$\neg(\texttt{GET\_FID\_FOR\_PROC}(caller, FileDescriptor, Fid)) \quad (2)$$
$$\otimes\, Return = \texttt{EBADF}. \quad (3)$$

This first clause describes the execution of `read` with an invalid file descriptor as first argument. The clause's body starts with a query called `GET_FID_FOR_PROC` which returns, for a given process and a given file descriptor, the file referenced by the file descriptor for the process. This is a logical predicate having logical value `true` if the file descriptor is valid, in which case Fid is a reference to the file and `false` if the file descriptor is invalid. In the first clause, the file descriptor is invalid. What comes after is an addition of the CTR to the predicate calculus. The operator $\otimes$ is the *sequential conjunction*. The intuition of this operator is that if both A and B are transactions, then $A \otimes B$ is the transaction whose execution consists of the correct execution of A *followed by* the correct execution of B. If A has not logical value *true*, which means it cannot be executed, then B is not executed (its effects on the database, if any, are ignored) and the whole transaction $A \otimes B$ is *false* (which can be thought of as *aborted*). If A is true, B is executed and if B is false, the transaction is false too. Here, we test the negation of `GET_FID_FOR_PROC` so if the file descriptor is invalid, the previously undetermined value *Return* is now known to be equal to the constant `EBADF`. If the file descriptor is valid, the clause is rejected and another one is tried. When a clause is rejected, all its effects on the database are *rollbacked*.

$$\texttt{read}(\text{+FileDescriptor, +Buffer, +Size, -Return}) \leftarrow \quad (4)$$
$$\texttt{GET\_FID\_FOR\_PROC}(caller, FileDescriptor, Fid) \quad (5)$$
$$\otimes\, Fid.opts \not\ni \texttt{READ} \quad (6)$$
$$\otimes\, Return = \texttt{EIO}. \quad (7)$$

In the second clause, we first check that the file descriptor is valid (5), then if it is, we check if the file was opened in reading mode (6) and if it is not, we conclude with the returned value of constant `EIO` (7) (Input/Output Error). If it is, the transaction aborts. Here, we see how the transaction representing the system call refers to the content of the database representing the OS's objects. Fid refers to an entry in the file descriptors table, and $Fid.opts$ refers to the value of the attribute $opts$ inside this entry.

$$\texttt{read}(\text{+FileDescriptor, +Buffer, +Size, -Return}) \leftarrow \quad (8)$$
$$\texttt{GET\_FID\_FOR\_PROC}(caller, FileDescriptor, Fid) \quad (9)$$
$$\otimes\, Fid.opts \ni \texttt{READ} \quad (10)$$
$$\otimes\, Size > 0 \quad (11)$$
$$\otimes\, ReqSize\ is\ min(Size, Fid.file.size - Fid.pos) \quad (12)$$
$$\otimes\, Buffer \ll Fid.file \quad (13)$$
$$\otimes\, Fid.pos := Fid.pos + ReqSize \quad (14)$$
$$\otimes\, Return = ReqSize. \quad (15)$$

Finally, the third clause describes the execution of `read` that actually ends up with an information flow. In this clause, after the first check identical to the previous case (9), the file open mode is tested (10). The size to read is checked to be non-null (11). Then, a new name *ReqSize* is created and immediately given the minimum value between the requested size and the number of bytes left in the file (12). The special syntax that follows means that an information flow took place from the file to *Buffer*, which represents the memory zone where the content of the file is read by the process (13). Next, another syntax element is introduced (14). The current position in the file (which is one of the numerous elements of the database) is *updated* and becomes equal to the old position plus the number of bytes read. Finally, the returned value is the number of bytes read (15).

V. Usage of the Semantics

The semantics for system calls is executable. Sleghel et al. built in 2000 an interpreter for the CTR [9]. Of course, their work was not directly usable for us because they used a very generic model of relational database. In our very specific case, we needed a special database model to account for the particular nature of an OS's kernel. Fortunately, the CTR is not bound to any specific database but can be used with any model as long as elementary operations to query and update the database are provided. So, we built our interpreter basing our implementation on Sleghel et al.'s work for the inference engine and implementing our own elementary predicates. The inference engine, which implements the inference rules of CTR, is written in SWI Prolog [10] and the database part in C++. The latter lets us implement all kinds of side effects in the database and instrument the interpreter. Sleghel et al.'s interpreter was built for another Prolog flavor, but we chose to port it to SWI Prolog because of its handy C++ interface.

The motivation behind the construction of our semantics was to test Blare [1], an information flow monitor. While this is still an on-going work we already have encouraging results on our semantics and interpreter. We are able to set the model of the OS, i.e., the database, in a given initial configuration. Then, we can run simple test cases such as two concurrent processes trying to atomically write in the same file at the same time. One of the processes output gets overriden by the other and the result depends on the order of execution of the writings. Our interpreter lets us effectively see that fact and we can see the content of the database in each situation. This is very interesting because, using the same test case once we get a full formal semantics for Linux system calls, if we reproduce the same situation on an OS equipped with Blare, we should see the same result. If Blare tells us that the file contains information from both processes or information from none of the processes, we can tell it is wrong. Of course, this is a simple example and there are many cases much more complex to deal with.

VI. Conclusion and Future Work

We presented a formal semantics for MINIX system calls for and a methodology to build similar ones for any UNIX-like OS. To the best of our knowledge, this is the first time a semantics is proposed for system calls in preexisting OSes. Our work will be useful to get more confidence in the correct functionning of those information flow monitors. The semantics may also have other uses in the future, such as proofs of correctness for the implementation of system calls. Research work on information flow control is far from being over and more work is needed to achieve a control as accurate as possible, and thus, to bring more security to end-users of OSes. We follow on this track to improve existing solutions and make both OSes and security mechanisms more trustworthy.

References

[1] L. George, V. Viêt Triem Tong, and L. Mé, "Blare tools: a policy-based intrusion detection system automatically set by the security policy," Recent Advances in Intrusion Detection. Saint-Malo, France: Springer Berlin Heidelberg, Sep. 2009, vol. 5758, pp. 355–356.

[2] M. Krohn *et al.*, "Information flow control for standard OS abstractions," ACM Symposium on Operating Systems Principles. Stevenson, WA, USA: ACM, 2007, pp. 321–334.

[3] C. Wright, C. Cowan, S. Smalley, J. Morris, and G. Kroah-Hartman, "Linux security modules: general security support for the linux kernel," USENIX Security Symposium. San Francisco, CA, USA: USENIX Association, 2002, pp. 17–31.

[4] X. Zhang, A. Edwards, and T. Jaeger, "Using CQUAL for static analysis of authorization hook placement," USENIX Security Symposium. San Francisco, CA, USA: USENIX Association, Aug. 2002, pp. 33–48.

[5] T. Jaeger, A. Edwards, and X. Zhang, "Consistency analysis of authorization hook placement in the linux security modules framework," ACM Transactions on Information and System Security, vol. 7, no. 2, May 2004, pp. 175–205.

[6] A. Tanenbaum and A. S. Woodhull, Operating Systems: design and implementation, 3rd ed. Upper Saddle River: Prentice Hall, 2009.

[7] A. J. Bonner and M. Kifer, "Concurrency and communication in transaction logic," Joint Intl. Conference and Symposium on Logic Programming. MIT Press, 1996, pp. 142–156.

[8] L. Georget, "ALFRED, an interpreter for the semantics of system calls," 2014, [Retrieved: 2015.03.02]. [Online]. http://www.lgeorget.eu/alfred/

[9] A. F. Sleghel, "An optimizing interpreter for concurrent transaction logic," Ph.D. dissertation, University of Toronto, 2000.

[10] J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager, "SWI-prolog," Theory and Practice of Logic Programming, vol. 12, no. 1-2, Jan. 2012, pp. 67–96.

Secure Communication in a Heterogeneous Sensor System

Ondřej Čožík, Jaroslav Kadlec, Radek Kuchta
Department of microelectronics
Faculty of Electrical Engineering and Communication, Brno University of Technology
Brno, Czech Republic
e-mail: xcozik00@stud.feec.vutbr.cz, kadlecja@feec.vutbr.cz, kuchtar@feec.vutbr.cz

***Abstract*—The article deals with security of communication in a sensor system combining a wireless RF network and Ethernet. The article deals with design of complete secure sensor system combining a wireless RF network and Ethernet. Within the suitable sensor system topology design the requirements for the fast and efficient data exchange between individual logical system layers are taken into account. One of the major requirements for our design was to develop solution with minimum computing power consumed by the communication sensor nodes. The particular system layer security is based on the known security methods and protocols (TLS protocol, improved Diffie-Hellman protocol GDH.3), which have been extended by the methods needed for their practical use (method called AVOM, which is intended for discovering and labeling of all RF network devices). A partial goal of the designed solution is to improve the robustness of the implemented security mechanism for wireless logical group security key establishment.**

Keywords-Sensor system; RF network; TLS protocol; Diffie-Hellman protocol; GDH.3 protocol.

I. INTRODUCTION

The article briefly describes the development of a sensor system including the selection of used components and also the encryption principles used in communication between devices. Wireless system security must be designed in such a way that from the beginning it is developed so as not to allow an attacker to retrieve any information from the system [1][2].

Within the research the various sensor system security methods have been observed [9][10][12][13][14][15]. Unfortunately, these methods did not meet some of our basic requirements on the developing sensor system, i.e. rate of the data exchange, computing power, synchronization, simple implementation of the secure algorithm into the microcontroller, possibility to immediately decrypt every received secured messages, and the ability to remove / add a new member to the secured sensor network.

In the survey of available methods intended to protect common data exchange in the sensor system, there has been found several security techniques, which did not meet the important demands [14][15], or meet the demands just partially[14][15]. An overview and comparison between different security methods including their features can be found in [18]. Since this security systems are not appropriate, the logical step was to create a security system focused on the implementation simplicity, the rapid encryption key determination for all devices in the sensor network, device access to the system management capability and minimum traffic load necessary for additional information transmitting. The proposed system will have two modes – the initialization mode and the normal mode. During the initialization mode, the encryption key will be provided by below mentioned mechanisms. Subsequently, when the normal mode become active, the messages are encrypted using the agreed key and can be sent into the sensor system.

The encryption key distribution within the higher hierarchical system layer is based on the TLS protocol [2], extended Diffie-Hellman protocol GDH.3[3] which is the crucial part of the encryption key distribution in the wireless RF network.

From the outset, demands on the topology of the sensor system corresponding to their planned use are mentioned. Also, the main requirements for particular devices in the system are specified. For both the selected topology and each component the pros and cons are outlined.

The requirements for simplicity and speed of encryption/decryption are especially important, because every algorithm will be implemented into the microcontroller. In this part of the article, the provision of a secret key and the onward transmission of the encryption key from the top system layer down to the lowest layer is illustrated. Following this is the assessment of the designed solution in terms of communication security and the time demands for the encryption / decryption process.

The final section of the article deals with the selection of a safe and relatively simple encryption method for communication between devices in a wireless RF network and also at the Ethernet level.

II. REQUIRED SYSTEM TOPOLOGY

The proposed topology of the complete system is adapted for rapid message exchange between master and slave devices. During the exchange, relatively large data flows between certain devices may be included (messages size is up to tens of bytes in RF network). Furthermore, emphasis was placed on the possibility of an accurate synchronization between the wireless modules and control device.

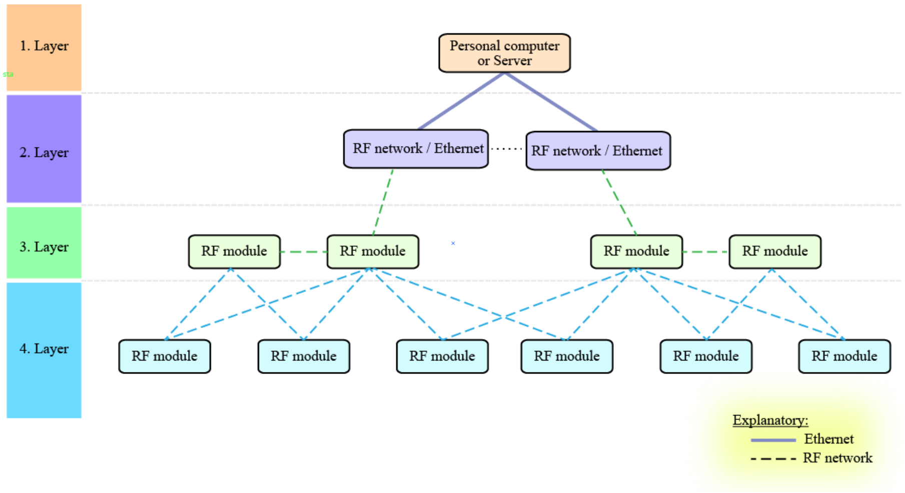


Figure 1. Proposed sensor system hierarchy

The resulting topology is presented in Figure 1 and it can be seen, that the system topology is divided into the four layers. The control device, which could be, e.g., server or personal computer with appropriate control software installed, is in the top (first) system layer and it processes packets over Ethernet. The second system layer of created topology contains a converter placed between the Ethernet and wireless RF network. In this layer, there could be more than one converter, but only one converter for one logical group of the RF modules. The main task for them is to forward data messages from the wireless network to the Ethernet and vice versa. The converters should be positioned correctly to avoid overloading any RF module at the lower level. A suitable compromise between the number of transmitters (converters) and wireless modules in the group, which can use it, must be found (it depends on node message size and message exchange frequency between the nodes and server).

The third layer represents the RF master modules, which are fixed in the area, serving as a messages repeater to a desired device and back.

Finally, in the lowest (fourth) layer there are portable wireless RF slave modules periodically sending data and status information to the parent device in the hierarchy (sensors). Therefore, it is desired to establish secure communication in order to avoid a leakage of sensitive information transmitted in the system or misuse of invalid data by a possible attacker that could cause an error in the sensor system, or even cause it to malfunction.

Portable RF (slave) modules transmit data to the third layer, that takes care of transferring messages to the RF module, which is able to directly communicate with the converter between the RF network / Ethernet in the third layer.

The messaging system for one RF slave module in the fourth layer is shown in Figure 2. One of the requirements for the proper function of the system is RF modules in the fourth layer should always be available for at least two RF master modules.

Since the RF slave modules are portable, there is a complication with security options due to the possibility that a module may be in the area of the first RF master module at one moment, but in the next moment it could be in another RF master module area. Because of this feature, the system must be secured in a manner allowing all devices in the third layer to decrypt the message from every device in the fourth system layer. A similar situation occurs between the second and the third layer in the hierarchy of the sensor system, the only difference is, that the modules in those layers will be moved very rarely.

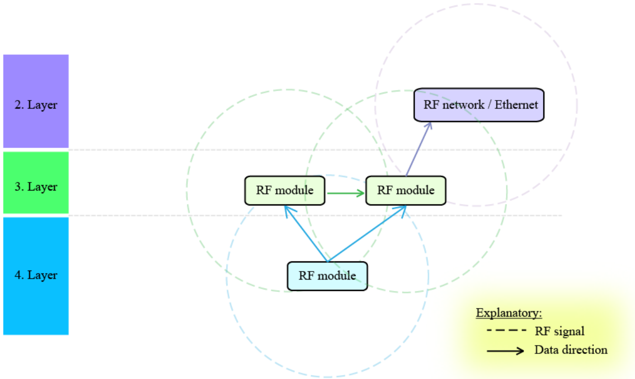


Figure 2. Message forwarding at lower sensor system layers in the RF network

A method of message forwarding between devices in the third layer is shown in Figure 2. Due to sensor system price reduction it is necessary to have the lowest number of converters in a sensor network. To make a successful transmission from a desired RF master module to the converter device, the message has to be sent from the sender towards the converter. If the converter is not in the sender's area, the message will be forwarded towards it (in the third layer) until the converter receives the sender's message.

III. SECURE COMMUNICATION ON AN ETHERNET NETWORK

Transport layer security (TLS) protocol for the security of communication in the highest level in the sensor system hierarchy will be used [3][4]. TLS protocol is a free version of SSL protocol. Using the mentioned protocol, it is possible to create encrypted communication channel between the converters and the server via Ethernet.

A. Communication establishment using the TLS protocol

A communication establishment has to be performed before the two parties are able to transfer data via a secured channel using the TLS protocol. An identity and other information has to be exchanged between server and client to create the encrypted channel and then secure communication can begin.

The connection establishment (handshake) includes a total of four consecutive phases [1]. The description of each handshake stage is not included in the article, because the TLS protocol in general is well known and used.

Once the handshake process between the client (converter from RF network to the Ethernet) and the server has been completed, the data exchange between two devices, that are authenticated and have the necessary keys, can be initialized.

B. A Data exchange via TLS protocol

After a successful TLS handshake protocol between the transmitter and server, both sides are able to encrypt communication using the agreed key. The sending procedure for application data via secure communication channel is shown in Figure 3.

In the first step, the user's secret data are taken from the application layer and divided into blocks with a maximum size of 2^{14} B (according to the TLS protocol specification). In the second step, a lossless compression function can be applied to the separate data blocks from the previous step. The compress function between both sides was arranged in the handshake protocol, in the current TLS protocol version there is no compression method by default.

In the next step, a message authentication code is added to the encrypted and compressed data block. The code is determined by the HMAC technique [1].

In the fourth step, the compressed segments with the authentication code are encrypted using the selected algorithm. The AES encryption algorithm [6] will be used in the proposed system due to availability of an AES hardware encryption module in the Texas Instrument microcontroller named CC430F5137, which is contained in all designed RF devices. The encryption key length in these devices could be 128 b (it is sufficient length for this kind of sensor network).

In the last step, a 5 B header to the encrypted data block is added. The first byte of the header represents the protocol version used for attached data processing. The next two bytes contain the major and the minor version of the used protocol and the last 2 bytes carry the entire encryption segment length.

The communication between all other devices on the second layer of sensor system topology and server is established in the same way. When all devices on the second layer are securely connected to the server, it is necessary to create an encrypted connection also on the lower system layers. This issue is the subject of the next section IV: Secure communication in the wireless section of the sensor system.

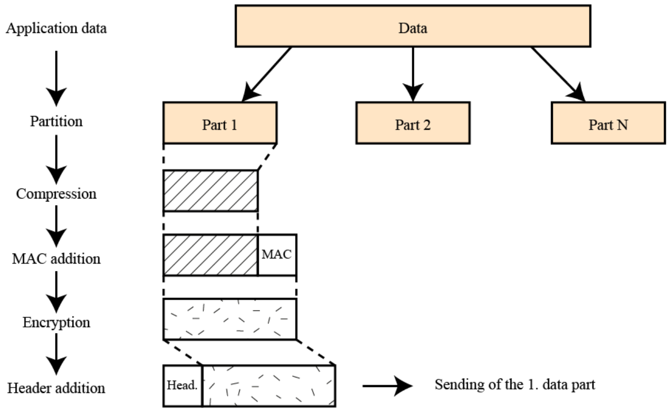


Figure 3. Application data encryption in the TLS protocol [1]

IV. SECURE COMMUNICATION IN THE WIRELESS SECTION OF THE SENSOR SYSTEM

In this section, the security of communication between the first and second layer is not considered, this has been described in the section 3. Only secure communication from the second system layer below is taken into account, i.e., RF network security.

For the actual communication design between the devices it is important to analyze all potential attacks and feasible security risks for this type of system and the most suitable security method should be chosen [7][8][9].

A. Potential risk and related problems

In wireless networks generally, there are many ways of how to attack system security [1][3]. One of them may be listening to network communication. In the case of an unsecure network, an attacker can read all transmitted messages. The solution, which removes this problem, may be encryption of all messages in the designed RF network. Another difficulty could then emerge – how to manage the encryption keys for all wireless devices?

Following this we must assume that the wireless RF system is already secured. Although the attacker can intercept the transmitted cipher, without the used encryption key the cypher is then irrelevant. Data collected in the sensor system are from a large number of RF measuring modules. They send this data to the parent layer in the network hierarchy. Occasionally the measured value changes very slowly (or not at all) and the module will send the same value over and over. An attacker can take advantage of this information and break the encryption. To prevent this sensor system feature, the *COUNTER* field will be added to all messages. When the RF module sends the measured value, it immediately increments the *COUNTER* field (1).

$$COUNTER_{(t+1)} = COUNTER_{(t)} + 1. \quad (1)$$

The advantage of a block cypher is that a change of a single bit at the encryptor input causes a large change at its output. Basically, it is possible to achieve a completely different cypher even with two identical data frames sent via an RF module at the lowest level, only with a varying *COUNTER* field.

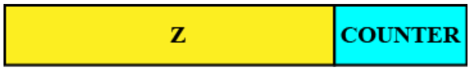


Figure 4. Message Z in RF network with a COUNTER field

The situation of the adding a *COUNTER* field to the message is shown in Figure 4. The field *COUNTER* will be added to the permanent data field Z . This precaution should also reduce the possibility of re-sending a previously intercepted message from the attacker. This is because the transmitter and receiver know the current *COUNTER* parameter value and receive the message only if the parameter from the message is identical to its *COUNTER* value. If any attacker captures a packet and subsequently sends it, the receiving party will assess the message as invalid, because the same parameter has already been received and the *COUNTER* value is different.

To prevent other types of attacks such as a brute force attack, or an attempt to capture as many messages as possible in order to determine the encryption key, the proposed security feature allows the encryption key change in a secure way. All microcontrollers used in all devices on all layers, except the highest one, contain AES module with the option of a 128 b key [4][5]. It is appropriate to use this module for standard encryption. Presently, the main problem is how to securely set up the key in all devices.

There is a simple way to solve this problem, if the communication is only between two sides, as shown in Figure 5.

Figure 5. Two parts communication (Device A and B)

In this case, it would be sufficient to use the Diffie-Hellman (DH) key establishment protocol [6]. The protocol deals with an ideal two sided communication. This situation does not appear in the proposed sensor system and it is necessary to securely establish the encryption key for multiple devices [7][8][9]. One of the suitable protocols, which can be used in the sensor system, is named GDH.3 [10].

B. *The algorithm for automatic detection and labeling of modules in an RF network*

In order to use the GDH.3 protocol in the system, it is necessary to specify a logical group of modules that have the same secret key used for device identification. When all wireless RF devices prove their identity, the parent device will determine a new encryption key for the whole group. The encryption key will be sent using the previous established secret group key. For the execution of the GDH.3 protocol, each module in the logical group must have a unique number (address).

The unique number inside the group must be assigned automatically due to adding a new wireless module and avoiding collision with another previously labeled device. For this purpose a method of automatic detection and labeling of the wireless device (AVOM) was designed.

For correct search functionality it is necessary to send a token in the RF network. The device assigns the unique numbers (addresses) to the new identified modules in its communication area. Along with the token, the highest assigned address will also be transmitted. Due to this mechanism, the next device knows exactly the following address, which can be assigned to the new devices and there will be no address collision in the RF network.

The resulting AVOM method, which was used in the system hierarchy, is shown in Figure 6. It should be noted, that before the start of this function, all devices have to know about the new sequence of searching and labeling devices. This notification in the RF network can be done using a broadcast message, i.e., message delivered to all devices in the network. On the basis of that message, all devices delete their current addresses, related information and stop all further communication until the system is completely secured (it can be sent using a broadcast message again).

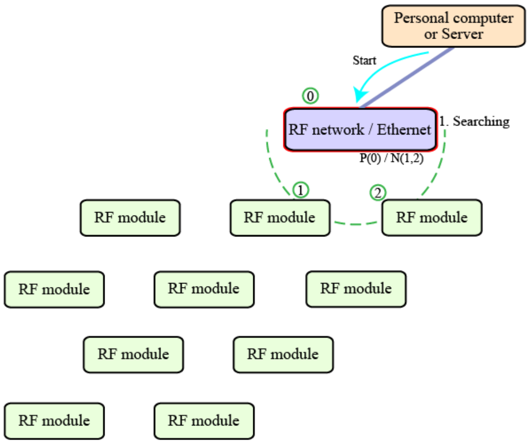


Figure 6. Basic system hierarchy for an address determining - includes a first scanning step in the RF network

In Figure 6, we can see the typical hierarchy of the proposed sensor system. In this illustration is also shown the first step of the searching method. Firstly, the server sends the AVOM start command to the RF / Ethernet Converter in the second layer, which is on the highest layer from all the RF devices. Therefore, the RF / Ethernet converter selects the address 0 (in Figure 6 is the assigned address above the module in a green circle). Secondly, the scanning of all available devices in the converter area is launched. When the converter gets all directly available devices via RF communication, the converter individually assigns to these devices their addresses according to a chosen criteria (e.g.

signal strength, response time, etc.). In Figure 6 addresses 1 and 2 are assigned.

Each device will always save the parent device address (address of the device sending the token) and all unlabeled devices in its area to which the token has not been forwarded yet. In the figures these numbers are always written below the module. For example, the RF / Ethernet converter in Figure 6 has the previous address device equals $P(0)$, this means there is not a device above it in the system hierarchy. The algorithm is able to recognize, that all devices have been labeled. Numbers $N(1,2)$ mean, that the token has not been sent to the device with address 1 and 2 yet.

The module highlighted in red (in the figures) currently has the token and is allowed to label the modules in its RF communication range; it is indicated by the green dashed line.

Thirdly, the token is passed to the next device in the network (with address 1). The token is always sent to the device with the lowest newly assigned address. When a selected device receives the token, it saves the address of the previous device (0), then scans its communication area for new unlabeled devices and assigns them appropriate addresses, i.e., 3, 4 and 5 in Figure 7. The module holding the token saves in its memory all newly labeled devices to send them the token in the future. The device with the lowest address (3) is chosen and the token is forwarded to this device.

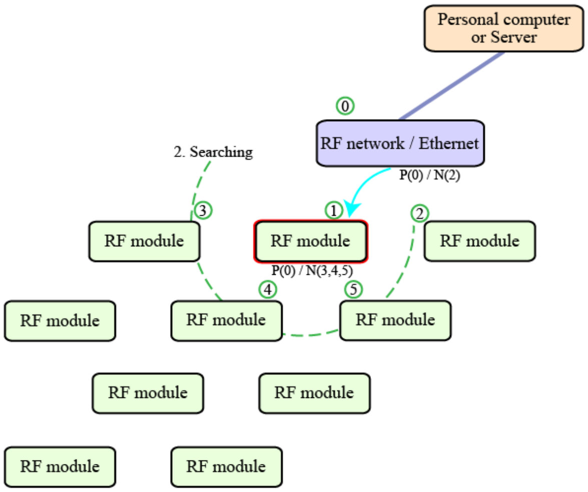


Figure 7. Second step of the AVOM function

The same procedure is applied until the algorithm arrives at the device in its communication area where there is no an unlabeled module. This scenario is shown in Figure 8, the token holds the device with address 7 and in its range there are only labeled devices with addresses 8, 9 and 10, which are final. The device with address 7 subsequently forwards the token to the final devices 8, 9 and 10. Each of the final devices checks its communication area, but there is no new device, so the token is sent back to the device with address 7 and so on. The forwarding process is shown in the figures by arrow. The numbers near the arrows represent the individual steps of token forwarding. When the device 7 verifies all 3 devices in its area, there is no device to forward the token, so it sends the token back to the parent device with address 6. The device with address 6 also does not have any device in its area and sends the token back until the token arrives back at the device with address 1.

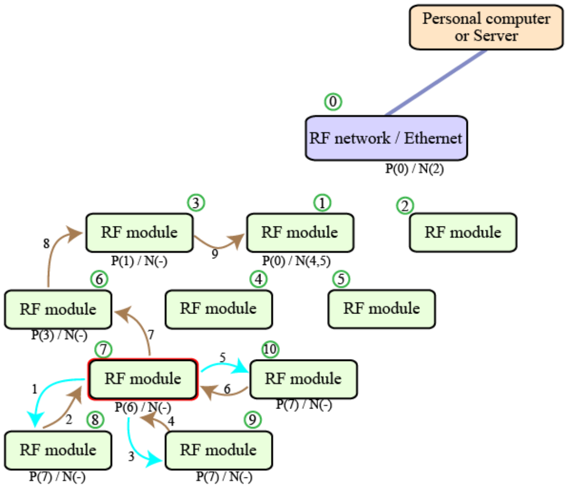


Figure 8. The end of the forward phase of the AVOM algorithm

The device 1 has 2 modules stored, which have not received the token, so it sends it immediately to the module with a lower address, i.e., 4. When a new device is not found within its communication range the token is sent back to the device with address 1. It stores the last device, which still has not received the token. The same procedure is undertaken with the device with address 5. This module also has an empty queue and returns the token to the device with the address 0 (RF / Ethernet converter). This situation is shown in Figure 9.

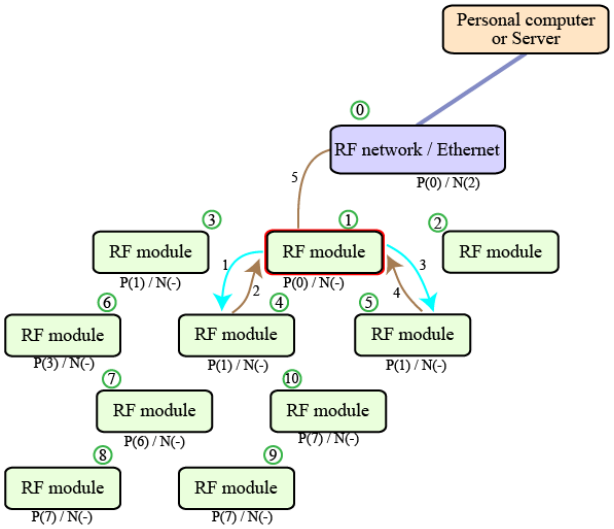


Figure 9. Token forwarding during the AVOM function

The module with address 0 (converter) still has one device in its queue, which has not received the token yet, specifically it is the RF module with address 2. When the token exchange is complete, the new address is assigned to all identified devices in the RF network. Since the address of the previous module is identical to the converter, the converter sends a message to the server with a device numbering completion announcement in the RF network (see in Figure 10).

When the labeling process required for the GDH.3 protocol startup is completed, a temporary secret key will be established in the labeled logical group for precise device identification and subsequently encryption key transmission.

C. *A group key arrangement using the GDH.3 protocol*

In the basic Diffie – Hellman protocol the key is placed between the member M_1 and M_2 using several steps. In the first step, member M_1 sends the value u according to (2) to member M_2 [1][6].

$$u = \alpha^{N_1} \left(\mathrm{mod}\ p \right). \tag{2}$$

In (2), there is a group generator α, a random number N_1 is generated by M_1 and p is a prime number. The values p and α are known to both sides.

Once the member M_1 sends the value u, the member M_2 sends another value v to the member M_1. A computing v value describes (3).

$$v = \alpha^{N_2} \left(\mathrm{mod}\ p \right). \tag{3}$$

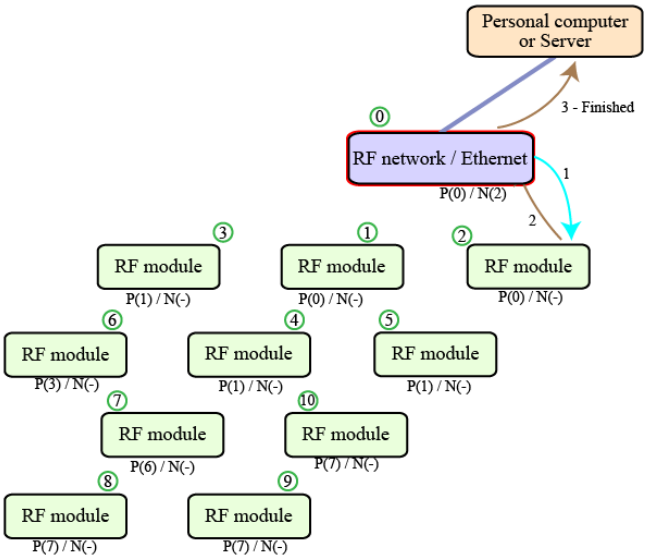


Figure 10. Final phase of the AVOM function

Equation (3) is similar to (2), except there is a different coefficient denoted as N_2. When both sides exchange their values, they are able to determine the same secret key K. The first member M_1 will use (4) and the second member will use (5). Now, both of them have the same secret key K.

$$K = v^{N_1} = \left(\alpha^{N_2} \right)^{N_1} \left(\mathrm{mod}\ p \right). \tag{4}$$

$$K = u^{N_2} = \left(\alpha^{N_1} \right)^{N_2} \left(\mathrm{mod}\ p \right). \tag{5}$$

To establish a group key the extended Diffie – Hellman protocol (generally for n devices) has to be used [3].

During the initialization process of the entire protocol, which allows determination of the shared key it is necessary to assign the address to all devices in the group. The devices without the address will not be able to determine the new key (for more information, see section IV.B The algorithm for automatic detection and labeling of modules in an RF network). As in the basic DH protocol, all devices know the public parameters denoted p and α. In addition, each group device M_i must generate its own random exponent N_i. Because of simplicity, the operation $\mathrm{mod}\ p$ will not be shown in the next equations.

In the first stage of the protocol, the first module M_0 will generate a value α^{N_0} using its secret exponent N_0 and sends the value to the module with the following address (device M_1). The device M_1 computes the value $\alpha^{(N_0 \cdot N_1)}$ and transmits the new value to the next device M_2. The procedure is repeated until the transmitted value reaches the device M_{n-2}, where n is total number of modules in the group. The value is then also transferred to the device M_{n-1}, which calculates the last value, but does not send it to the last device M_n. Generally, it is possible to determine the computed value u_k in the device M_k by (6) [3].

$$u_k = \alpha^{\prod_{k=0}^{i} N_k}. \tag{6}$$

In the following (second) stage of the protocol GDH.3, the broadcast message with the value computed by the module M_{n-1} is sent. All modules, including the module M_n, receive the message containing a value u_{n-1} specified by (7) [3]. The last module M_n has to save this value due to potential extension of the group.

$$u_{n-1} = \alpha^{\prod_{k=0}^{n-1} N_k}. \tag{7}$$

During the third stage, each device M_i receives the broadcast message with the value (7), and subsequently excludes its own random exponent N_i by extraction of the root (7) by inverse value of the exponent N_i^{-1} and the result is sent to module M_n. Generally, the sent value u_i from the module M_i can be described by (8) [3].

$$v_i = \alpha^{\prod_{k=0}^{n-1} N_k} \mid k \neq i \,. \tag{8}$$

In the fourth stage, module M_n must save all received values, raise them by its random exponent N_n and send them using the broadcast message again. An individual message contains the value s_i done by (9) [3].

$$s_i = \alpha^{\prod_{k=0}^{n} N_k} \mid k \neq i \wedge i \in [1, n-1] \,. \tag{9}$$

Each module M_i obtains the value s_i in this stage. When the module M_i uses again the random exponent N_i on the received value s_i, it computes the secret group key K, which will be used for exact module identification and subsequently for distribution of the communication encryption key.

$$K = \alpha^{\prod_{k=0}^{n} N_k} \,. \tag{10}$$

The value of secret group key K , which was established using the GDH.3 protocol, can be determined by (10) [3].

D. Adding a member to the group with a secret key

Over the sensor system`s lifetime, there could be a requirement for a system expansion by adding a new module into the existing group with the secret key. All the devices communicate using the established encryption key, but the values for computing the secret group key are stored within it. The new group key, which will also be used for the new member, is based on the stored values.

The group member M_n with the last address must store the values from the second and the third stage of the group key establishment. Initially, the last member M_n will generate a new random exponent $\overline{N_n}$, which raises the second stage stored value by the new exponent $\overline{N_n}$ and obtains a value defined by (11) [3].

$$\alpha^{\prod_{k=0}^{n} N_k} = \alpha^{N_0 * \ldots * N_{n-1} * \overline{N_n}} \,. \tag{11}$$

The module M_n sends the value (11) to the new device M_{n+1}, which generates its own exponent N_{n+1} and computes a new secrete key K_{n+1} for the whole group (12) [3].

$$K_{n+1} = \alpha^{\prod_{k=0}^{n+1} N_k} = \alpha^{N_0 * \ldots * \overline{N_n} * N_{n+1}} \,. \tag{12}$$

The final stage of adding a member is that the device M_{n+1} calculates n new values obtained from the device M_n. Into these values it has to add a generated exponent N_{n+1} and send them out by broadcast messages to allows other modules to determine the new group key K_{n+1}. Basically, the third and the fourth GDH.3 protocol stage is executed once more.

$$\alpha^{\prod_{k=0}^{n} N_k} \mid k \neq j \wedge j \in [1, n] \,. \tag{13}$$

The module M_{n+1} sends the value defined by (13) to the rest of the devices. One exponent N_j is missing in each of the sent values so it can be completed only by the device M_j.

E. Removing a group member

Due to security reasons, the algorithm has to have the ability to remove a particular device from the group. The device M_n is important for removing the device M_p (where $p \in [1, n-1]$), because all values are saved in it from the fourth stage of group key K establishment. The module M_n must generate a new random exponent $\overline{N_n}$, which will be used for the calculation of the $n-2$ values according to (9).

$$K = \alpha^{N_0 * \ldots * N_{p-1} * N_{p+1} * \ldots * N_{n-1} * \overline{N_n}} \,. \tag{14}$$

A new value for the device M_p is omitted so it is not possible to determine the new secret key K for this module in the future (14) [3].

The removal of M_n, device M_{n-1} takes over the role of the last module with the designation M_n. It also stores all the data from the fourth phase of the key establishment. Firstly, exponent N_{n-1} is cleared of stored messages (12) and it generates a new exponent N_n, and uses it to calculate new values (12), which sends the results to all devices in the group. Since the random coefficient N_n and N_i (according to recipient M_i) are missing in all messages, therefore, the last module M_n is not able to determine the resulting communication key K.

F. Portable RF modules security at the lowest system layer

The lowest layer (fourth) in the hierarchy of the sensor system contains only the portable RF modules communicating with RF devices at the higher layers. From the requirements on the sensor system it follows, that each portable RF module has to be reachable from more than one higher layer RF device. It is obvious, that communication with more than one encryption key would be too difficult due to key management.

The final encryption key will be transferred to the portable device using DH protocol between the module and the server after a successful authentication of both parties. The authentication will take place immediately when the device sends the access request to the network. The portable device creates a stipulated request and encrypts it using devices private key SK_m. The created cypher M_s device encrypts once more with the server's public key VK_s, which is available to all RF modules.

The outcome is that the encrypted message M is created and is sent through the transmission channel to the server. The entire encryption process of the requirement is captured in Figure 11.

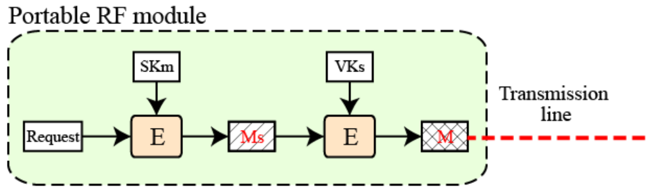


Figure 11. Request encryption in the portable RF module for sensor network access

The server receives a message M and applies its private key SK_s and gets the message M_s. The server stores all the ID's of all portable RF modules, which are allowed to access the RF network. These records can be edited by an authorized person with access to the server databases only. In the database, along with the ID the public keys VK_m are saved.

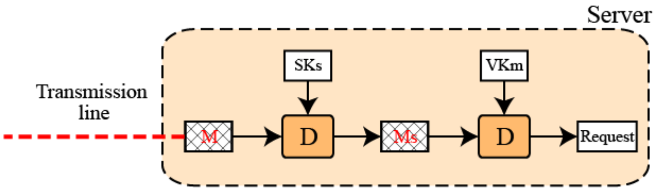


Figure 12. Message request for sensor network access and decryption in the server

The server is able to apply the proper public key associated with the module requiring access to the network. Thus, the server can decrypt the original message (see Figure 12).

When the server receives a valid request, it generates a random value for the key establishment according to the DH protocol between two devices (2), and encrypts the result in reverse order (firstly it uses its private key SK_s and following this encryption by the portable RF module public key VK_m). The message with an encrypted random value is forwarded back via the transmission channel to the RF module.

The message in the RF module is sequentially decrypted using the private key SK_m and the public key VK_s. The device generates another random number. It describes (3). The number is appropriately encrypted (as in the first request) and sent to the server (see Figure 11). In this way, the possibility of an attack by the man in the middle is excluded. This type of attack could occur only by sending unencrypted values for the DH protocol.

Now, both participants are able to determine a shared key, which is used only for transferring the final encryption key. All devices in the RF network have the final encryption key and using this key, they are able to communicate with all the RF devices at the third sensor system layer. When the final key is transferred to the RF module, it is allowed to start full communication with all devices with an RF interface across the RF network and it is guaranteed, that in the case of a network security breach, there is a possibility of how to securely establish a new encryption key without changing the firmware of each device.

V. Conclusion

In the article, the proposed sensor system is described including the necessary requirements for proper functioning of the system. The system topology of the sensor system and the communication principle at various levels of the system was described.

In the second part of the article, the secure communication possibilities at the highest level in the hierarchy of the sensor system for the Ethernet network were discussed. For Ethernet security, the TLS protocol was chosen. The basic principle of secure communication establishment and message encryption in the TLS protocol was also referred to.

The third part of the article deals with the security of the wireless section of the sensor system. Firstly, the wireless network scanning and address assignment to the individual RF modules in the second and the third layer was demonstrated in detail for the group key negotiation. For the group key arrangement, the GDH.3 protocol was used. The protocol allows adding another member to the group that was already established, as well as the removal of any group member. Through the negotiated group key, the server forwards to all RF modules the encryption key, which will be used for normal communication encryption (data in the RF network will be encrypted using the AES algorithm with a 128 bit key length).

At the lowest (fourth) model hierarchy layer, which contains the portable RF devices, initially, it was necessary to choose an authentication method for these devices and subsequently, upon successful authentication, the connection is established with the requesting device (DH protocol). After the establishment of a secure connection, the final encryption key is sent to the RF module.

The method of determining the encryption key in the proposed sensor system was designed and also illustrated.

Acknowledgements

Research described in this paper was financed by the TA04010476 project TACR "Secure Systems for Electronic Services User Verification" and by Czech Ministry of Education in frame of National Sustainability Program under grant LO1401. For research, infrastructure of the SIX Center was used.

REFERENCES

[1] W. Stallings, “Cryptography and Network Security: Principle and practice”, 5, Boston : Prentice Hall, 2011. 719 pages, ISBN 01-360-9704-9.

[2] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol”, Version 1.2. [Online] August 2008, [Cited: 28. 2 2015.] http://tools.ietf.org/pdf/rfc5246.pdf.

[3] M. Steiner, G. Tsudik and M. Waidner, “Diffie-Hellman Key Distribution Extended to Group Communication”, [Online] 1996, [Cited: 13. 2 2015.] http://corsi.dei.polimi.it/distsys/2007-2008/pub/p31-steiner.pdf.

[4] J. R. Vacca, “Network and System Security”, Burlington : Syngress/Elsevier, 2010, 368 pages, ISBN 15-974-9535-2.

[5] NIST Computer Security Division (CSD), :FIPS 197, Advanced Encryption Standard (AES)”, [Online] 26. November 2001, [Cited: 3. 2 2015.] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[6] Texas Instruments, Inc. “AES128 - A C Implementation for Encryption and Decryption”, [Online] A, March 2009, [Cited: 25. 2 2015.] http://www.ti.com/lit/an/slaa397a/slaa397a.pdf.

[7] J. F. Raymond and A. Stiglic, “Security Issues in the Diffie-Hellman Key Agreement Protocol”, [Online] 2003, [Cited: 16. 2 2015.] http://crypto.cs.mcgill.ca/~stiglic/Papers/dhfull.pdf.

[8] Y. Kim, A. Perrig and G. Tsudik, “Tree-based Group Key Agreement”, [Online] 2002, [Cited: 18. 2 2015.] http://www.ics.uci.edu/~gts/paps/kpt04a.pdf.

[9] K. Stewart, T. Haniotakis and S. Tragoudas, “A Security protocol for sensor networks”, Illinois : GLOBECOM '05, IEEE , vol.3, 2005, pp.1827-1831.

[10] G. A. Jolly, “Low-Energy Key Management Protocol for Wireless Sensor Networks”, [Online] 2002, [Cited: 3. 2 2015.] http://www.gta.ufrj.br/wsns/Security/LowEnergykey.pdf.

[11] H. Chan, A. Perrig and D. Song, “Random Key Predistribution Schemes for Sensor Networks”, [Online] 2003. [Cited: 2. 2 2015.] http://repository.cmu.edu/cgi/viewcontent.cgi?article=1025&context=ece.

[12] E. Shi and A. Perrig, “Designing Secure Sensor Networks”, December 2004, IEEE Wireless Communications, 2004, pp.38-43, ISSN 1536-1284.

[13] S. J. Jang, “A Study on Group Key Agreement in Sensor Network Environments Using Two-Dimensional Arrays”, [Cited: 28. 2 2015.] http://www.mdpi.com/1424-8220/11/9/8227, 2011, pp.8227-8240, ISSN 1424-8220.

[14] A. Perrig, “SPINS: Security Protocols for Sensor Networks”, [Online] 2002, [Cited: 21. 2 2015.] http://www.csee.umbc.edu/courses/graduate/CMSC691A/Spring04/papers/spins-wine-journal.pdf.

[15] A. Perrig, “The TESLA Broadcast Authentication Protocol”, [Online] 2002, [Cited: 21. 2 2015.] http://users.ece.cmu.edu/~adrian/projects/tesla-cryptobytes/tesla-cryptobytes.pdf.

[16] Texas Instruments, Inc. “C Implementation of Cryptographic Algorithms”, [Online] August 2012, [Cited: 28. 2 2015.] http://www.ti.com/lit/an/slaa547/slaa547.pdf.

[17] E. Bresson, D. Pointcheval and O. Chevassut, “Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks”, [Online] 2002, [Cited: 2. 2 2015.] http://www.di.ens.fr/~bresson/papers/BreChePoi02c_full.pdf.

[18] Y. Wang; G. Attebury, B. Ramamurthy, "A survey of security issues in wireless sensor networks," Communications Surveys & Tutorials, IEEE , vol.8, no.2, 2006, pp.2,23, doi: 10.1109/COMST.2006.315852.

Implementation of a Group Encryption System in a Cloud-based Environment

Tomasz Hyla
West Pomeranian University of Technology
Szczecin, Poland
e-mail: thyla@zut.edu.pl

***Abstract*— Nowadays, mobile devices offer almost the same level of functionality as standard personal computers. Cloud solution and faster Internet connections allow developers to build applications that do most of data processing in the cloud. On the other hand, cyber-crimes are growing problem and complex information system like cloud solutions are vulnerable to more threats. One of the most dangerous threats, i.e., data loss or leakage, requires countermeasures that will protect against dishonest cloud provider. Group encryption mechanisms are one of the key elements to ensure data privacy. This paper presents a new architecture for group encryption system that uses bilinear mappings. The architecture uses cloud solutions and supports mobile devices. Pros and cons of moving cryptographic operations to a cloud and resulting from the analysis of the demonstration system are discussed.**

Keywords-group encryption; cloud; mobile device; architecture; bilinear mapping.

I. INTRODUCTION

Nowadays, mobile devices offer almost the same level of functionality as standard personal computers. Of course some engineering applications requiring high-end processors are not available for mobile devices and someone might not use them due to lower screen sizes. However, cloud solution and faster Internet connections allow developers to build applications that do most of data processing in a cloud. The use of mobile devices and cloud computing increases, because of its desirable properties, like rapid elasticity or broad network access [1].

The cybercrimes are growing problem. The protection against them requires to constantly develop new security measures that will secure more and more complex systems that are using mobile devices and the cloud. The new threats related to the cloud together with proposed countermeasures are listed in [2]. One of the most dangerous threats, i.e., data loss or leakage, requires countermeasures that will protect against dishonest cloud providers.

One of the main business applications of mobile devices is reading and writing different types of documents. Those documents usually contain some kind of information that cannot be disclosed. When many entities are involved in documents' exchange, group encryption schemes can be used to ensure data privacy. The scheme must have properties that will allow to encrypt a document in such a way, that entities from authorised group can decrypt it when they will meet certain conditions.

The conditions required to access an asset can be described using access structures [3]. An access structure is a rule that defines how to share a secret, or more widely, who has an access to particular assets in information system. Access structures can be classified as structures with or without threshold. Although threshold access structures are frequently used, the non-threshold structures (i.e., general access structures) are more versatile.

In this paper, a new architecture for group encryption system, that uses advanced cryptographic operations is presented. The architecture uses cloud solutions and support mobile devices. Pros and cons of moving cryptographic operations to a cloud and these resulting from an analysis of demonstration system are discussed.

The reminder of this paper is as follows. Section 2 contains description of group encryption schemes with an emphasis on Certificate and ID-Based group-oriented Encryption scheme with General Access Structure (CIBE-GAS) scheme and library which contains its implementation. Section 3 introduces a cloud-based architecture for a group-encryption scheme together with a presentation of encryption and decryption processes in the demonstration system. The paper ends with conclusions.

II. GROUP ENCRYPTION

In the group encryption schemes a group of users must act together to decrypt or encrypt a file. This can be achieved in two ways. In the first one, group members consecutively encrypt the file using private keys. In the second one, the encryption key is calculated using private keys from each group member (the simplest solution is to use xor operation) by a designated user from the group. The designated user encrypts the file using the group key and deletes the key. In both cases, it is assumed that intermediate, temporary files (e.g., partial keys, partially decrypted files) that are created during encryption or decryption are deleted after the process is finished.

The private keys of each group member should be kept in secret. Several techniques exists: a key is created on the fly from a password that is entered by a user; a key is stored in an encrypted form and a user password is used to decrypt a key; or a key is stored inside a secure device (e.g., a smart card, a trusted platform module) and can be accessed after a user authenticates to the device.

Currently, many group encryption algorithm exists. Further in this section is described the group encryption algorithm, which is using pairing-based cryptography and has properties interesting from the perspective of cloud implementation.

A. Cryptograhic Scheme

The CIBE-GAS [4] is a group encryption scheme that is more suitable, comparing to threshold secret sharing methods, when the same access rights to decrypt data should be selectively assigned to all participants belonging to the same well defined group of users. The original version of CIBE-GAS scheme works with limited length messages only, while its modification Certificate and ID-Based group-oriented Encryption scheme with General Access Structure Hybrid (CIBE-GAS-H) [5] works with arbitrary length messages.

In CIB-GAS scheme a designated user (i.e., a dealer) is responsible for encrypting documents. The encryption algorithm requires as an input: dealer identity; public and private keys; public system parameters; information about privilege set of users who will be able to decrypt a document; and public share information belonging to users from the privileged set. During encryption no communication between the dealer and users from the privilege set is required. Public share information enables the dealer to encrypt a file in such a way, that only users who have a private keys associate with public share information will be able to partially decrypt a file. Decryption has two phases. In the first phase, each of users from the designated set using the ciphertext partially decrypts the text. In the second one, combiner (a user whom other users have transfer rights to decrypt the document) decrypts the ciphertext using the values obtained from partial decryption from all required users from the designated set.

The scheme combines three different ideas [4]: the secret sharing scheme [6], publicly available evidence of being a member of a particular group [7] and Sakai-Kasahara Identity Based Encryption (SK-IBE) scheme [8] with technique introduced by Fujisaki and Okamoto [9]. As a result the following properties where achieved:

a) the dispatcher (i.e., an entity which encrypts the document) is not required to know the structure of qualified subsets, which members are authorised to decrypt the information;

b) there is no need to designate a specific recipient of encrypted information - each member within a qualified subset can decrypt it; moreover, a dispatcher can temporarily remove some subgroups from having access rights to encrypted information (i.e., a dispatcher can arbitrarily select the recipients by overlaying the appropriate filter on the access structure);

c) the CIBE-GAS scheme is the certificate and identity based encryption scheme; in the scheme partial key created by trusted authority is published as a certificate and it allows simplifying the user’s identity verification.

More about the CIBE-GAS and CIBE-GAS-H schemes can be found in [4] [5].

B. Code Libraries

The CIBE-GAS and CIBE-GAS-H schemes were implemented and are a part of the mobile Pairing-Based Cryptography (mPBC) library which is a part of MobInfoSec project [10][11]. The schemes are built on bilinear mappings [12]. Bilinear mappings requires complex mathematical operations, so the schemes were implemented using Pairing-Based Cryptography (PBC) library written by Ben Lynn [13]. PBC is one of the first libraries that allowed to write a code using bilinear mappings. Developer only needs to know mappings properties and internals are hidden.

PBC is written in C language and uses GNU Multiple Precision Arithmetic (GMP) library [14]. Similarly to PBC, which hides bilinear mapping internals, the mPBC hides the CIBE-GAS schemes internals from users and provides high level Application Programming Interface (API). Hence, mPBC user does not need any knowledge about pairing-based cryptography. Also, mPBC contains data structure definitions, import and export functions and tests that demonstrate basic functionality. The mPBC purpose is to provide implementation of cryptographic schemes that can be used directly or indirectly on mobile devices.

Except CIBE-GAS and CIBE-GAS-H schemes mPBC library also contains other schemes that support digital signature and public key encryption built on top of bilinear mappings.

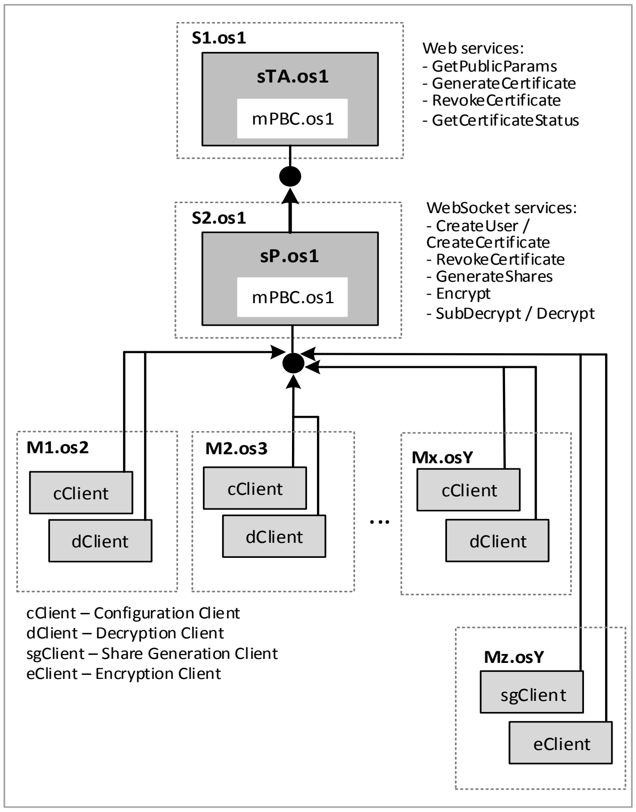


Figure 1. Cloud-based architecture.

The mPBC library can be easy used on Linux and Windows operating system as they natively support C language. The Android, Windows Phone 8.1 and iOS also supports C language, although mPBC library will require some minor modifications to work in each of these systems.

III. Cloud-based Implementation

The traditional way to implement a group encryption scheme in a mobile environment would be a client-server model. However, nowadays when mobile devices use many different operating systems and cloud solutions are available, the cloud-based approach has several advantages. The two are the most important. The first one, is a simple design of mobile clients. The second one, is that the mPBC library needs to be implemented only in one programming language.

A. Architecture

The cloud-based architecture consists of two logical servers, which provide two sets of services (Figure 1). The first one, Trusted Authority server (*sTA*) deployed on the server *S1* is responsible for management of system parameters, users and certificates as it is required by the CIBE-GAS scheme and provides appropriate services. The *sTA* server uses mPBC library in the version for *S1.os1* operating system.

The second server, secret Protection (*sP)* server, provides web socket services. The services enable cryptographic operations (from CIBE-GAS scheme) that normally would be executed on the mobile devices. The services provide operations like encryption and partial decryption. Transferring cryptographic operations to *sP* server eliminates the need to port mPBC library for every mobile operating system, but requires creating secure communication channels to mobile devices. Also, it requires that *sP* server is trusted and provides the same level of security as *sTA* server. This might be seen as drawback, but it also simplifies the development of client applications for mobile devices. Particularly reducing the number of security issues that must be considered.

A client application, deployed on the mobile device, can be developed as native or web browser application. In both cases, user’s private files (keys, parameters) are stored locally by *sP* server. Private files from all users are managed by *sP* server and are used indirectly by users through the cryptographic services provided by the *sP* server.

B. Demonstration System

The demonstration system consists of two servers written in C# language using MS Visual Studio 2013:

- the certification server, which provides *sTA* services;
- auxiliary server *sP* which provides functionality required to, among others, initiate certificates generation, encrypt, partially decrypt or decrypt a document; the server uses WebSocket technology to provide that functionality.

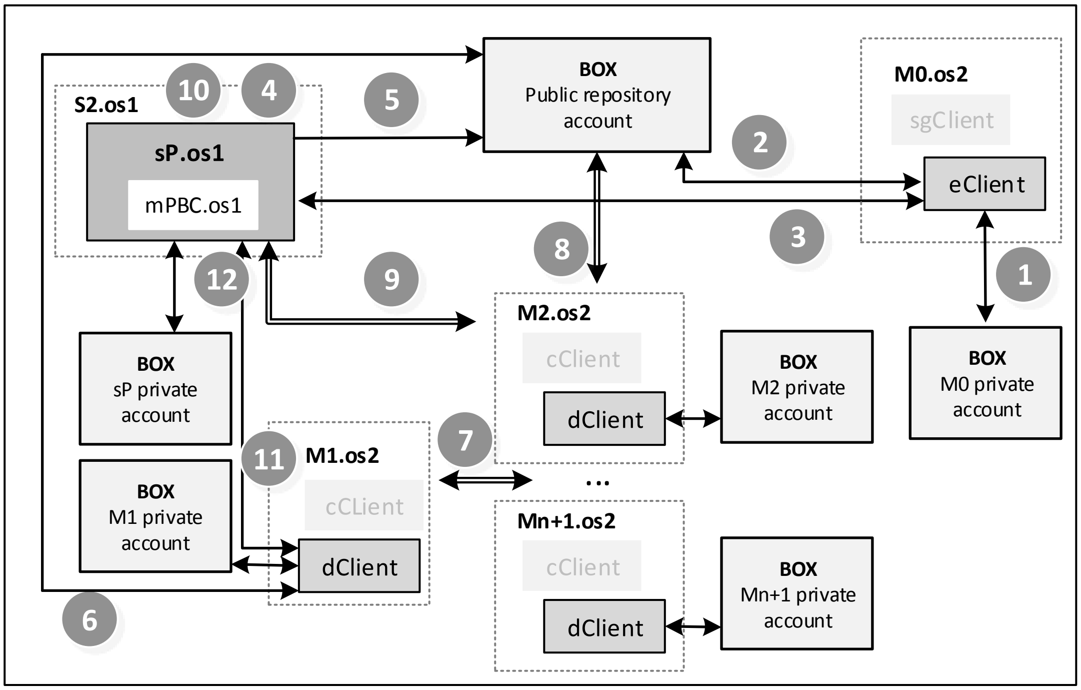


Figure 2. Encryption and decryption processes in the demonstration system

Client applications *cClient*, *dClient* (run by users) and *sgClient*, *eClient* (run by a dispatcher) are native Windows Phone 8.1 applications. The communication channel between mobile devices and *sP* server is not secured to simplify system development. In a working system technologies, like some version of TSL (Transport Layer Security), would be probably used. Also, other security measures, e.g., presented in [15], should be deployed to protect *sTA* and *sP* servers and client applications.

The demonstration system uses Box cloud drive to store data. Each mobile device, i.e., mobile device user, and each server has one associated box.com account. Also, there is one cloud drive for the public repository, which is publicly available for all client applications to temporary store generated files and then to share their public address to a specific authorised user.

The encryption and decryption processes main steps are presented in the Figure 2. The encryption process is as follows:

1. Dispatcher (a user with dispatcher rights in a dispatcher role) using *M0.eClient* application retrieves a public link to a file that he wants to encrypt.
2. A file with access structure information is retrieved from public repository, an access policy is created based on that file and then the policy is stored in *M0* private Box account.
3. *M0.eClient* sends to *S2.sP* the public links to the file and to the access policy.
4. The *S2.s:* downloads files from the links, gets user keys form *S2.sP* private Box account, downloads from the public repository required users' public share information, and using mPBC library executes CIBE-GAS-H Encryption algorithm.
5. The *S2.sP* stores an encrypted file and the access policy in the public repository Box account.

The decryption main steps are:

6. The *M1.dClient* searches repository and finds links to the selected encrypted file and to accompanying access policy. Then downloads the access policy.
7. The *M1.dClient* chooses *n* number of devices which are required to do partial decryption (based on the access policy) and sends them partial decryption requests.
8. Each *Mi.dClient* where *i=2..n+1*, downloads using provided links the encrypted file and the access policy.
9. Each *Mi.dClient* sends request to *S2.sP* to execute CIBE-GAS-H SubDecryption-H algorithm.
10. The *S2.sP* for each *Mi.dClient* executes the requested algorithm using keys associated with each *Mi.dClient* and sends to each device a public link to the partially decrypted file stored on *S2.sP* private Box drive.
11. The *M1.dClient* collects the public links from each *Mi.dClient* and sends to *S2.sP* links with requests to combine partially decrypted files.
12. The *S2.sP* executes the CIBE-GAS-H Decryption-H algorithm and returns to *M1.dClient* a public link to a decrypted file in its Box drive.

IV. CONSLUSION

In this paper, firstly, the CIBE-GAS scheme was described. Subsequently, the cloud-based architecture for the CIBE-GAS scheme was presented together with the presentation of encryption and decryption processes in the demonstration system.

The cloud computing have essential properties such as rapid elasticity and resource pooling [1]. This enables an operator to easily adjust number of server instances and other resources to change number of users. From the other side, resource pooling characteristic says that users generally do not know where physically their data are processed. The architecture presented in this paper is a typical hybrid cloud. The public repository can be in public cloud in contrast to servers and private drives which must be held in a private cloud. In presented demonstration system, clients use also the public cloud (Box.com drive), because it simplifies the implementation process.

The performance tests have shown, that in cases of encryption and decryption using *sP* server the time of cryptographic operations from CIBE-GAS scheme is significantly shorter in relation to the time required to retrieve documents from cloud drives. However, the total time of these operations is acceptable and mostly depends on Internet connection speed. It must be noted, that in a working system the authentication of users before usage of clients on mobile device is required.

The scalability is an important issue in cloud-based application development. The ability to scale up the application to millions of users depends mostly on the mutual relation between application instances. In the proposed architecture, the servers in the cloud can perform calculations independently for each request from client applications on mobile devices. This is very good situation as it is possible to run many instances of servers in the cloud with minimum effort.

The key advantages of cloud-based approach for encryption system implementation are: better scalability of the system when number of users increases and also faster and simpler implementation for different mobile operating systems. Especially, mPBC library which contains complicated cryptographic operation needs only to be implemented in the version for one operating system.

The main drawbacks are the necessity to create another trusted auxiliary server for cryptographic operations (*sP* server) and the need to create more trusted channels. The channels must be created between clients on mobile devices and between clients and *sP* server. Also, security threats associated with the cloud must be considered during system development. Particularly, the *sP* server must have the same security level as *sTA* server that manages user certificates.

The further works will mainly focus on general access structures as currently they are implemented in the simplest way that is required by CIBE-GAS scheme.

ACKNOWLEDGMENT

This scientific research work is supported by National Centre for Research and Development (NCBR) of Poland (grant No. PBS1/B3/11/2012) in 2012-2015.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST SP-800-145, September 2011.

[2] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing v3.0", P. Simmonds et al. (Eds.), 2011.

[3] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol. 72.9, 1989, pp. 56-64.

[4] T. Hyla and J. Pejaś, "Certificate-Based Encryption Scheme with General Access Structure," In: Cortesi, A. et al. (Eds.), CISIM 2012, LNCS, vol. 7564, Springer-Verlag, 2012, pp. 41–55.

[5] T. Hyla and J. Pejaś, "A practical certificate and identity based encryption scheme and related security architecture," K. Saeed, R. Chaki, A. Cortesi, S. Wierzchon (Eds.), CISIM 2013, LNCS, vol. 8104, Springer-Verlag, 2013, pp. 178-193.

[6] Y. Sang, J. Zeng, Z. Li, and L. You, "A Secret Sharing Scheme with General Access Structures and its Applications," International Journal of Advancements in Computing Technology, Vol. 3, No. 4, May 2011, pp. 121-128.

[7] Y. Long and Chen Ke-Fei, "Construction of Dynamic Threshold Decryption Scheme from Pairing," International Journal of Network Security, Vol.2, No.2, March 2006, pp. 111–113.

[8] R. Sakai and M. Kasahara, "ID based cryptosystems with pairing on elliptic curve," Cryptology ePrint Archive, Report 2003/054.

[9] E. Fujisaki and T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," In Proceedings of CRYPTO '99, Santa Barbara, CA, 1999, pp. 537–554.

[10] T. Hyla, J. Pejaś, I. El Fray, W. Maćków, W. Chocianowicz, and M. Szulga, "Sensitive Information Protection on Mobile Devices Using General Access Structures," The Ninth International Conference on Systems (ICONS 2014), IARIA, Feb. 2014, pp. 192-196, ISSN: 2308-4243, ISBN: 978-1-61208-319-3

[11] I. El Fray, T. Hyla, and W. Chocianowicz, "Protection Profile for Secure Sensitive Information System on Mobile Devices," K. Saeed and V. Snasel (Eds.): CISIM 2014, LNCS 8838, Springer-Verlag, 2014, pp. 636–650.

[12] B. Lynn, "On the implementation of pairing-based cryptosystems," PhD Dissertation, Avaliable from: http://crypto.stanford.edu/pbc/thesis.pdf, June 2007, [retrieved: February, 2015].

[13] B. Lynn, "Pairing-based cryptography library," Avaliable from: http://crypto. stanford.edu/pbc/, v-0.5.14, C language, LGPL license, [retrieved: February, 2015].

[14] The GNU Multiple Precision Arithmetic Library, Avaliable from: https://gmplib.org/, Edition 6.0.0, [retrieved: February, 2015].

[15] K. Salah, J. M. Alcaraz Calero, S. Zeadally, S. Al-Mulla and M. Alzaabi, "Using Cloud Computing to Implement a Security Overlay Network", IEEE Security and Privacy, Vol. 11, No. 1, January/February 2013, pp. 44-53.

Fuzzy Logic in Location-based Authentication

David Jaros, Zdenka Kuchtova, Radek Kuchta, Jaroslav Kadlec
Dept. of Microelectronics, FEEC
Brno University of Technology
Brno, Czech Republic
email: jarosd@feec.vutbr.cz, xkucht06@stud.feec.vutbr.cz, kuchtar|kadlecja@feec.vutbr.cz

***Abstract* - The article is focused on the possibility of using fuzzy logic principles in the authentication process in a computer network environment. Fuzzy logic could play an important role in authentication techniques where the input data is unclear or inaccurate and, therefore, the result of the process will also become unclear. Examples of this can be found in processing positional information concerning the user's location at a particular moment within the authentication process. The paper shows a possible solution to these difficulties by using a location-based authentication system which relates to the user's biometric data.**

Keywords* - *fuzzy logic; authentication; location; biometric data

I. Introduction

The article deals with the possibility of the usage of fuzzy logic in the authentication process and especially in location-based authentication. Currently electronic systems make decisions exclusively by using bivalent values when they perform an authentication. The more native approach could be by using a value within a continuous interval. One of the sources where fuzzy logic was used is detailed in [1]; this is focused on password security enhancement.

Let us imagine a specific situation: two people talk to each other, the first of them declares something to the second. The second person has to make a decision whether s/he will believe this declaration or not. The decision will not happen with absolute certainty.

Another aspect of the authentication process is the position of the authenticated user. The number of mobile devices such as laptops, smartphones and tablets continues to grow. The question "Where are you?" in the mobile environment is being asked more and more frequently. This is where fuzzy logic could be exploited with regard to location-based authentication, and in the authentication process generally. This will have the result that the user will be allowed access to protected services dependent on his/her position and on his/her trust level. This could be achieved by the use of several methods, for example; assessing the age of the provided position information and its accuracy.

In this article we will introduce the methodology of how to use fuzzy logic principles in the authentication system. The rest of the article is organized as follows. In Section 2 fuzzy logic is introduced and an overview is given concerning what is required for its usage in authentication systems. Section 3 will discuss the possibilities of how to get the user's positional information and how to transfer it through a chain form, from user to authenticator. Section 4 will show an example of an authentication system and an authentication terminal designed for location-based authentication. Section 5 is concerned with future work and issues that have to be taken into account.

II. Fuzzy Logic Fundamentals

Fuzzy logic is an extension of set theory and logic operators [2].

In comparison with classic set theory the main difference is membership of an element to the set. In classic set theory an element is a member of a set or not, no other option is possible. In fuzzy logic theory an element is mapped to a fuzzy set by usage of a membership function. The difference is described in (1).

$$\mu_K: X \to \{0,1\}\mu_F: X \to [0;1], \qquad (1)$$

, where μ_K is a membership function of classic set and maps elements to universe set X into two member set {0,1}. and μ_F is a membership function of fuzzy set and maps elements from universe set into values in the range from 0 to 1. This relation is depicted in the Figure 1.

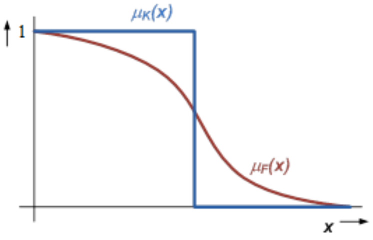


Figure 1. Fuzzy set membership function example

In fuzzy logic, we can talk about a "linguistic variable". When we consider a variable, in general, it takes numbers as its value. If the variable takes linguistic terms, it is called a "linguistic variable". Let us imagine the next example. We have a variable *X* called password strength, which has values (terms) *weak* $(\mu_N(n))$, *moderate*$(\mu_S(n))$ and *secure* $(\mu_B(n))$. We can define the member function for each term as follows.

ISBN: 978-1-61208-399-5

$$\mu_N(n) = \begin{cases} 1, for\ n \in <0;1>;\ n \in Z^+ \\ \frac{3-\mathrm{n}}{2}, for\ n \in (1;3>;\ n \in Z^+ \\ 0\ for\ n > 3;\ n \in Z^+ \end{cases} \qquad (2)$$

$$\mu_s(n) = \begin{cases} 0, for\ n\ <2\ \vee n > 6;\ n \in Z^+ \\ \frac{\mathrm{n}-2}{2}, for\ n \in <2;4>;\ n \in Z^+ \\ \frac{6-n}{2}, for\ n < 4\,;6>;\ n \in Z^+ \end{cases} \qquad (3)$$

$$\mu_B(n) = \begin{cases} 0, for\ n < 5;\ n \in Z^+ \\ \frac{n-5}{2}, for\ n \in <5;7>;\ n \in Z^+ \\ 1, for\ n > 7;\ n \in Z^+ \end{cases} \qquad (4)$$

The equations stated above are displayed in Figure 2.

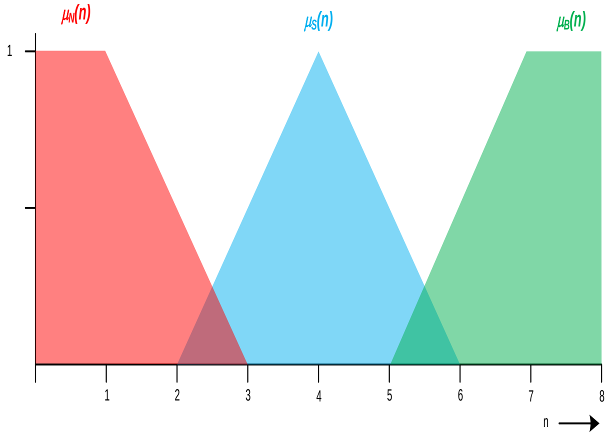


Figure 2. Password strength fuzzyfication

The fuzzy system is composed of input variables, output variables and inference rules. The inference rules are responsible for behavior of the system. Generally the rule is written in the form:

If*(ascendant)* ***then*** *(consequent),*

where the *ascendant* is one or more logically connected input variables and *consequent* is the output variable. Usually a system consists of a set of rules most of the time in several stages.

The logical connection of variables could represent Mamdani's implication, which could be explained by the equation 5 and Figure 3.

$$\mu_{\mathfrak{I}}(x_1, x_2) = \min\{\mu_A(x_1), \mu_B(x_2)\} \qquad (5)$$

With regards to (6) the membership function of consequent will be cropped on layer equals to a minimum of values for both ascendant *min(α,β)*. The situation is illustrated in the figure 3.

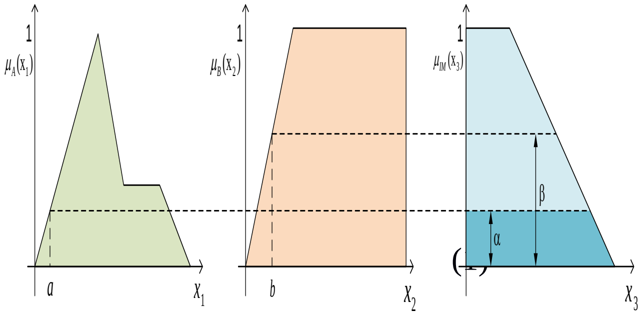


Figure 3. Mamdani's fuzzy implication

In this case we need to get a numerical value of the output linguistic value of the defuzzyfication to be done. Several solutions are possible for this task. In our case we will use the strategy: Center of Area (COA).

The widely used COA strategy generates the center of gravity of the possibility distribution of a fuzzy set (6).

$$x_{OUT} = \frac{\sum_{k=1}^{r} \alpha_k x_k}{\sum_{k=1}^{r} \alpha_k} \qquad (6)$$

III. USER'S POSITION

The position information could be figured out absolutely or relatively.

The relative position is stated as proximity to the object with a known position in the system. Objects with a known position are called anchor points in the system. This way of how gaining information concerning a position is suitable especially in a Global System for Mobile Communications (GSM) network. Here the user's position is estimated by exploiting the known position of the Base Transceiver Station (BTS), in the network where there is a mobile terminal connected [3]. This kind of localization is mentioned in references [1], [2], [3].

The second way is possible by using an absolute position. Information about the position consists of two or three coordinates. This way is usually used in cartography or in the Global Positioning System (GPS).

In the authentication and authorization process, we can consider both kinds of interpretation concerning the user's position.

In certain cases it is not necessary to use an absolute position. If we know the user is located in the proximity of an anchor point it could be sufficient information. The accuracy of the position information decreases with increasing distance from the anchor point.

In the Figure 4 you can see a basic schematic of the principle of relative positioning, as previously described in [6]. The shaded area is a room covered by the signal from the anchor point (x_{AP}, y_{AP} and z_{AP} are coordinates of the anchor point's position). Between the authenticator and anchor point there has to be the establishment of mutual trust, it means the authenticator believes in the information form of the anchor point and vice-versa. If the user is in the signal range of the anchor point, it means it has to be able to communicate with the anchor point. If the user's terminal claims to the authenticator it is located near to the anchor point, the authenticator is able to validate this claim by the authenticator.

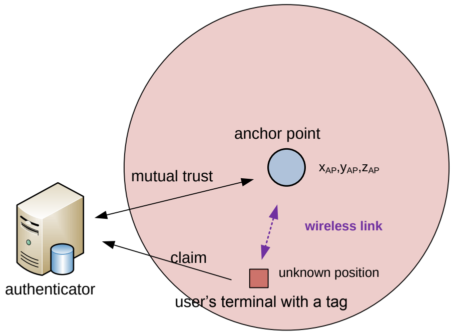


Figure 4. The relative positioning principle

In the rest of the paper we will consider the next sources for localization that could be used as position formation sources in an authentication terminal. For outdoor usage it would be a GPS receiver and a terminal GSM module. Exclusively for indoor usage it will be a module with a wireless interface regards corresponding to IEEE 802.11.

IV. THE AUTHENTICATION SYSTEM

In relation to the previous sections, here is an example of an authentication system concerning the processing of a user's positional information. Below is a list of steps which should be gone through in the design period.

Description - in this step there should be a general description of the behavior of future system.

Authentication techniques - all considered authentication techniques should be considered.

Relations – all relations between used authentication techniques should be specified.

Splitting, used techniques – all listed techniques in the second step should be split into two groups. The first group will be formed by techniques performed in an authentication terminal and the second in an authenticator.

Difficulty – we should imagine how strong each technique is and also how trusted it is as well.

Influences – all the main influences for used techniques, which could have an impact on the authentication process have to be taken into account.

Quantification - all listed influences should receive a value which describes its importance.

Scheme assembly – with regards to the list of authentication techniques, their relations and splitting into two sites schemes of how a whole system could be assembled.

Fuzzyfication – all input and output variables with their influences have to be transformed to linguistic variables

Inference rules – the behavior of the whole system and especially output variables are dependent on used rules.

An example is given below detailing an authentication system which performs a strong authentication where the user's position is one of the processed factors.

A user will use an authentication terminal (Figure 5) to prove its identity to the authenticator. The authentication terminal contains modules for the determination of position such as: GPS receiver, terminal GSM or radio interface IEEE 802.11. Because we need to prove the user's position we have to demonstrate the user is in the same place as the authentication terminal. The authentication terminal uses a fingerprint reader for this task, as well as a tested biometric authentication technique [6]. Positional data is not sent to the authenticator until the fingerprint is checked. The authentication terminal is assigned to the concrete user (it's personalized by a fingerprint and encryption key KEY and password). Data is encrypted by an encryption key (unique for the terminal) is transmitted from the terminal to the authenticator. Note, we assume using AES128 or AES256 as secure enough encryption algorithms [6]. It means each terminal could be used as a unique token in the system and works as an additional authentication factor. Positional data is strongly related to the time when the positional data was created. In the Figure 5 you can see several sources of time information.

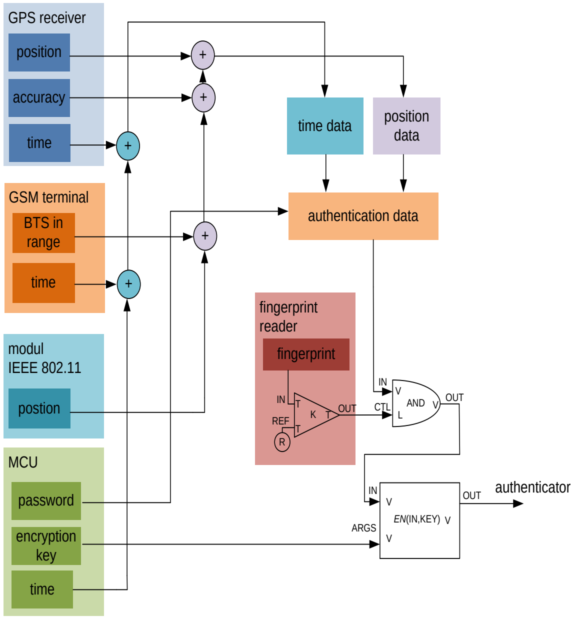


Figure 5. The authentication terminal

The authenticator which stores the user's profile is on the other side. The user's profile holds all the necessary data for the user for defined locations (from where the user could be authenticated), encryption key, password etc. The schematic of the authenticator is illustrated in Figure 6. All necessary data is stored in a knowledge base, also *inference rules* or amplification coefficients A_X. Each subsystem on Figure 6 is a fuzzy system which performs the authentication of a specific factor (for example, *subsystem GPS* processes data from the GPS receiver).

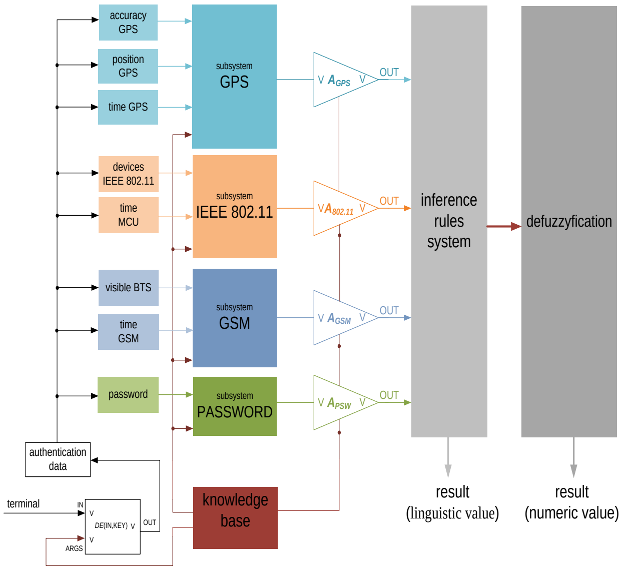


Figure 6. The authenticator

The behavior of the whole system is defined by inference rules which are part of the knowledge base. The rule is described in section 2 and it usually takes the form of a "if then" condition. Table 1 lists the top level rules for the authenticator for evaluating the state of positional information.

TABLE I. THE INFERENCE RULES

Trust GPS	Trust GSM	Trust IEEE 802.11	Position
high			well proved
low	high	high	well proved
low	low	low	not proved
low	moderate	low	not proved
low	high	low	proved
low	low	high	proved

The result of the processing of the submitted authentication data is the level of trust that the user has identified which he claims and he is in the location where he claims. How the level is varied with different input data can be seen in Figure 7. The result in Figure 7 is based on the application inference rules set (table 1) and provided by Matlab. There we can see how trust concerning positional information $\delta_{POSITION}$ is dependent on the results from the subsystems, in this case from the GSM subsystem δ_{GSM} and the IEEE 802.11 subsystem $\delta_{IEEE80.11}$. We can see the highest value of $\delta_{POSITION}$ is in the case when both of the input values are also in high values. This is the simplest example of dependence but could vary with different inference rules according to authentication system requirements.

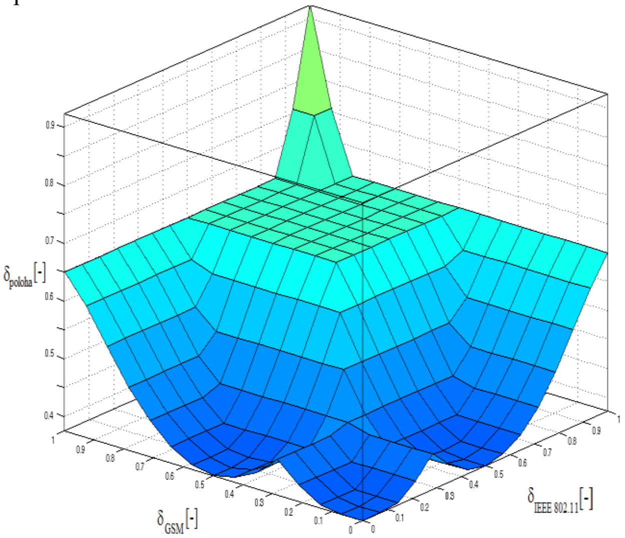


Figure 7. The trust to position information

The development board was designed for an authentication terminal (Figure 8). The board is based on 16-bit RISC (Reduced Instruction Set Computing) microcontroller MSP430F5529 from Texas Instruments and is equipped with a new version of eMMC memory, where all required data are stored. The board contains IEEE 802.11 radio interface RN131C from ROVING.

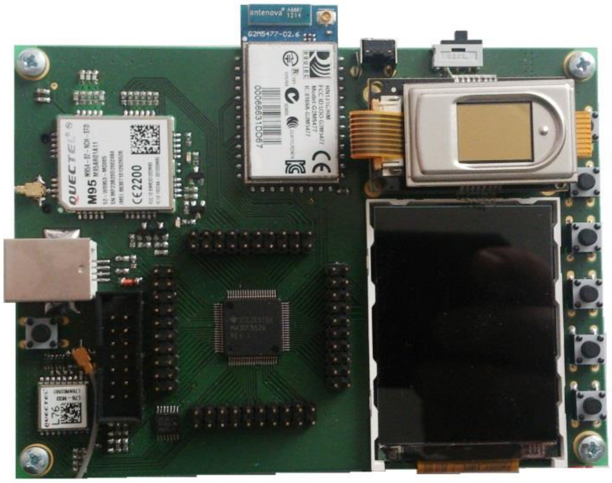


Figure 8. The Authentication terminal development board

This radio is able to list Media Access Control (MAC) and addresses devices in the neighborhood with their appropriate Received Signal Strength Indication (RSSI). Two modules have been chosen from Quectel. The first one is the GPS receiver L76 and the second one is the GSM module M95. The board is equipped with five buttons, user defined functions and an LCD with a resolution of 240 x 320 pixels. As was mentioned previously the board contains a fingerprint reader: FPC-AM3 from Fingerprint.

V. Conclusion

The article deals with the possibility of using fuzzy logic principles in the authentication process in a computer network environment. The basic idea is make the result of the authentication process more diffusive with regards to vague input data (authentication factors). Presently systems produce a result in bivalent logic as “Yes/No” or “True/False”. In many cases the right one is somewhere in the middle. This could be correct especially in the evaluation of positional information, this means where an authenticated user is located. This information could be unclear or inaccurate and therefore, the result will also be unclear or inaccurate. Fuzzy logic could represent a possible way of how to control a system with vague values.

The focus is on positional information. We introduce a GPS receiver or GSM terminal as positional information sources in the authentication process. We also introduce the idea of relating positional information with a human biometric element for strong authentication (the user has to be in the same place as the authentication terminal).

Next, the paper presents a possible way of how to set up a basic authentication model step by step. Further to this the example of an authentication system is presented. For testing purposes the development board of an authentication terminal was designed and realized.

Future work will be aimed at testing the presented principles in a real environment. Although the basic principles were verified in a previous version of an authentication terminal, our next work will focus on the implementation of advance techniques related to positional information.

Acknowledgment

Research described in this paper was financed by the TA04010476 project TACR "Secure Systems for Electronic Services User Verification" and by the Czech Ministry of Education in the framework of the National Sustainability Program under the grant LO1401. For research, infrastructure of the SIX Center was used.

References

[1] W. de Ru, J. H. P. Eloff, "Enhanced Password Authentication through Fuzzy Logic," , 1997.

[2] Kwang H. Lee, First Course On Fuzzy Theory and Application. Berlin: Springer-Verlag, 2005.

[3] M. Ibrahim and M. Youssef, "A Hidden Markov Model for Localization Using Low-End GSM Cell Phones," in Communications (ICC), 2011 IEEE International Conference, 2011, pp. 1-5.

[4] N. Deblauwe and G. Treu, "Hybrid GPS and GSM localization — energy-efficient detection of spatial triggers," in Positioning, Navigation and Communication, 2008, pp. 181-189.

[5] A. Goetz, S. Zorn, R. Rose, G. Fischer, and R. Weigel, "A time difference of arrival system architecture for GSM mobile phone localization in search and rescue scenarios," in Positioning Navigation and Communication (WPNC), 2011, pp. 24-27.

[6] D. Jaros, R. Kuchta, R Vrba, "The Location-based Authentication with The Active Infrastructure," in : The Sixth International Conference on Internet and Web Applications and Services, Sint Maarten, 2011, pp. 228-230.

[7] A. K. Jain. On The Uniqueness of Fingerprints. [retrived: January, 2015],[Online].http://biometrics.cse.msu.edu/Presentations/AnilJain_UniquenessOfFingerprints_NAS05.pdf

[8] A. Bogradlow, D. Khovratovich, Ch. Rechberger. Biclique Cryptanalysis of the Full AES[retrived: January, 2015], Research Microsoft. [Online]. http://research.microsoft.com/en-us/projects/cryptanalysis/aesbc.pdf

Creating an ITIL-based Multidimensional Incident Analytics Method: A Case Study

Kari Saarelainen
Management Consulting
KPMG Finland
Helsinki, Finland
e-mail: kari.saarelainen@kpmg.fi

Marko Jäntti
School of Computing
University of Eastern Finland
Kuopio, Finland
e-mail: marko.jantti@uef.fi

***Abstract*—Many IT organizations have recognized incident categorization as a challenge because there are no general policies or guidelines for incident categorization. This leads to incident categorization usually being seen as an optional task for the specialists who handle incidents. The research problem of this study is as follows: what type of incident and root cause categorization model would be efficient and would also support ITIL-based (IT Infrastructure Library) continual service improvement? The results of this study consist of two parts: First, the software incident and root cause categorization model, which helps an IT organization to categorize incidents and their root causes effectively and recognize the weak points of the IT service delivery, and second, the provision of the lessons learned for improving incident categorization and measurement practices. The research was conducted as a case study that was carried out in an IT service company.**

Keywords-** **IT service management; ITIL; continual service improvement; incident management, root cause; categorization

I. INTRODUCTION

IT service providers are constantly seeking more effective methodologies, processes and tools in order to optimize the efficiency and quality of the process. IT Infrastructure Library (ITIL) is the most widely used best practice framework for IT service management [1]. ITIL provides a set of guidelines for managing information technology (IT) infrastructure, development, and operations as well as addresses the quality of IT services in several different ways. The most notable quality concepts within IT service management are service level management, incident management, problem management processes and Continual Service Improvement (CSI), which is related to all the stages of IT service lifecycle. This study will focus on the Incident Management and Service Operation processes and CSI [3] lifecycle stage, which have given guidelines used within an action research cycle.

The purpose of incident management is to restore normal service operation as quickly as possible and minimize the adverse impact on business operations [2]. Problem management seeks to minimize the adverse impact of incidents and problems on the business that are caused by underlying errors within the IT Infrastructure. Problem management process is invoked, e.g., if similar types of incidents are recurred or a major incident has occurred and the root cause needs to be analyzed. Root causes can also be divided in categories. Incidents are categorized in incident logging by service desk personnel. This category can be changed during incident management process. The category in this phase usually involves the selection of service, activity, type, function or configuration item (CI) and answers the question “what”. Root cause is the output of problem management and answers to the question “why”.

In this study, we gather data from a fairly large number of incidents, which have their root cause analysis activities made and reports written. Using attributes available (incident category, root cause category, location, incident duration, date, responsible, etc.) and simple analysis tools we are able to, in most cases, to point statistical deviations, which helps to recognize the weak points of services and processes. Most of the attributes above are fairly standard, and the measurement is unambiguous independent of person, country, and organization, e.g., date, time, duration, etc. Incident and root cause categories, are however, not described in IT service management standards such as in ISO/IEC 20000 [4] or frameworks such as ITIL and Control Objectives for Information and Related Technology (COBIT) [5]. In order to statistically analyze or otherwise identify recurring incidents or incident types or root cause types, the incident and root cause categories must be agreed and defined.

A. Related work

Previous research on incident management has addressed the importance of incident classification. Caldeira and Brito e Abreu [6] have used statistical methods to analyze product related incidents. Jäntti and Kalliokoski [7] have identified challenges related to service desk work and reported that service desk staff had problems in finding and selecting a correct Service Level Agreement and configuration item for incidents. Dan et al. [8] use business driven IT management and risk decision making theory to prioritize incidents. One of the key objectives of the incident categorization activity is to provide information for identifying the same type of incidents or incidents that are caused by the same root cause. Do Mar Rosa et al. [9] have used incidents to identify provided IT services for the organization’s service catalogue. Marcu et al. [10] have presented an incident correlation model based on category-based correlation aiming at identifying similar incidents by automating the process. Cusick and Ma [11] discuss how incident management approach (including single point of contact and escalation procedures) was established in a service provider organization.

While in the IT service management the problem management process is responsible for investigating the root cause of incidents by Root Cause Analysis (RCA), in Software Engineering this activity is called a Defect Causal Analysis (DCA). DCA [12] aims at identifying causes of defects in order to prevent defects or to find them earlier. Defect prevention activity is also included in Capability Maturity Model (CMM) [13].

DCA typically includes causal analysis meetings, group meetings, that focus on identifying root causes as well action team meetings. Software Engineering Institute has introduced a Framework for Counting Software Defects and Problems [14]. This framework provides description how to classify problems and defects. In the Personal Software Process [15], the defect collection method includes a defect classification scheme and defect attributes. According to Jäntti et al. [16], IT organizations seem to have remarkable difficulties in managing defects and problems. In our study, we highlight the role of improving incident analysis.

B. *Our contribution*

The main contribution of this paper is an incident and root cause categorization model. The purpose of this model is to locate and point weaknesses in IT service delivery as a part of CSI.

This categorization can be used in service improvement efforts that are done as one-time exercises, as part of regular service quality reviews or as on-going activities integrated in ITSM systems. The purpose of the model is to help IT organization to identify the weak areas (people, process, technology, etc.) in IT service management, that are usually sources of identified root causes.

The rest of the paper is organized as follows. The research problem and methods are described in Section 2. The creation and validation of the incident and root cause categorization model is covered by Section 3. The analysis of the findings is covered in Section 4. The conclusion in Section 5 summarizes the case study.

II. Research Methods

The research problem in this study is: How incident and root cause categorization can be used to support CSI. This qualitative research study was built using the case study and action research methods. The research problem was divided into the following research questions:

RQ1: What type of information can be used in creating an effective incident and root cause categorization model?

RQ2: How the combination of incident and root cause categories could be used for trending of incidents.

RQ3: What other incident related attributes could be used for more effective incident trending and other IT service improvement activities.

A. *Case Organization and Data Collection Methods*

The research was conducted in 2012 – 2014 on several distinct occasions reflecting several customers and several types of environments. All of these service environments were maintained by a single IT service provider, later the case organization. Multiple data collection methods proposed by Yin [22] were used during the study and the following data sources were used:

- **Participant observation:** Meetings and discussion with managers.
- **Interviews**: Interviews of roles responsible of services offered to customers and interviews of experts of service provider involved in the incident
- **Documents:** Incident reports, process descriptions, work guides and guidelines
- **Records and archives:** Change, incident and problem records.
- **Physical artifacts:** ITSM tool.

The research followed the action research cycle as proposed by Baskerville [23]. The five phases of this cycle are presented in Fig. 1: A. Diagnosis, B. Action Planning, C. Action Taking, D. Evaluation, E. Specifying Learning.

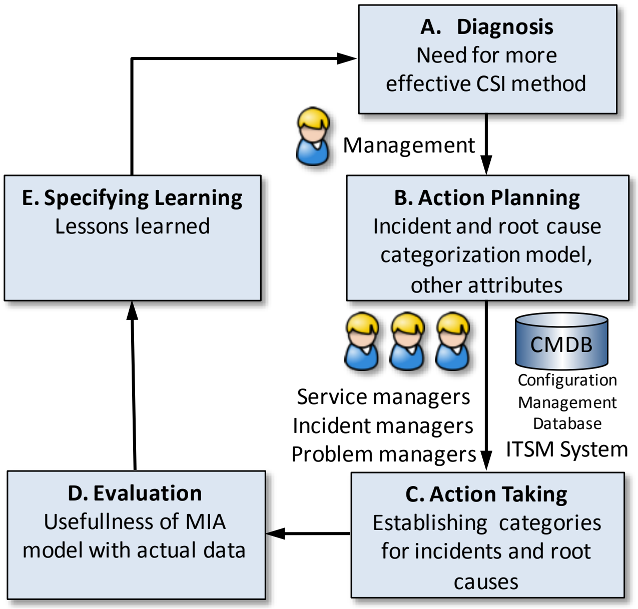

A. Diagnosis
Need for more effective CSI method
Management
E. Specifying Learning
Lessons learned
B. Action Planning
Incident and root cause categorization model, other attributes
CMDB
Configuration Management Database
Service managers
Incident managers
Problem managers
ITSM System
D. Evaluation
Usefullness of MIA model with actual data
C. Action Taking
Establishing categories for incidents and root causes

Figure 1. The Action Research Cycle in this research

B. *Data Analysis Methods*

The data of this action research study were collected and analyzed using a within case analysis technique [17]. The incident categorization scheme and findings were validated through discussions and improvement workshops with the representatives of the case organization. The principal author was responsible for producing a summary of findings that was delivered to the case organization. The established incident categories were improved taking into account the comments from case organization’s managers and some new categories were added to the original classification.

III. Results

As main results, we document the action research cycle that focused on improving the categorization of IT service incidents. The action research resulted in a method for analyzing incidents with multiple attributes. This method is called Multidimensional Incident Analytics (MIA).

A. Diagnosis

The diagnosis phase aims at identifying the primary problem, which needs to be solved through action research. This phase started by organizing a meeting with the case organization's management. The objective of the management was to check the current state of IT service delivery and identify improvement areas.

The case organization's management requested a more effective method for continual service improvement of IT services. There are several alternative and complementary methods to approach the improvement. The management chose the improvement based on analysis on those incidents, from which there has been a separate incident report including root cause analysis.

B. Action planning

The action planning phase specifies organizational actions to solve and investigate the problem identified in diagnosis phase. The action planning phase started with designing the incident analysis method. According to ITIL framework, incident categorization can be used for incident analysis for, e.g., establishing trends for use in proactive problem management and other service improvement activities, as well as measuring the performance of incident management. This method sorts the incidents mainly according to chosen incident category.

There are, however, several other attributes, which can be connected to the chosen incident. If these attributes are in numerical format or they are lists with a defined set of values, one can apply data analytics methods on them.

Root cause is one of the most useful attributes. The incident record usually contains also other potentially useful information for incident analysis, for example date and time of incident, duration of the incident, CI (this may also the basis for incident categorization), priority, impact, and relationship with other CIs, incidents, problems, changes or known errors. Other sources for attributes of incidents to be analyzed include monitoring and capacity information, and application portfolio attributes.

In order to keep the data manageable, incident category, root cause and occurrence data and time were chosen incident attributes to be analyzed. The incident reports were not in the standard format, and especially the root causes were free-form textual descriptions. For analysis purposes relevant incident and root cause categories were to be developed.

1) Incident categories

In general, incident category answers the question "**what** (was impacted by the incident)". It can be used to support proactive problem management activities such as trend analysis and the targeting of preventive action. Identified problems from these activities will be used in continual service improvement activities. In ITIL framework [2], an incident is defined as "an unplanned interruption to an IT service or reduction in the quality of an IT service". A problem is ”a cause of one or more incidents". A root cause is "the underlying or original cause of an incident or problem". Problem management process is responsible for the resolution of the root cause.

All incidents should have a standard category schema. This provides faster access to incident and troubleshooting information, and also better support proactive incident trending. Since organizations are unique on the categories, it is hard to find a universal set of categories, especially on the detailed level. ITIL proposes the following generic steps for incident categorization [1]:

1 Hold brainstorming sessions with stakeholders.
2 Generate the initial main categories incl. 'other'.
3 Use the main categories for a short trial period.
4 Analyze the incidents logged during the trial period. The number of incidents logged in each category will confirm if the category was successful. If the number of incidents is high in 'other' category, consider creating a new main category.
5 A breakdown analysis of incidents in main categories tells, if subcategories are required.

2) Root cause categories

Root cause categories answer the question "**why** (the incident happened)". For proactive problem management sole incident category inspection is a not a very powerful tool. Several root causes may cause symptoms that may be spread across several incident categories. For example, human errors, facility problems (electricity, cooling, humidity), supervisory, guidance and process problems typically cause incidents in several different incident categories. Incident and root cause categories can be considered orthogonal, independent from each other. Virtually all the root cause categories can exist in all the incident categories.

In literature, root cause classification or categorization is handled very little. Where root cause categories are handled, it is done in other domain, e.g., in nuclear plant context [21], independently of domains [18] or only one main category of root causes, e.g., human factors [20]. Generally the root cause analysis methods [1][18][19]][21] and categories can be used also in other domains or they can be used as a basis of root case category development. There is no guidance in root cause categorization in ITIL unlike in incident categories. As a phenomenon it is similar to incident category, and the researchers decided to handle it with the same procedures as in incident categories.

3) Other attributes

Other potential incident attributes from incident records, changes, CIs, other related incidents, problems, changes or known errors, monitoring and capacity information, and application portfolio were not systematically studied. If this information existed in the incident report, it was taken into account when drawing conclusions and proposing improvements. Starting time of the incident was, however, taken as an attribute. Starting time made possible to place the incident on a time line and in correlate it with other events at the same time.

Information about the responsible party was left as an attribute. This is often a big question in an incident, especially if penalties are to be paid because of service level breaches. The conclusions and recommendations are also different, if the incident was because of actions of customer or subcontractor rather than the actions of IT service provider itself. The attribute responsible party answers the question ”**who** (was responsible)”.

C. Action taking

The action taking phase focus on implementing the planned action.

1) Establishing incident categories

The case under study was a company internal quality improvement effort. The purpose was to collect a representative amount of incident reports, where the root cause was analyzed. Since this was a set of separate reports and not incident tickets there were no obligations to follow the allocated categories in incident records. The researchers were free to choose the categories, which suited best to service improvement activities.

ITIL encourages using multilevel categorization, which is also supported in most ITSM systems. Examples of typical categories of requests include:

- **By service:** For example, a request to create a new user email account may be part of an email service.
- **By activity:** For example, password reset, laptop installation or printer cartridge replacement.
- **By type:** The request is categorized by its type, e.g., an informational request and a standard change.
- **By function:** For example, service desk, technical management, IT operations management and application management.
- **By CI type:** The type of CIs that the request or event impacts.

The chosen category type reflects also who is providing and using that information and what is the strategy in incident trending. Customer of IT service or business may be more interested to categorize in business service terms incidents, which are visible in the customer interface. Then categorization by service or by activity may be more natural. If the objective is to measure and improve internal quality of IT service delivery, one usually takes categories close to internal organization (by function) or by CI type.

The environments included environments dedicated to customers and also shared by several customers. In all of these IT service delivery was organized as technology oriented teams, technical domains. The teams performed several processes and activities, e.g. change management, incident management, problem management, monitoring, etc. The organization of technical domains reflected the hierarchy of CIs. Because of all of these reasons it was a natural choice to take the responsibility areas of technical domains as starting point of main incident categories. After some trial exercises and iterations, the researches added couple of extra categories. Because the aim was not in this phase to create a global ongoing incident categorization for use in incident management process and with the ITSM system, no lower level categories were defined. The following incident categories were chosen:

- **Network:** All passive and active network devices including switches and firewalls.
- **Server:** Servers in general. Memory incidents were separated in a category of its own.
- **Memory:** Incidents in physical memory, e.g. lack of memory and physical memory errors
- **Storage**: Storage systems and storage networks.
- **Database:** Errors in databases.
- **Application:** Application service running in servers.
- **Integration/job processing:** Integrations and batch job between services.
- **Facility systems:** Electricity UPS, cooling, etc.
- **Other:** Incidents not fitting in other categories.

2) Establishing root cause categories

According to ITIL the operation of ITSM as a practice is about preparing and planning the effective and efficient use of the four Ps: the people, the processes, the products (services, technology and tools) and the partners (suppliers, manufacturers and vendors). These were chosen as a starting point. Since it was decided to establish a set of attribute answering the questions who, partners were moved to that attribute. After analyzing initial incident profiles, it was decided to split products into two subcategories: software and hardware. Since later it was suspected that a some server and network devices running certain operating systems were unreliable, a third device category was added: firmware.

Root cause categories:

- **Software:** bugs, malfunctions and configuration errors in applications.
- **Firmware:** bugs, malfunctions and configuration errors in firmware and operating systems of servers and network and security devices.
- **Hardware:** Malfunctions and errors in hardware, e.g. fans, CPU, memory, bus, cards, etc.
- **Process:** Poorly defined, implemented, communicated or supervised process, e.g. too loose change management process
- **Human error:** Something that was not intended by the actor thus causing the incident.
- **Other:** Any other root cause for the incident.
- **Unknown:** Root cause was not found or it was not analyzed.

*3) **Establi**shing responsible party categories*

The responsible parties were divided into following subcategories

- **IT service provider:** IT service provider self caused the incident.
- **Network operator:** The incident was caused by network operator.
- **HW/SW vendor:** The incident fell in this category, if hardware or software had bugs, and IT service provider could not avoid them, e.g. with updates according to recommendations.
- **Customer:** Incident was caused by erroneous actions or faulty information of customer.
- **Other:** Some other party caused the incident.

D. Evaluation

The evaluating phase aims at evaluation the outcomes of the action research. In our case we focused on evaluating usefulness of the MIA method.

After the data collection and categorization work the researches had 165 incident reports from different environments with enhanced incident records. From the point of view of this study the core attributes of the incident records were: incident category, root cause category, starting time of the incident, and responsible party. The tool used in this analysis was a spreadsheet application. One may, however, apply also more sophisticated tools, such as data analytics and BI tools in order to analyze incidents. With more sophisticated tools it is possible to study a larger set of attributes (dimensions).

E. Specify learning

In this study, we specified learning in the form of lessons learned. In the Specify learning phase organization identifies and creates knowledge gained during the action research. Both successful and unsuccessful actions enable learning. The following lessons learnt were derived.

Lesson 1: Incident and root cause logging, categorization and analysis should be one part of CSI. The analysis of theoretical frameworks showed that incident logging is part of operative incident management, while incident root cause analysis is performed in problem management. Our empirical findings suggest that it might be useful to include the root cause of incident also in the incident record but at least in the problem record. Adding additional dimensions to incident analysis may bring essentially more accuracy and efficiency to high level incident analysis.

Lesson 2: Root cause categories are needed. Root causes are not typically categorized, when they are analyzed and logged. The root causes are rather described only in free text form if at all logged. In order to perform systematic analysis root causes should also be categorized in the same way as incident categories.

Lesson 3: Investigate human errors. Investigation of root causes revealed a significant number of human errors.

Lesson 4: Improvements in incident classification may reflect to other ITSM processes. Incident categorization triggered improvements in change management process.

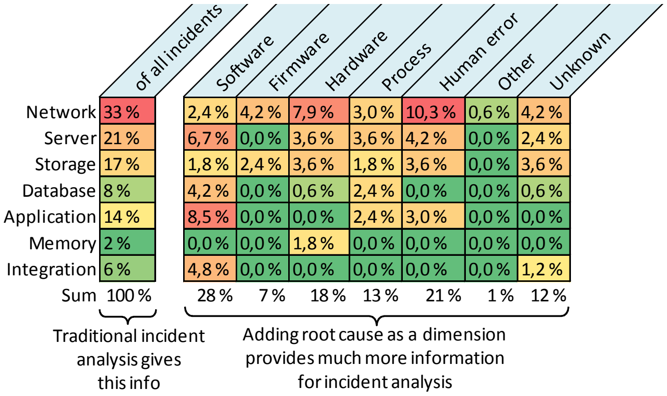
Lesson 5: Clarify the difference between categorization and classification. The difference between categorization and classification is unclear to the ITSM practitioners. Our theory-based findings show that ITIL v2 used a term classification while ITIL v3 used categorization and prioritization.

Lesson 6: Capture multiple root causes. Ensure that the tool is able to capture multiple root causes in IT service management tool. Root cause changed in appr. 30% of cases after interviewing the persons involved with the incident.

IV. ANALYSIS

The analysis phase started by establishing a two-dimensional analysis (incident category – root cause category) for IT service incidents. The correlation of incident categories with root cause categories answered the question, whether the some of the root cause categories was dominant in some of the root cause categories. Example is presented in Fig. 2. Traditionally, the incidents are inspected in one dimension. Adding root causes to incident categories improves essentially the accuracy of the analysis (see Fig. 2). E.g. in the example we see that 33% of all errors are network errors. The other columns allocate this figure in root cause categories. Now we see, that 10,3 % of network incidents are because of human errors and 7,9% hardware problems and 4,2% firmware problems. Excessive human errors can be handled, e.g. with tighter change management, training, supervision, etc. Firmware errors may be mitigated, e.g. by paying attention to use proven versions of the software. One may also consider other vendors, especially, if the hardware problems were caused by the vendor.

The analysis above could be done to each environment. Environments may also easily be compared with each other or the baseline consisting of all environments.



	of all incidents	Software	Firmware	Hardware	Process	Human error	Other	Unknown
Network	33 %	2,4 %	4,2 %	7,9 %	3,0 %	10,3 %	0,6 %	4,2 %
Server	21 %	6,7 %	0,0 %	3,6 %	3,6 %	4,2 %	0,0 %	2,4 %
Storage	17 %	1,8 %	2,4 %	3,6 %	1,8 %	3,6 %	0,0 %	3,6 %
Database	8 %	4,2 %	0,0 %	0,6 %	2,4 %	0,0 %	0,0 %	0,6 %
Application	14 %	8,5 %	0,0 %	0,0 %	2,4 %	3,0 %	0,0 %	0,0 %
Memory	2 %	0,0 %	0,0 %	1,8 %	0,0 %	0,0 %	0,0 %	0,0 %
Integration	6 %	4,8 %	0,0 %	0,0 %	0,0 %	0,0 %	0,0 %	1,2 %
Sum	100 %	28 %	7 %	18 %	13 %	21 %	1 %	12 %

Traditional incident analysis gives this info
Adding root cause as a dimension provides much more information for incident analysis

Figure 2. Example of incident category - root cause analysis.

We found that improvements in incident classification triggered improvement of other ITSM processes. This supports the findings of a previous study [7], where improvement of classification led to improvement of service level management. Our findings also support the findings of an existing defect management study [16] that proposed that organizations have difficulties in managing problems and defects. One of the major difficulties is defect classification.

It seems that traditional defect classification models, such as classification scheme of Personal Software Process are mixtures of software development lifecycle activities, architecture and infrastructure building blocks [15]. These are useful in the IT service design and transition phases, but don't fit in the IT service operation phase. The IT service operation phase emphasizes services, where products in maintenance phase are in production use in a given environment. We propose that IT service management could benefit from continuous root cause analysis and MIA as a CSI method. Additionally, service oriented classification models have some unique features that software oriented classification models lack, such as service level agreements.

V. CONCLUSIONS AND FUTURE WORK

The research problem in this study was: What type of incident and root cause categorization model would efficiently support ITIL-based continual service improvement. The unit of the study was a Finnish IT service provider company and its incident management process.

The research was conducted according to the phases of the action research cycle. The main contribution of the study was to describe how a MIA model was designed, and what was learned during the action research.

There are certain limitations related to this study. First, action research typically includes several iterative research cycles. However, we focused on describing only one improvement cycle that focused on improving the analysis of IT service incidents. Second, action research benefits from a collaborative effort. Although we used multiple data sources, more effort could have been put on collaborative actions instead of a consultancy style problem solving. Third, regarding method triangulation, we could have used organizational change theory more extensively to support our technology- and process-based problem solving approach.

Future research could aim at refining our MIA model by adding new dimensions to it, as well as exploring the root cause category models based on IT service management practices.

ACKNOWLEDGMENT

We would like to thank the case organization's representatives for valuable feedback and responses that helped us to perform this study.

REFERENCES

[1] Cabinet Office, ITIL Service Strategy. The Stationary Office, UK, 2011.

[2] Cabinet Office, ITIL Service Operation. The Stationary Office, UK, 2011.

[3] Cabinet Office, ITIL Continual Service Improvement. The Stationary Office, UK, 2011.

[4] ISO/IEC 20000:1, Part 1: Service management system requirements. ISO/IEC JTC 1 Secretariat, 2010.

[5] COBIT 5, Control Objectives for Information and related Technology: COBIT 5: Enabling Processes. ISACA, 2012.

[6] J. Caldeira and F. B. e Abreu, "Influential factors on incident management: Lessons learned from a large sample of products in operation," in Product-Focused Software Process Improvement, A. Jedlitschka and O. Salo, Eds., vol. 5089. Springer Verlag, 6 2008, pp. 330–344.

[7] M. Jäntti and J. Kalliokoski, "Identifying knowledge management challenges in a service desk: A case study," in Proceedings of the Second International Conference on Information, Process, and Knowledge Management, eKNOW 2010. St. Maarten, Netherlands Antilles: IEEE Computer Society, February 2010, pp. 100–105.

[8] W. Dan, Z. Zhiqiang, and S. Hao, "An incident prioritization algorithm based on BDIM," in Proceedings of the ICCMS '10. Second International Conference on Computer Modeling and Simulation, 2010., vol. 3, Jan 2010, pp. 536–540.

[9] M. do Mar Rosa, N. Gama, and M. da Silva, "A method for identifying IT services using incidents," in Eighth International Conference on the Quality of Information and Communications Technology (QUATIC), 2012, Sept 2012, pp. 172–177.

[10] P. Marcu, G. Grabarnik, L. Luan, D. Rosu, L. Shwartz, and C. Ward, "Towards an optimized model of incident ticket correlation," in IFIP/IEEE International Symposium on Integrated Network Management, 2009. IM '09., June 2009, pp. 569–576.

[11] J. Cusick and G. Ma, "Creating an ITIL inspired incident management approach: Roots, response, and results," in Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, April 2010, pp. 142–148.

[12] D. N. Card, "Learning from our mistakes with defect causal analysis," IEEE Software, vol. 15, no. 1, pp. 56–63, January/February 1998.

[13] CMMI, Standard CMMI Appraisal Method for Process Improvement (SCAMPISM) A, Version 1.3: Method Definition Document. USA: Software Engineering Institute, Carnegie Mellon University, 2011.

[14] W. Florac, "Software quality measurement a framework for counting problems and defects," Technical Report CMU/SEI-92-TR-22, 1992.

[15] I. Hirmanpour and J. Schofield, "Defect management through the Personal Software Process," Crosstalk, The Journal of Defense Software Engineering, 2003.

[16] M. Jäntti, T. Toroi, and A. Eerola, "Difficulties in establishing a defect management process; a case study," in Proceedings of the 7th International Conference on Product-Focused Software Process Improvement, ser. LNCS 4034, J. Munch and M. Vierimaa, Eds. Amsterdam, The Netherlands: Springer-Verlag, Berlin Heidelberg, June 2006, pp. 142–150.

[17] K. Eisenhardt, "Building theories from case study research," Academy of Management Review, vol. 14, 1989, pp. 532–550,.

[18] Paul F. Wilson, Root cause analysis: A Tool for Total Quality Management, ASQ Quality Press, Jan 1, 1993.

[19] Lee N. Vanden Heuvel and Donald K. Lorenzo, Root Cause Analysis Handbook, Rothstein Associates Inc., Brookfield, Connecticut, USA, 2008.

[20] Wiegmann, D. A., & Shappell, S. A., A human error approach to aviation accident analysis: The human factors analysis and classification system, Burlington, VT: Ashgate Publishing, Surrey, United Kingdom, Ltd, 2003.

[21] Doe guideline – Root cause analysis guidance document (DOE-NE-STD-1004-92), U.S. Department of Energy, Office of Nuclear Energy, Office of Nuclear Safety Policy and Standards, Washington, D.C., USA, 1992.

[22] Robert Yin. Case Study Research: Design and Methods. Beverly Hills, CA:Sage Publishing, 1994.

[23] Baskerville, Richard L. "Investigating Information Systems with Action Research," Communications of the Association for Information Systems: Vol. 2, Article 19,1999.

MZZ-GA Algorithm for Solving Path Optimization in 3D Printing

Mateusz Wojcik, Leszek Koszalka, Iwona Pozniak-Koszalka and Andrzej Kasprzak
Department of Systems and Computer Networks
Wroclaw University of Technology
Wroclaw, Poland
e-mail: wojcik.mateusz991@gmail.com, {leszek.koszalka, iwona.pozniak-koszalka, andrzej.kasprzak}@pwr.edu.pl

Abstract— **This paper is focused on fused deposition process which is one of the technologies that can be used in rapid prototyping process. This process is divided into four different stages, one of which is path planning. This stage has a remarkable impact on the overall timing of the printing process. In this paper the implemented algorithms for solving the path optimization problem are presented. The properties of the implemented MZZ-GA algorithm are investigated, with the use of the designed two-stage experimentation system. Basing on the obtained results, we can conclude that the proposed approach seems to be promising.**

Keywords-algorithm; pathfinding; printing; optimization; simulation experiments

I. INTRODUCTION

Nowadays rapid prototyping is one of the fastest growing technologies. This process allows for creating a solid object without any specific tooling. The main advantage of this process is the ability to create a very complex object, in short time. There are several different systems which can be used in this technology, including: (i) Stereo-lithography (SL), (ii) Selective laser sintering (SLS), and (iii) Fused deposition modeling (FDM). In this paper, we use FDM which belongs to the so-called Layered Manufacturing (LM) technology, in which a solid object is produced by the deposition of material layer. The object in LM has to be processed before printing. According to [1] this process requires the completion of the four main tasks:

- Object orientation - the best orientation for the object is determined.
- Support generation – the additional element is generated in order to holds the parts of the object (after printing these additional elements can be dissolved).
- Slicing – a special algorithm extracts the layers (in the vector form) from the object. The 3D model is converted into the 2D images.
- Path planning – the algorithm plans the moves of the extruder.

For the purposes of this paper we have focused on the last task. The path planning problem can be divided into two different sub-problems: (i) path generating, and (ii) path optimization. To solve the first sub-problem an algorithm should generate and group the tool path segments into individual sub-paths. The paths can belong to one of the two groups: the contour paths or the raster paths. The raster paths are filling the interior section of the layer (always after contours). To solve the second sub-problem, an algorithm should optimally link the sub-paths which were found previously. The criteria for the optimality that should be met include: best possible surface quality, minimum tool wear, shortest machining time achieved, or minimum machining cost.

There are several different algorithms for solving these sub-problems. The algorithms proposed for solving path generating problem are based on strategies, such as: Raster [2], ZigZag [3], Contour [4], Spiral [5], and Fractal space curves [6]. The algorithms proposed for solving path optimization problem are based on approaches, such as: Genetic Idea [7], Adaptation of TSP [8], and Neural Network [9].

In this paper, the algorithm called MZZ-GA is proposed for solving path planning problem. This algorithm enables path generating with the designed Modified Zig Zag (MZZ) algorithm, and next, using the obtained result, it solves path optimization problem with the implemented Genetic Algorithm (GA). The properties of MZZ-GA are evaluated with a designed and implemented experimentation system.

This paper is organized as follows. In Section II, the related works are discussed. In Section III, the formal model of the considered problem is stated. Section IV presents the proposed algorithms for a solution to the problem. Findings from computational experiments are presented in Section V. Conclusion and plans for further research in the area appear in Section VI.

II. RELATED WORK

A. Paths generating algorithms

In paper [10], it is showed how to improve planning process for Rapid Prototyping / Manufacturing for the complex product models, such as biomedical models. That work contains a description of several different algorithms, including a path generating algorithm used to generate contour tool–paths along the boundary. Also, the zig-zag tool-paths used to generate paths for the internal area of the layer are described. The most interesting is the proposition for the solution of the path optimization problem with the use of zig-zag algorithm in which the variable for the optimization is the slope of the zig-zag direction. The most important conclusion of that work is that the mixed tool-path algorithm can balance the geometrical quality and the effectiveness.

Paper [11] presents the Zig-Zag algorithm. This algorithm was developed for 3-axis computer numerical control (CNC) machine. In this algorithm the tool moves in a straight line in a feed-forward direction. Also, an algorithm for finishing operation on the machines is described. Because that algorithm was developed for CNC machines, the authors of [11] have also developed a tool holder collision detection algorithm.

B. Path optimization algorithms.

In [8] two different methods for path optimization in Layered Manufacturing are presented. One of them is based on genetic approach, and the other one is based on the strategy which is used for solving a combination of Asymmetric Traveling Salesman Problem and Assignment Problem (TSP-Assign). The authors of [8] formulated the problem of path finding as constructing a set of curves on a layer, from which the algorithm should find the optimum sequence and direction of curves. They conclude that GA optimization can improve path planning tremendously, but this method is computationally expensive. Moreover, they state that TSP-Assign algorithm for path optimization was even more time-consuming than the GA approach. Also, an approach which combines GA and local search approach is proposed. This technique may improve path planning by reducing the jump distance by up to 50%.

The authors of paper [12] propose two different algorithms. The first is based on a simple greedy option (nearest neighbor procedure). A heuristic algorithms starts from the upper right corner and in every step adds points which belong to the counter that has not been visited yet. After the algorithm has visited all corners it begins searching for the path in the internal area. The second algorithm is based on the combination of the nearest and the farthest insertion method. At the beginning, it adds a point which belongs to the first corner. After that it checks if it should add a point which belongs to the second corner. The authors of [12] draw the conclusion that depending on geometry different methods can give better or worse results. The first algorithm produces better results when there are only a few counters and a few continuous raster segments. If the number of those objects increases, the second algorithm can produce better results.

III. PROBLEM STATEMENT

In this paper, we concentrate mainly on the path planning problem, in particular on the path optimization on the single layer. In the mathematical form, the problem can be stated as follows:

Given:

- Printing layer as a set of the binary points (an example of the layer can be seen at Figure 1):

$$X_{ij} = \begin{cases} 1\ if\ printing\ point \\ 0\ otherwise \end{cases}\quad i \in n\ j \in m$$
(1)

- Lengths between two points defined by (2):

$$L_{X_{ab}\,X_{cd}} = \sqrt{(a-c)^2 + (b-d)^2}$$
(2)

where:

- n – Lengths of the printing layer,
- m – Width of the printing layer.

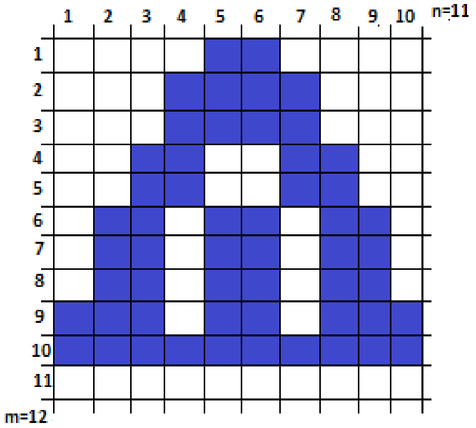


Figure 1. Example of the printing layer (grey points refer to printing area).

To find:

- V – order of points that will be visited:

$$V = [X_{ab}\ X_{cd}\ X_{ef\,\dots}]$$
(3)

- The decision about using the tool to extrude the plastic on the single point:

$$U_i = \begin{cases} 1 - if\ V_i will\ be\ printed \\ 0 - otherwise \end{cases}$$
(4)

- The total length of the final path:

$$L = \sum_{k=1}^{p}((V[k] - V[k-1]) * U_i)$$
$$= L_{printed} + L_{switching\ path},$$
(5)

where p is the number of the visited points.

Such that:

- *Lopt = min(L).*

Subject to constraints:

- Each point can be visited only once:

$$V[i] \neq V[j].$$

IV. ALGORITHMS

There were implemented two different algorithms for paths generating and one algorithm for path optimization. In this section, the both algorithms are presented in detail.

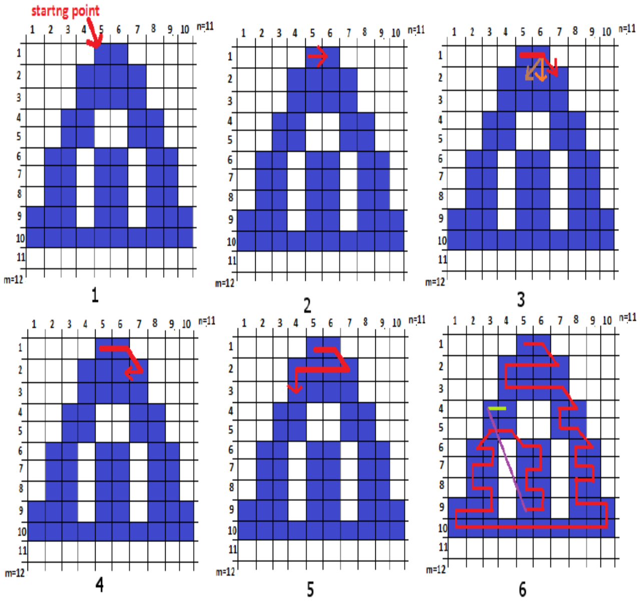


Figure 2. An example - the MZZ algorithm in action.

A. Modified Zig Zag algorithm

The MZZ algorithm works in the following way:

1. Start from top, left corner (Figure 2.1) – extruder's home position is point (0,0); because of that top left corner will be the closest point from extruder.
2. Move into the right corner (Figure 2.2) – next step is to select all points which lay to the right of the first pixel.
3. Check the bottom pixels (Figure 2.3) – when there are no more pixels on the right, the algorithm checks starting from the right corner below if there are pixels available.
4. Check the pixels above – when previous step ends with no results, algorithm checks the pixels above. Otherwise the algorithm skips this step.
5. Choose the pixel in the rightmost position (Figure 2.3) - if any of those pixels is available – i.e. it has not been visited yet – the rightmost pixel is selected.
6. Check the pixels to the left (Figure 2.4) – after step 4 the algorithm starts checking and selecting pixels which lay to the left of the pixel which has been chosen in step 4.
7. Check the pixels below (Figure 2.5) – when there are no more pixels to the left, the algorithm checks if the pixels below are available. It starts from the lower left corner.
8. Check the pixels above– when the previous step ends with no results, the algorithm checks the pixels above. Otherwise the algorithm skips this step.
9. Choose the pixel in the leftmost position (Figure 2.5) - if any of those pixels is available – i.e. it has not been visited yet – the lower leftmost pixel is selected.
10. Repeat the procedure – the algorithm repeats the steps from step 2 to 9. This creates the zig-zag-shaped paths.
11. When there are no more pixels available to select in the neighborhood of the previous pixel, but there are some available in the whole layer, the algorithm goes to step 1.

The important difference as compared to the standard Zig-zag algorithm appears, when the MZZ algorithm finishes checking available pixels on the bottom. Standard Zig-Zag algorithm will now start a new path, while the MZZ algorithm will also check the pixels above. If there are any pixels available there, it will continue adding those pixels (Figure 2.6).

B. Genetic Algorithm for path generating

GA as path generating algorithm works in the same way as the standard GA [7]. It starts with generating population. The single solution in this population is the path which contains all the points of the layer. The initial solutions in the population are generated randomly. After this GA selects the best solutions from the population, i.e., the algorithm selects the "m" solutions with the shortest path lengths. The next step is to crossover the paths to make another population. The crossover process is not so complicated. At first the algorithm randomly chooses two parents – two different paths from which the crossover will be made. Then algorithm selects "l" random points from the first parent. The rest of the points belong to the second parent. As in standard GA, in "p" percent of the new paths random mutations occur. If the mutation occurs, two points are swapped. The whole procedure is repeated "t" times.

C. Genetic Algorithm for path optimization

This algorithm is responsible for optimizing the path that is already found by MZZ algorithm. The standard MZZ algorithm connects sub-paths as they are found. This means that at the beginning the MZZ algorithm finds the first path, then the second, and so on. After that it connects the last point from the first path with the first point from the second path, and so on.

In the proposed implementation the GA looks for the best linking of the sub-paths found by MZZ algorithm. An example of the implemented GA can be seen in Figure 3, where 3(a) shows sub-paths which were found by the MZZ algorithm; 3(b), 3(c) and 3 (d) are possible final paths which appear after linking of the sub-paths.

The process of this algorithm goes like that of the standard Genetic Algorithm. At first it generates population. The single solution in this population is the path which contains all sub-paths. The first population is generated randomly. After that the “m” best solutions are selected. In this case the better solution is when the path is shorter. The next stage is to crossover the paths to make the next population. This process is done as follows. At first it selects two parents – paths that will be used to make a new solution.

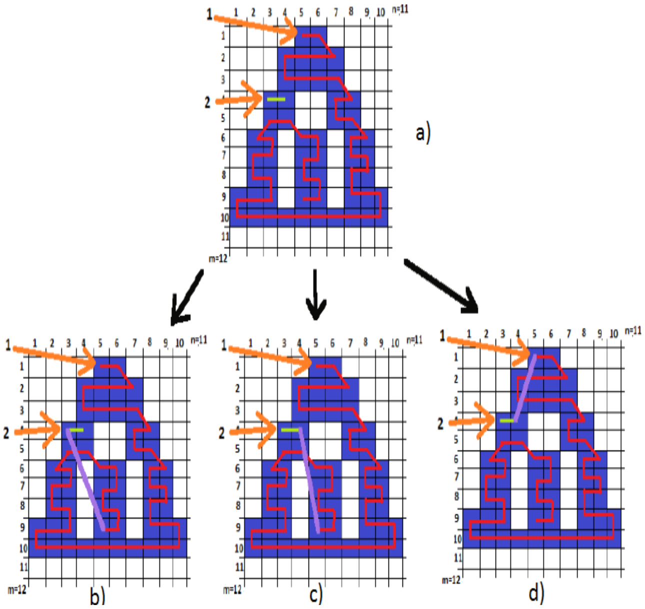


Figure. 3. An example of possible paths made from linking sub-paths.

The new solution contains the “l” sub-paths from the first parent, and rests of the sub-paths belong to the second parent. The parameter “l” is chosen randomly. The next process is mutation. A small percentage of the whole population passes through this process. In this process the randomly chosen sub-paths are swapped.

V. INVESTIGATION

A. *Software*

The experimentation system, i.e. the application for testing algorithms has been designed by the authors of this paper and implemented in Java. The experiments were carried out on computer with Intel Core i5-3210M CPU 2.50GHz.

B. *Experiment design*

We took into consideration four different objects denoted as Pic. 1, Pic. 2, Pic. 3, and Pic. 4. All objects can be seen in Figure 4. The main differences between the objects were in the number of pixels and complexity. The first three objects had a small number of pixels and were not complex. The last object had a much bigger number of pixels and was much more complex than the other objects.

Two-stage experiments were carried out in the following way:

1. In the first stage of the experiment the algorithms GA path generating and MZZ path generating were tested for all the objects. In order to adjust the internal parameters, the GA path generating algorithm was tested in respect of two internal parameters: the number of population and the number of epoch.

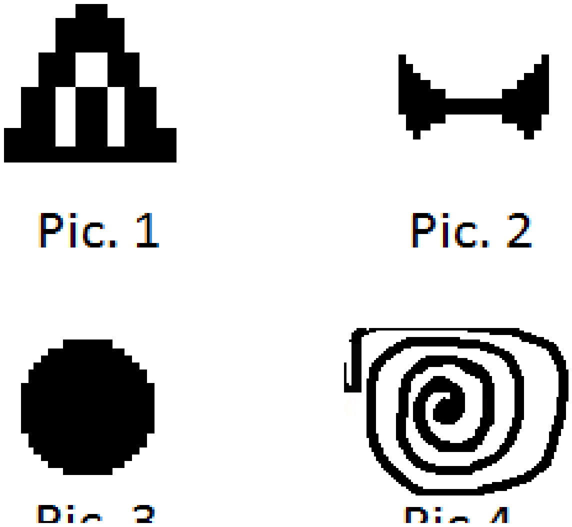


Figure. 4. Four different objects which are tested during the experiment.

2. In the second stage of the experiment the optimization algorithm MZZ-GA was tested. In this experiment GA was linking sub-paths which were found by the MZZ algorithm. Also, the influence of the number of population and the number of epoch on the obtained results was tested.

The experiments with GA path generating algorithm and MZZ-GA path optimization algorithm were carried out with different number of epoch (NP) and different number of population (NE). Three different values of NP and NE for each of the tested objects were planned.

As the result of a single experiment two values were taken:

- The execution time of the algorithm,
- The length of the founded path.

Because the GA contains random operations, the single experiments were repeated ten times. Therefore, the averaged values of two metrics were considered as the indices of the performance. Those metrics are:

- The average execution time (AET),
- The average length of the founded path (ALP).

Moreover, the standard deviation was calculated (sdLP).

C. *Results of experiments*

Table 1 shows the results of the experiments in the first stage for GA path generating algorithm. Table 2 shows the final results of MZZ-GA, i.e. the results of GA path optimization connected to MZZ.

It may be observed in Table 1 that GA path generating gave better results when the number of epoch (NE) was bigger than the number of population (NP).

The same conclusion can be drawn from Table 2. However, in this case, for objects with low number of pixels and less complexity, there is no need to use such a big number of populations and of epoch. For the objects named as Pic. 1, Pic. 2, and Pic. 3, the same results of ALP were obtained.

TABLE 1. RESULTS OF GENETIC ALGORITHM AS PATH GENERATING ALGORITHM.

	GA PATH GENERATING				
	AET [MS]	SDLP	ALP	NP	NE
PIC. 1	139	4.5	105.9	100	100
	1783	0.9	108.25	1000	100
	939	5.4	84.9	100	1000
PIC.2	176	2.2	130.4	100	100
	1062	2.8	135.0	500	100
	912	1.8	126.5	100	500
PIC. 3	1216	3.4	538.1	100	100
	4170	1.1	541.6	300	100
	3996	3.1	555.7	100	300
PIC. 4	606	8.7	10903.0	10	10
	8698	1.8	10912.0	100	10
	7201	25.2	10873.0	10	100

In the case of the object denoted as Pic. 4, the one with a high number of pixels and high complexity, using bigger number of epoch gave better results than using bigger number of population. Moreover, for those parameters the solution was found in shorter time.

TABLE 2. RESULTS OF GENETIC ALGORITHM AS PATH OPTIMIZATION ALGORITHM.

	GA PATH OPTIMIZATION				
	AET [MS]	SDLP	ALP	NP	NE
PIC. 1	93	0	57.2	100	100
	1416	0	57.2	1000	100
	542	0	57.2	100	1000
PIC. 2	71	0	108.7	100	100
	1392	0	108.7	1000	100
	583	0	108.7	100	1000
PIC. 3	55	0	304.0	100	100
	1325	0	304.0	1000	100
	534	0	304.0	100	1000
PIC. 4	781	51.4	4016.6	100	100
	4244	33.4	4015.7	500	100
	4065	37.3	3798.9	100	500

Table 3 shows the results of both indices of performance (ALP and AET) for all algorithms. The best results obtained for GA path generating (from Table 1) are shown in column GA-GEN, and the best results obtained for MZZ-GA optimization (from Table 2) are shown in the column GA-OPT. The column MZZ presents the results obtained with MZZ path generating algorithm.

TABLE 3. COMPARISON OF ALL ALGORITHMS

No. OF PIXELS	GA-GEN		MZZ		GA-OPT	
	AET	ALP	AET	ALP	AET	ALP
55	939	84.9	0	5.5	93	57.2
89	912	126.5	0	129.5	71	108.7
294	1216	538.1	1	484.2	55	304.0
2335	7201	10873.0	12	4158.8	4065	3798.9

It can be seen that the GA path generating produced the worst results – ALP and AET are the highest. The path length for the complex object (Pic. 4) is twice as big as the length of the path found by MZZ algorithm. From the results of MZZ algorithm it can be seen that work time in this case is the shortest, and that path lengths are almost in every case better than in cases when they were given by GA path generating algorithm. The results for GA path optimization algorithm are the best. It can be seen that for any object the GA significantly improved path lengths found by MZZ algorithm. It is also evident that the execution times are much shorter than those produced by the GA as path generating, but longer than for simple MZZ algorithm.

VI. CONCLUSION

On the basis of the obtained results of experiments, it can be concluded that MZZ algorithm is not the best algorithm that can be used for path finding individually. Also, the classic genetic path generating algorithm should not be used as a path generating algorithm.

The genetic algorithm certainly should be recommended (as the path optimization algorithm) when it is used together with MZZ path generating algorithm as a combined MZZ-GA algorithm.

When using MZZ-GA algorithm, it is necessary to bear in mind that its merits are governed by the reasonable number of epoch and the proper number of population (see Table 2). However, the user may face some difficulties - for the bigger number of pixels it will be difficult to use the same number of epoch and the same number of population as for smaller data.

In further research in the area the authors are planning to consider more algorithms for path generating and path optimization based on other evolutionary approaches, e.g. the one presented in [13]. In particular, using the hybrid approach and contour approach in constructing effective algorithms seems to be very promising.

There are also several interesting issues that might be discussed in future work in this area, such as designing and implementing experimentation systems to conduct the multistage experiments in the automatic manners, along with the issues presented in [14].

ACKNOWLEDGMENT

This work was supported by the statutory funds of the Department of Systems and Computer Networks, S40029_K0402, Wroclaw University of Technology, Poland.

REFERENCES

[1] K. P. Venuvinod and M. Weiyin, Rapid prototyping Laser-based and Other Technologies, Springer, 2004.

[2] M. R. Dunlavey, “Efficient polygon-filling algorithms for raster displays”, ACM Trans. Graph., 1983, pp. 264–273.

[3] S. C. Park and B. K. Choi, “Tool-path planning for direction-parallel area milling”, Comput Aided Design, vol. 32, 2000, pp. 17–25.

[4] R. Farouki, et. al., “Path planning with offset curves for layered fabrication processes”, J. Manuf. Syst., vol. 14, 1995, pp. 355–368.

[5] H. Wang, et. al., “A metric-based approach to two-dimensional tool-path optimization for high-speed machining”, J. Manuf. Sci. Eng., 2005, pp. 127-133.

[6] P. Kulkarni, et. al., “ A review of process planning techniques in layered manufacturing”, Rapid Prototyp. J., vol. 6, 2000, pp. 18–35.

[7] J. Balic, A. Nestler, and G. Schulz, “Prediction and optimization of cutting conditions using neural networks and genetic algorithm”, J. Mech. Eng., Assoc. Mech. Eng. Tech. Slovenia, ISSN 0039-2480, 1999, pp. 192–203.

[8] P. K. Wah, K. G. Murty, A. Joneja, and L. C. Chiu, “Tool path optimization in layered manufacturing”, IEE Trans., vol. 34, 2002; pp. 335–347.

[9] J. Balic and M. Korosec, “Intelligent tool path generation for milling of free surfaces using neural networks”, International Journal of Machine Tools & Manufacture, vol. 42, no. 10, 2002, pp. 1171-1179.

[10] G. Q. Jin, W. D. Li, and L. Gao, “An adaptive process planning approach of rapid prototyping and manufacturing”, Robotics andComputer-Integrated Manufacturing, vol. 29, 2013, pp. 23–38.

[11] D. Misra; V. Sundararajan, and P. K. Wright, “ZigZag tool path generation for sculptured surface finishing”, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 2003.

[12] N. Volpato, R. T. Nakashima, L. C. Galvão, A. O. Barboza, P.F. Benevides, and L. F. Nunes, “Reducing repositioning distances in fused deposition-based processes using optimization algorithms,” Advanced Research in Virtual and Rapid Prototyping. London: CRC Press - Taylor and FrancisGroup, vol. 1. 2013, pp. 417-422.

[13] D. Ohia, L. Koszalka, and A. Kasprzak, “Evolutionary algorithm for solving congestion problem in computer network”, Lecture Notes in Computer Science, Springer, vol. 5711, 2009, pp. 112-121.

[14] L. Koszalka, D. Lisowski, and I. Pozniak-Koszalka, “Comparison of allocation algorithms for mesh structured networks using multistage simulation”, Lecture Notes in Computer Science, Springer, vol. 3984, 2006, pp. 58-67

Partial GMP-CS-LBP Face Recognition using Image Subblocks

Soodeh Nikan and Majid Ahmadi
Electrical and Computer Engineering
University of Windsor
Windsor, ON, Canada
e-mail: nikan@uwindsor.ca and ahmadi@uwindsor.ca

***Abstract*—Proposed face recognition in this paper is a block based approach. Gabor magnitude-phase centrally symmetric local binary pattern (GMP-CS-LBP) has been used to extract directional texture characteristics of face at different spatial frequencies. CS-LBP is applied on the sub-blocks of magnitude and phase responses of Gabor images. Sparse classifier is employed as local classifier to find the sub-blocks class labels. We have evaluated the performance of the proposed algorithm on AR and ORL databases. In real world scenarios, partial face images are available to recognize the identity of an unknown individual. By comparing the recognition accuracy on the recognition results of image sub-blocks, we find the location and size of the most effective face sub-region for identification. Moreover, fusion of image sub-blocks at decision level leads to significantly improved recognition accuracy.**

Keywords-face recognition; block based; effective subregion; partial image.

I. INTRODUCTION

Face recognition is widely used as a biological identification technique which is applied to recognize an unknown individual by analyzing and comparing their facial image to the available database of known identities. It has a wide range of applications such as social networking, border monitoring, access control and law enforcement. The accuracy of face recognition is affected by variation in the appearance of face due to poor illumination, head pose, facial expression, partial occlusion or degradation. In recent years, many identification techniques were proposed in order to increase the accuracy of recognition versus appearance changes. In holistic based approaches, the whole face area is employed to extract features and deciding on the identity label. A robust image representation against occlusion and illumination variation was proposed in [1] using the combination of subspace learning and cosine-based correlation approach which was applied on the orientation of gradient. However, local based techniques by dividing image into sub-regions and fusion of the extracted features or classification results, leads to robustness against variations in the appearance. Local Gabor binary pattern histogram (LGBPH) technique was proposed in [2] where, the local binary pattern (LBP) histograms of sub-blocks of Gabor magnitude images are combined. Different sub-blocks were differentiated in concatenation of features, by assigning a Kullback–Leibler divergence (KLD) weight to the corresponding sub-blocks. In [3], a block-based face recognition technique was proposed by extracting uniform LBP histograms. The results of local nearest neighbour classifiers are combined using an entropy weighted decision fusion to reduce the effect of sub-blocks with less information content. Local phase quantization (LPQ) and multi-scale LBP were applied on the proposed gradient based illumination insensitive representation of image sub-blocks in [4]. Weighted fusion at score and decision level finds the identity of unknown individuals. In [5] the gray values of pixels in image sub-regions were concatenated and class specific multi sub-region based correlation filter bank technique (MS-CFB) was calculated for the training samples and test images. Local polynomial approximation (LPA) filter and directional scale optimization was proposed in [6]. LBP directional images were divided into sub-blocks at four levels. Finally, linear discriminant analysis (LDA) was applied on the concatenation of local histograms at four levels. Nevertheless, some facial areas which contain non-discriminative information can be excluded in the recognition process and computational complexity is reduced by analyzing fewer image sub-blocks instead of the whole face area. This technique is very effective when some parts of the face are occluded by an external object. In some application such as images acquired by surveillance cameras, only a small amount of discriminative information in a partial image of the face is available. We need to find the most effective sub-image to identify an unknown individual whose face is partially covered. The proposed approach in [7] addresses partial face recognition using an alignment-free combination of multi-keypoint descriptors (MKD) and sparse representation-based classification (SRC). A set of MKDs were applied on images in the gallery set and a partial probe image was represented as a sparse linear combination of gallery dictionary.

In this paper, the image is divided into sub-blocks and the proposed face recognition technique which is shown in Fig 1, is applied on local areas. The size and location of the most effective area of the face in identification process has been investigated through the experimental results. We proposed Gabor magnitude-phase centrally symmetric local binary pattern (GMP-CS-LBP) technique as feature extractor based on the symmetry in a local area around image pixels [8]. In order to include the magnitude and phase information of local characteristics of face which are insensitive against appearance changes, we have applied texture descriptor on the magnitude and phase responses of Gabor images. The extracted features are concatenated for each image sub-block.

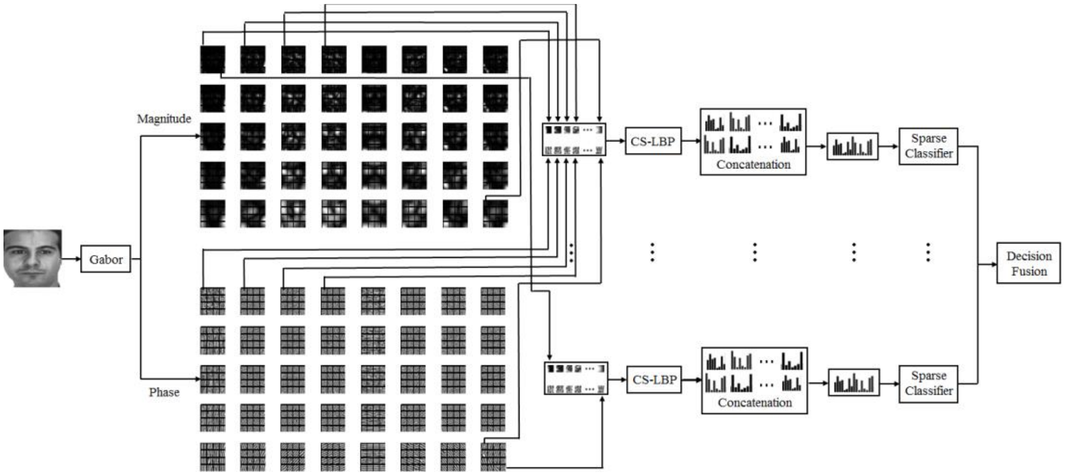


Figure 1. Block diagram of the proposed face recognition technique.

Sparse classifier is employed on image sub-regions to find the local class labels. Majority voting (MV) combines local decisions.

The rest of paper is organized as follows. In Section II, the configuration of feature extraction technique is explained in detail. Section III describes the classification approach. Section IV provides the experimental results. The paper is concluded in Section V.

II. Feature Extraction

The proposed GMP-CS-LBP feature extraction in this paper is the fusion of magnitude and phase information of Gabor coefficients. Configuration of the proposed feature extraction technique is shown in Fig 1.

A. Gabor Filter

Gabor filter extracts the characteristics of signal at different scales and orientations which resembles the mammalia response of vision cells. In order to acquire directionally selective local properties of a face image at various spatial frequencies which are invariant against appearance changes due to expression and illumination variations, 2-D Gabor filters at S_{max} scales and O_{max} orientaions are convolved by image. Gabor filters are obtained as follows by ranging the spatial scale s from 1 to S_{max} and orientation o from 1 to O_{max} [9, 10],

$$\psi_{s,o}(x,y) = \frac{q_{o,s}^2}{\sigma^2} \,.\, e^{-\left(\frac{z^2 q_{s,o}^2}{2\sigma^2}\right)} . \left[e^{(jzq_{s,o})} - e^{\left(-\frac{\sigma^2}{2}\right)} \right], \quad (1)$$

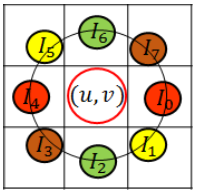
where, $q_{s,o} = q_s \exp(j\theta_o) = [\pi/2(\sqrt{2})^s] \exp(j\pi\, o/8)$ (in this paper we defined 5 scales and 8 orientations). $z = (x,y)$, and $\sigma = 2\pi$ [9, 10]. The magnitude and phase responses of Gabor filtered image are shown in Fig. 1.

B. Centrally Symmetric Local Binary Pattern (CS-LBP)

One of the most powerful local descriptors where the texture information are analysed by comparing the intensity value of local texture in a small neighbourhood and supress the monotonic offset of neighbour pixels is local binary pattern (LBP) analysis. LBP is very fast technique and easy to execute [8, 10]. In a circular neighbourhood with radius R and P neighbours around each image pixel, we compare the neighbours with the centre pixel and depending on the sign of their difference a 1 or 0 value (for positive difference or negative difference, respectively) is assigned to the corresponding neighbours. Therefore, a P-bit binary pattern is associated with the centre pixel. Thus, for image pixels we have decimal values ranging from 0 to 2^P which are used to construct a histogram of 2^P-bin as the texture features. We can reduce the number of histogram bins which decreases the size of extracted features by employing the symmetry in the local area around each pixel. In centrally symmetric LBP (CS-LBP) technique [8], the centre symmetric pairs of neighbours are compared instead of comparing each of them with the centre, as shown in Fig 2. Therefore, the range of decimal values is reduced to $0 - 2^{(P/2)}$ and the stability of the extracted features against flat texture is increased. The calculation of decimal value associated with the binary patterns is as follows [8],

$$CSLBP_{dec}(u,v) = \sum_{l=0}^{({}^{P}/_{2})-1} F\left(I_l - I_{l+({}^{P}/_{2})}\right) 2^l,$$

$$where \quad F(x) = \begin{cases} 1 & x \geq Th. \\ 0 & otherwise. \end{cases} \quad (2)$$



$$CSLBP_{dec}(u,v) = \{F(I_0 - I_4).2^0 + F(I_1 - I_5).2^1 + F(I_2 - I_6).2^2 + F(I_3 - I_7).2^3\}$$

Figure 2. Calculation of CS-LBP for a pixel at (u,v).

(u, v) is the position of centre pixel and I_l is the intensity value of lth neighbor of the centre. R and P are 1 and 8 in this paper. In order to increase the stability against flat areas, the intensity differences between centre symmetric pairs are compared to a threshold value (Th) greater than 0, which is used as threshold in LBP technique [8]. The value which is assigned as threshold is defined in the following section.

C. Local GMP-CS-LBP Histograms

In order to employ magnitude and phase information simultaneously, CS-LBP technique is applied on the magnitude and phase responses of Gabor images at different scales and orientations. However, the threshold value in (2) is different for comparing magnitude or phase information. Through the exhaustive search, in this paper we employ 0.1 as the magnitude threshold and 90° as phase threshold. Following by calculation of the binary patterns and the corresponding decimal values of image pixels and constructing histograms, the $2^{(\mathrm{P}/2)}$-bin magnitude and phase histograms are concatenated.

Furthermore, to find the most effective sub region of face image on the identification accuracy, we divide Gabor images into rectangular non overlapping sub blocks of $m \times n$ pixels. By concatenating the histograms of magnitude and phase responses of all scales and orientations of Gabor responses, we obtain a histogram of $2^{(\mathrm{P}/2)+1} \times S_{max} \times O_{max}$ bins for each image sub region.

III. Sparse Classification

Local classifiers are based on the sparsest representation of the probe sample using the combination of corresponding gallery samples of the same class label [11]. Image samples which are belonging to the same individual lie on a linear subspace.

$$g = [g_1, g_2, g_3, \dots, g_M]. \qquad (3)$$

$$g_i = \left[f_1^g, f_2^g, f_3^g, \dots, f_N^g\right]. \qquad (4)$$

Where, g is gallery dictionary which is including all gallery samples in the database. g_k is the matrix of k^{th} class of subject which consists of gallery feature vectors as its columns (f_k^g is the feature vector of the k^{th} sample in g_k), where M and N are the number of classes and gallery samples per class, respectively. Therefore, using the matrix of gallery dictionary and a coefficient vector we can define the feature vector of a probe sample as a linear combination as follows [11],

$$f_i^p = g.B. \qquad (5)$$

Where, $B = [0,0,\dots,0, \beta_1^k, \beta_2^k, \dots, \beta_N^k, 0,0,\dots,0]$ and $\beta_j^k$is the j^{th} coefficient corresponding to the k^{th} class. The sparsest representation of probe sample can be achieved, if only the coefficients associated with class label of the probe sample are non-zero. Those coefficients are calculated using the l_1-norm solution of equation (5) and the identity label of the probe sample as follows [11].

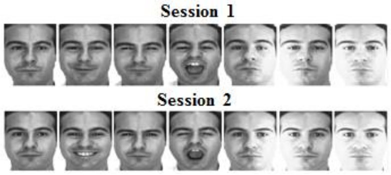


Figure 3. Sample images of one subject in AR database.

$$(l_1): \quad \hat{B}_1 = argmin\|B\|_1 \qquad while \;\; f^p = g.B. \qquad (6)$$

IV. Experimental Results

In order to evaluate the performance of proposed face recognition technique and effectiveness of face image sub blocks on the recognition accuracy, we employ two popular databases and apply the identification algorithm on the 128×128 pixel images in the databases.

A. AR Database

AR face database includes 2600 images of 100 individuals (50 men and 50 women) [12]. Each subject has 26 images taken at two different sessions in two weeks (13 images per session). The images in the database are affected by illumination variation, facial expression and partial occlusion. We have employed non-occluded images in session 1 as gallery set and non-occluded images in session 2 with appearance changes in different time as probe set. Sample images of one subject in AR database are shown in Fig. 3.

B. ORL Database

Olivetti research lab (ORL) database consists of 40 individuals with 10 images per subject and appearance variation due to illumination changes, different time of acquiring image, facial expressions (open/close eyes and smiling/not smiling), up to 20 degree tilting and scales [13]. We randomly used 5 samples per individual in the gallery set and the remaining 5 images per subject in the probe set. Thus, we have 200 images per set. Fig. 4 shows of gallery and probe image samples of one individual in ORL database.

C. Partial Recognition Based on the Image Subblocks

In this experiment, we employ the proposed face recognition algorithm using an image sub-block at different locations and sizes. In order to find the effective size of selected sub-block, we find the accuracy of face recognition versus block size which is shown in Fig. 5. It is shown that for both databases, block size 32×16 pixels leads the highest recognition accuracy. The location of the sub-block is near the eye area. Fig. 6 shows the selected subregion for AR and ORL databases.

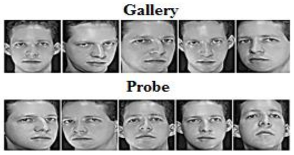


Figure 4. Sample images of one subject in AR database.

D. Decision Fusion for Selected Size of Subblock

Based on the results of previous section, the highest recognition accuracy is obtained at the block size of 32x16 pixels for both face databases. In this experiment, we employ the most effective block size and apply majority voting scheme by adding up the votes of local classification results of image sub-blocks and finding the class label with maximum vote as the final decision. The result of sub-blocks fusion is shown in Table 1 and compared to the accuracy of the state of the art techniques which shows the effectiveness of the proposed face recognition technique.

However, by employing the recognition process using only one sub-block of 32×16 pixels rather than the whole image or fusion of local recognition results, the computational complexity is reduced up to $\frac{1}{40}$.

V. CONCLUSION

A block based face recognition technology has been proposed in this paper by dividing the magnitude and phase responses of Gabor filtered images. CS-LBP is applied on image sub-blocks and concatenation of local histograms at different scales and orientations gives the features of image sub-regions. Fusion of local decisions made by applying sparse classifiers, leads to the final decision on the identification of unknown individuals which outperforms the state of the art algorithms. However, evaluating the recognition accuracy of different sub-regions of the face images in AR and ORL databases, gives the size and location of the most effective local area which reduces computational complexity up to 2.5%.

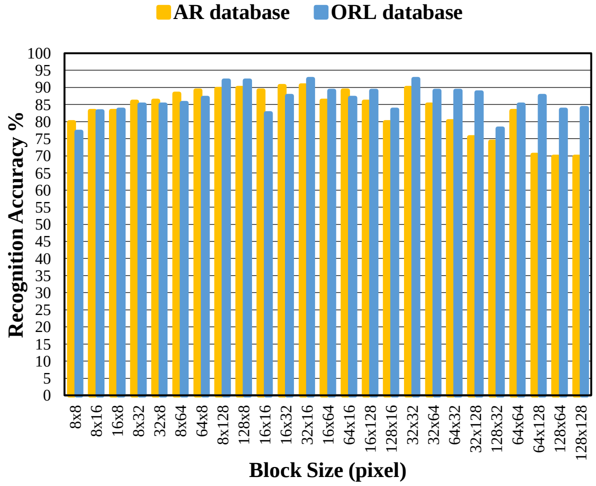


Figure 5. Recognition accuracy (%) of image subblocks for different block sizes.

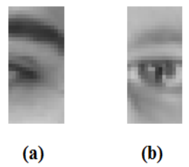


Figure 6. Location of the most effective image subregion: (a) AR database, (b) ORL database.

TABLE I. RECOGNITION ACCURACY (%) OF DIFFERENT ALGORITHMS.

Block Size	Recognition Accuracy (%)	
	AR	***ORL***
LBP+MV [3]	93.42	95.50
CS-LBP+MV	80.42	91.50
LGBFR [4]	99	98
MS-CFB [5]	95	-
SADTF [6]	-	98.50
LCCR [13]	95.86	98
Proposed Method (Decision Fusion using MV)	99.42	98.50

REFERENCES

[1] G. Tzimiropoulos, S. Zafeiriou and M. Pantic, “Subspace learning from image gradient orientations,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, pp. 2454-2466, 2012.

[2] W. Zhang, S. Shan, X. Chen and W. Gao, “Local Gabor binary patterns based on Kullback–Leibler divergence for partially occluded face recognition,” IEEE Signal Processing Letters, vol. 14, pp. 875-878, 2007.

[3] S. Nikan and M. Ahmadi, “Human face recognition under occlusion using LBP and entropy weighted voting,” Proc. International Conference on Pattern Recognition (ICPR12), Tsukuba, November 2012, pp. 1699-1702.

[4] S. Nikan and M. Ahmadi, “Local Gradient-Based Illumination Invariant Face Recognition using LPQ and Multi-Resolution LBP Fusion,” IET Image Processing, 10 pp, 2014, in press.

[5] Y. Yan, H. Wang and D. Suter, “Multi-subregion based correlation filter bank for robust face recognition,” Pattern Recognition, vol. 47, pp. 3487–3501, November 2014.

[6] R. Mehta, J. Yuan and K. Egiazarian, “Face recognition using scale-adaptive directional and textural features,” Pattern Recognition, vol. 47, pp. 1846-1858, 2014.

[7] S. Liao, A. K. Jain and S. Z. Li, “Partial Face Recognition:Alignment-Free Approach,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pp. 1193-1205, May 2013.

[8] M. Heikkila, M. Pietikainen and C. Schmid, “Description of interest regions with local binary patterns,” Pattern Recognition, vol. 42, pp. 425-436, 2009.

[9] Z. Lei, S. Liao, M. Pietikäinen and S. Li, “Face recognition by exploring information jointly in space, scale and orientation,” IEEE Trans. Image Process., vol. 20, pp. 247-256, 2011.

[10] S. Nikan and M. Ahmadi, “Classification fusion of global & local G-CS-LBP features for accurate face recognition,” accepted in: Int. Conf. on Testing and Measurement: Techniques and Applications (TMTA’15), 2015.

[11] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, “Robust face recognition via sparse representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, pp. 210-227, 2009.

[12] A. Martinez and R. Benavente, “The AR face database,” CVC Technical Report, vol. 24, 1998. [Online]. Available from: http:// www2.ece.ohio-state.edu/~aliex/ARdatabase.html.

[13] AT&T Laboratories Cambridge website. [Online]. Available from:http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. 2015.02.17.

[14] X. Peng, L. Zhang, Z. Yi and K. K. Tan, “Learning locality-constrained collaborative representation for robust face recognition,” Pattern Recognition, vol. 47, pp. 2794-2806, September 2014.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-399-5

IPOL - A Domain Specific Language for Image Processing Applications

Christian Hartmann, Marc Reichenbach, Dietmar Fey
Chair of Computer Architecture
Friedrich-Alexander University Erlangen-Nürnberg (FAU),
Martensstr. 3, 91058 Erlangen, Germany
{christian.hartmann,marc.reichenbach,dietmar.fey}@cs.fau.de

***Abstract*—In recent years, the use of image processing systems has increased steadily. However, most of them are very complex and contain several tasks with different complexities which result in varying requirements for computing architectures. Nevertheless, a general processing scheme in every image processing application has a similar structure, called image processing pipeline: (1) capturing an image, (2) pre-processing using local operators, (3) processing with global operators and (4) post-processing using complex operations. Therefore, application-specialized hardware solutions combined in a heterogeneous system are used for image processing. To archive this, finding an optimal heterogeneous hardware architecture to meet the image processing application requirements is the central problem and still unsolved. Instead, engineers use languages like VHDL, Verilog, C/C++ and Cuda for designing such systems. But, these kind of languages are not suitable for system analysis - they provide a hardware specific solution for a specific algorithm. Therefore, a holistic modeling of a complete image processing pipeline, with automatic optimization and assignment to different heterogeneous computing cores is not possible. To overcome this problem, we propose in this paper a new domain specific language, called Image Processing Operator Language (IPOL). This description language contain all needed components hardware components like Sensors, Displays, execution units and software parts like image processing algorithms.**

Keywords*—*DSL, design flow, image processing

I. Introduction

Setting up an embedded application, which uses high performance image processing architectures is a very complex task. In the traditional industrial image processing field, engineers follow Moore's Law and use standard CPUs for their image processing applications. This solution is not resource and energy aware and therefore, does not work for embedded applications. Due to continuous rising requirements on the one hand, and physical limitations of embedded applications concerning area, time and energy, embedded image processing systems become more heterogeneous for fulfilling their functions. For example, in [1] [2] already heterogeneous systems consisting of FPGA, CPU and GPU were used for fast image processing. In general, the usage of an oversized general purpose hardware architecture is not allowed for fast embedded image processing. That leads to the approach of using more application-specialized computing architectures like GPUs or own specialized circuits in FPGAs (Field-Programmable-Gate-Arrays) or ASICs (Application-Specific-Integrated-Circuit).

Although, heterogeneous hardware is available, choosing the right parts (processors or computational units) and programming it, is a hard challenge. Every architecture uses its own language and follows its own programming paradigm. Examples are simple processors (e.g., ARM) with C/C++, FPGAs with VHDL and Verilog, GPUs with CUDA. Moreover, a written, hand crafted solution for one architecture, does not allow the port to another architecture. Therefore, more abstract (and parallel) languages have been developed in the past, which allows an automatic compilation for different architectures. One examples is OpenCL which supports GPUs as well as CPUs. Also a backend for Altera and Xilinx FPGAs were added. But these languages are used for general purpose and they are not specialized for a specific domain, e.g. image processing. Moreover, they support just one target at compile time, which means the code is then executed either at a GPU or FPGA or CPU - they do not work simultaneously together on a problem. Other examples are existing high-level synthesis tools such as Vivado HLS [3] and Intel CoFluent [4]. These tools are often suboptimal and not fully developed. The bad performance of this tools results from the non-specific approach. The tools are too general and do not consider the advantages of image processing architectures. In our design flow we are focused on the image processing domain and use image specialized hardware architectures such as the Full Buffering [5].

Thus, the design flow of mapping complex image processing applications on heterogeneous architectures is a tough challenge and not sufficiently solved in the past. Complex image processing algorithms with different tasks in different granularity have different hardware requirements. These algorithms have to utilize the advantages of specialized hardware architectures for fulfilling the system constraints. Therefore, not only software engineers, but especially hardware engineers, application engineers and system designers are needed, in order to cover all parts of such a system development. Regular programming languages like C/C++ or hardware description languages like VHDL are not applicable for that, because they do not promote a holistic view of the system development. These languages lead to a strict separation of software and hardware and do not consider evaluation techniques for the whole system. With the IPOL language, we want to introduce a holistic domain specific system description language for hardware-software co-design.

This paper is organized as follows. The next section presents the IPOL itself. After that, we present a workflow, how this language can help to find a suitable hardware architecture for a given image processing algorithm. In Section III, an example based on a simulation environment is presented. At the end, we give a conclusion and outlook for future work.

II. Image Processing Operator Language (IPOL)

As described above, with IPOL a whole image processing pipeline could be modeled. Therefore, a detailed description of such a pipeline has to be shown which can then transfered to the language. Normally, an image is acquired using a camera consisting an image sensor. Due to different types of sensors, the data stream from the sensors can vary. Some parameters consist of resolution, bit per pixel, count of pixels per clock cycle, etc. After that, the image has to be processed. This is done via a concatenation of different operators. For example, Gauss, Median and Sobel, can also be parametrized e.g., with the size of the filter mask. Moreover, complex operators like Hough [6] or Fast-Fourier-Transformation have to be executed. After all operators processed the image, an output device has to be specified. That could be for example an display with a parameterizable resolution.

With IPOL it is possible to model such an image processing pipeline in a formal way. Therefore, IPOL is a XML based language. In that language components of an image processing, as described above, exist as a XML-component. In general there are existing 3 types: Sources (e.g. Sensor), Processing (e.g. Operators) and Sinks (e.g. Display). All these components can be extended with content-related parameters (e.g. filter-mask for gauss). In Figure 1 the structure of IPOL is demonstrated.

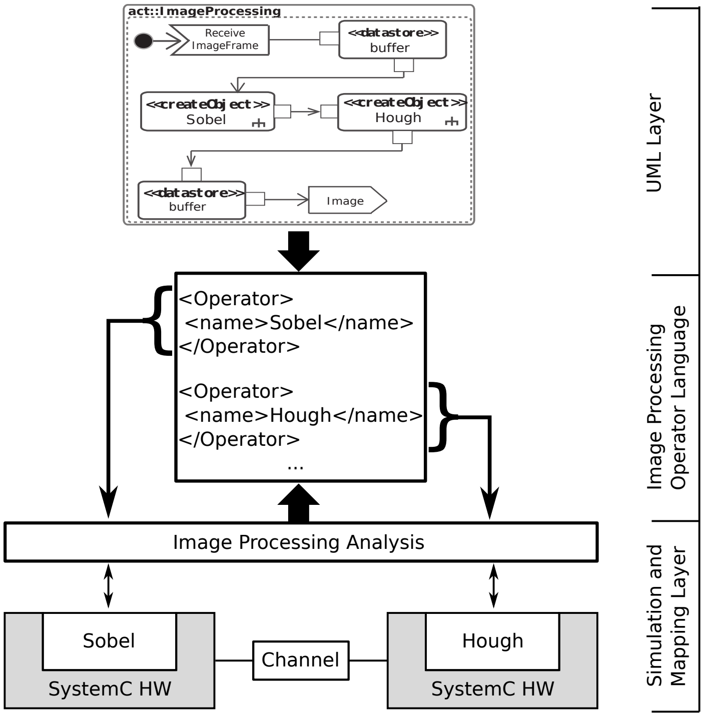
The example shows the whole image processing application named ”operatorchain”, with a sensor, display and two image processing operators (sobel filter and hough transformation). In this example, both image processing operators work on each pixel in the image. Other possibilities could be for example partitioning, where a defined area of pixels is reduced to a feature. The properties ”input_area” and ”output_area” specify the memory access pattern of every operator. An operator with a discrete access window has an other access pattern as an operator with a random access of the whole image. The ”base_calc” block includes a formal description of the algorithm. In this paper ”base_calc” remain omitted. In the example of Figure 1 the image processing operator named Sobel needs a 3×3 neighborhood of picture elements for the calculation of an 1×1 output region. The input and output area of each operator could vary. As shown in Figure 1 an other image processing algorithm, e.g. the Hough Transformation [7], has a different input and output behavior as the Sobel operator. In that example the Hough operator needs only one pixel for a processing step. The ”input_area” of the Sobel is larger, but with a smaller ”output_area” than the Hough operator. For unknown image processing algorithms, our approach provides a library of common image processing operations, like matrix operations. These library could be used for creating new custom image processing algorithms. The known behavior of the common image processing operations makes it possible to analyze these kind of algorithm without knowing the algorithm itself. If an image processing system is specified in such a language, an automatic optimization and derivation to hard- and software components becomes possible. This approach was often discussed in hardware-software co-design publications [8] [9], but investigated only small grained architectures with static solutions and not specialized for image processing applications. In this paper, we propose an additional approach for whole image processing systems, with heterogeneous architectures consisting of complete processing units like CPU cores, GPUs or special architectures on FPGAs. Detailed apsects will be discussed in Section III.

```
<operatorchain>
<globalconstraints>
        <accuracy>20</accuracy>
        <powerconsumption>20</powerconsumption>
        <fps>20</fps>
        ...
</globalconstraints>

<component id="0">
  <type>Sensor</type>
  <res><x>1920</x><y>1080</y></res>
  <pixres>12</pixres>
  <fps>30</fps>
</component>

<component id="1">
        <type>Operator</type>
        <access>each_pixel</access>
        <name>Sobel</name>
        <input_area><x>3</x><y>3</y></input_area>
        <output_area><x>1</x><y>1</y></output_area>
        <base_calc><src>...</src></base_calc>
</component>

<component id="2">
        <type>Operator</type>
        <access>each_pixel</access>
        <name>Hough</name>
        <input_area><x>1</x><y>1</y></input_area>
        <output_area><x>3000</x><y>3000</y></output_area>
        <base_calc><src>...</src></base_calc>
</component>

<component id="3">
  <type>Display</type>
  <res><x>800</x><y>600</y></res>
  <pixres>12</pixres>
  <fps>60</fps>
</component>

<connections>
  <con><out>0</out><in>1</in></con>
  <con><out>1</out><in>2</in></con>
  <con><out>2</out><in>3</in></con>
</connections>
</operatorchain>
```

Fig. 1. Example of an operator chain in the image processing language

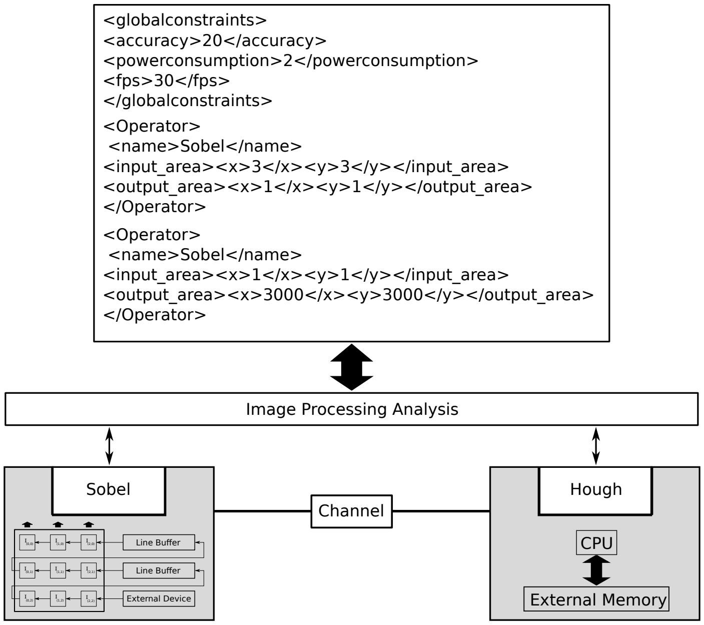
III. System Optimization

With IPOL it is feasible to introduce a new system design workflow for image processing systems. The design flow will cover all the aspects of system design, to consider non-functional properties in the abstract design layers. Therefore, a description of an image processing system utilizing IPOL allows an abstract and programming language independent development. However it is bound to the image processing domain. The new structural features provide the basis for later mapping into concrete hardware and tracing back hardware features to the UML. Figure 2 shows the concept of an image processing design flow.



```
<Operator>
 <name>Sobel</name>
</Operator>

<Operator>
 <name>Hough</name>
</Operator>
...
```

Fig. 2. Image processing design flow: The design flow enables the possibility to create an image processing application using UML.

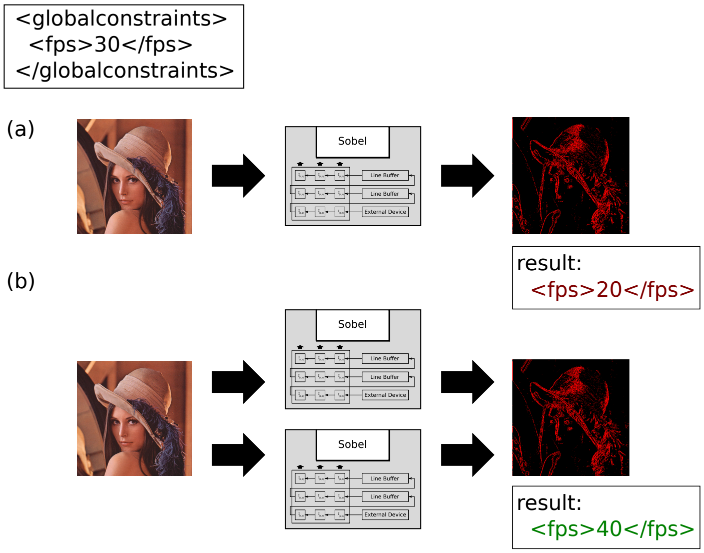


```
<globalconstraints>
<accuracy>20</accuracy>
<powerconsumption>2</powerconsumption>
<fps>30</fps>
</globalconstraints>
<Operator>
 <name>Sobel</name>
<input_area><x>3</x><y>3</y></input_area>
<output_area><x>1</x><y>1</y></output_area>
</Operator>
<Operator>
 <name>Sobel</name>
<input_area><x>1</x><y>1</y></input_area>
<output_area><x>3000</x><y>3000</y></output_area>
</Operator>
```

Fig. 3. Image processing analysis with an image processing language

It enables the user to develop the image processing application in an abstract layer such as UML. An automated mapping of the image processing operators to the simulation environment with virtual hardware makes a holistic UML-based design approach feasible. Thus the user is able to design an image processing application without detailed knowledge of the underlying hardware architecture. The UML model will be automatically transfered in an executable SystemC simulation. This is done by the rules of the domain-specific language, image processing operator language (IPOL) and controlled by the image processing analysis. A more detailed view on the optimization tool is shown at Figure 3. In contrast to existing approaches the image processing analysis considers the in- and output pattern of the algorithms for an automated mapping on specialized or general hardware.

By the example of Figure 3 the Sobel algorithm has a 3×3 input and a 1×1 output environment. The second algorithm, called Hough has a different data access pattern. This algorithm needs only one input pixel for processing, but a 3000×3000 area for the output. Such memory access pattern influences the image processing analysis regarding the hardware architecture selection. In the example of Figure 3 the Sobel is mapped on the full buffering architecture [1]. Specialized hardware such as full buffering could be used for algorithms with strong limited input and output environments. An algorithm with a random or widely scattered data access pattern is not suitable for that kind of architecture. The Hough algorithm would be mapped on an other architecture, with a good connection to external memory. Because of its large amount of memory in the output area. The memory access pattern would be checked for all algorithms in the image processing application. Additionally to the algorithm behaviour, domain specific requirements could be defined, shown at Figure 3. They are called non-functional properties or ”globalconstraints”. The ”globalconstraints” are used as inputs for the system optimization. The image processing analysis uses the non-functional properties to find a suitable architecture for fulfilling the constraints by using the least resources. In the example of Figure 3 the image processing analysis reads the system requirements : $accurarcy = 20$, $powerconsumption = 2$ and $fps = 30$ and proposes a system architcture. This means if the image processing system makes 20 frames per second (fps) with the configuration of one, 40 fps with two and 60 fps with three full buffering processing elements. The system will select two processing elements for fulfilling the requirements of $fps = 30$. A graphical representation of the optimization example is shown at Figure 4.



```
<globalconstraints>
  <fps>30</fps>
</globalconstraints>
```

Fig. 4. Image processing analysis: Global constraints and hardware mapping

ISBN: 978-1-61208-399-5

It shows that the architecture with one processing element do not perform the 30 fps. An architecture with two processing elements passes the test. This optimization will be done for all algorithms and all ”globalconstraints”. Once the static mapping is done the dynamic SystemC simulation run the image processing application with the selected hardware for testing the system configuration. Depending on the simulation result, the system will be mapped on real hardware or has to make an other iteration of the optimization. By a new optimization step the simulation results serve as input for the image processing analysis.

IV. Conclusion and Future Work

In this paper, a new concept and first steps of the underlying methodology have been introduced for modeling and implementing complex image processing systems with a holistic top down approach on heterogeneous computer architectures. The approach covers all layers from abstract UML to a domain-specific language to the executable specification with virtual hardware down to real hardware. In one of our next research steps, we are going to design a connection to the Open Virtual Platform (OVP) [10] and SoClib [11]. That will help developers to include the simulation results for specific processor architectures in their model, without the need of possessing that processor in physical.

Acknowledgment

This work was financially supported by the Research Training Group 1773 ”Heterogeneous Image Systems”, funded by the German Research Foundation (DFG).

References

[1] M. Schmidt, M. Reichenbach, and D. Fey, “A Smart Camera Processing Pipeline for Image Applications Utilizing Marching Pixels,” in *Signal and Image Processing : An International Journal (SIPIJ)*. SIPIJ, 2011, pp. 137–156.

[2] M. Reichenbach, R. Seidler, and D. Fey, “Heterogeneous Computer Architectures: An Image Processing Pipeline for Optical Metrology,” in *Proceedings of ReConFig*. IEEE, 2012, pp. 1–8.

[3] “Vivado HLS,” April 2015. [Online]. Available: http://xilinx.com

[4] “Intel Cofluent,” April 2015. [Online]. Available: http://intel.com

[5] M. Schmidt, M. Reichenbach, and D. Fey, “A generic vhdl template for 2d stencil code applications on fpgas,” in *International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW)*. IEEE, October 2012, pp. 180–187.

[6] W. Burger and M. Burge, *Principles of Digital Image Processing*. Springer, 2009.

[7] J. C. Russ, *The Image Processing Handbook*. Crc Pr Inc, 2011.

[8] F. Mischkalla, D. He, and W. Mueller, “Closing the Gap between UML-based Modeling, Simulation and Synthesis of Combined HW/SW Systems,” in *Design, Automation and Test in Europe (DATE)*, 2010.

[9] A. Fidjeland and W. Luk, “Archlog: High-Level Synthesis of Reconfigurable Multiprocessors for Logic Programming,” in *Field Programmable Logic and Applications, 2006. FPL ’06*. IEEE, 2011, pp. 1–6.

[10] “Open Virtual Platform,” April 2015. [Online]. Available: http://www.ovpworld.org/

[11] “SoClib,” April 2015. [Online]. Available: http://soclib.fr

Automatic Lighting Control System and Architecture Using Ambient Light Sensor

Hyun-Chul Kang, Jung-Sik Sung, Seong-Hee Park, Hyun-Joo Kang, Jong-Woo Choi, Tae-Gyu Kang
Electronics and Telecommunications Research Institute
138 Gajeongno, Yuseong-gu, Daejoen, KOREA
Email :{kauni, jssung, pshee, hjkang,jwchoi,tgkang}@etri.re.kr

Abstract — **In this paper, we propose an automatic lighting control system and architecture using ambient light sensor, which is built in a lighting device. The proposed method can predict user preference lux value on the floor by referencing measured lux value on the ceiling surface with ambient light sensor. We also implement real time lighting automatic lighting control system using the ambient light sensor. An optimized lighting control service can be provided to users in the home network environments.**

Keywords- automatic , lighting control, ambient light sensor.

I. INTRODUCTION

According to the spread of LED (Light Emitting Diode) lighting devices, there have been many researches of lighting control system using various sensors for energy saving in office or home environments. Nowadays, it is possible to design and realize the intelligent lighting control system along with the increasing maturity of computer technology, network technology, control technology and embedded system [1]. The lighting system is one of the largest single consuming units, which accounts up to 26% of the total energy consumption of an inefficient building [2].

Also, it is not easy to adjust the optimal illumination environment for the office worker. The lighting control system, which provides most suitable brightness for each office worker is necessary by using ceiling lighting fixtures [3]. This paper proposes automatic lighting control mechanism with ambient light sensor built-in a lighting device for the efficiency of user-oriented home environment. Section II presents the automatic lighting control system architecture for lighting control using ambient light sensor. Section III describes the simulation result of data analysis and mechanism of the automatic lighting control system. Section IV provides some concluding remarks regarding our proposal and future work.

II. AUTOMATIC LIGHTING CONTROL SYSTEM ARCHITECTURE

The automatic lighting control system architecture includes seven blocks. Figure 1 shows automation lighting system architecture for the optimized lighting control. External interface block performs function of communication with the application server or terminal.

The sampling block collects the initial sampling data of ambient light sensor when the dimming level of the lighting is changed from 1 to 100%.

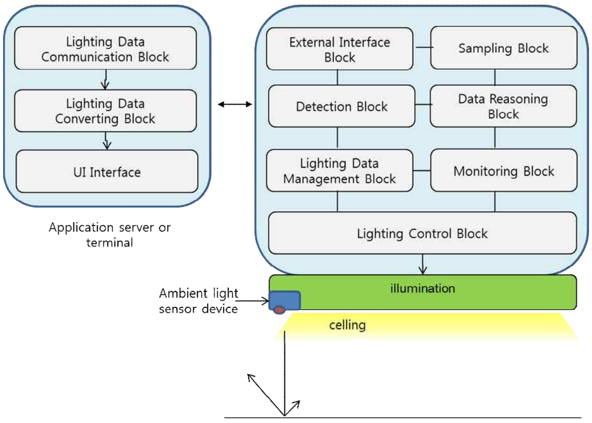

Lighting Data Communication Block
Lighting Data Converting Block
UI Interface
Application server or terminal
External Interface Block
Sampling Block
Detection Block
Data Reasoning Block
Lighting Data Management Block
Monitoring Block
Lighting Control Block
illumination
Ambient light sensor device
celling

Fig. 1. Automatic lighting control system architecture

The sampling data is used as the basis data of the self-control mechanism. The data detection block collects raw sensor data and converts the extracted sensor information into valid ambient light sensor data. The data reasoning block can infer the floor surface data based on the ceiling surface data of the built-in ambient light sensor in the lighting fixture. The lighting data management block performs flow management of lighting data through application server or terminal. The monitoring block performs ambient light sensor data monitoring periodically. The lighting control block performs the optimization of illumination and controls the lighting device according to the service environment. The position of the ambient light sensor device is attached to the illumination.

III. AMBIENTL LIGHT SENSOR DATA ANALYSIS AND AUTOMATION LIGHTING CONTROL MECHANISM

In the simulation, the ambient light sensor is connected to the processor board (Micro Controller Unit) through I2C (Inter-Integrated Circuit) communication. The 10W~50W lighting device is implemented and ambient light sensor is embedded in the lighting device. The simulation environment of illumination is shown in Figure 2. The size of the lighting space is about 60cm * 60cm * 60cm. The illumination value is measured and analyzed by comparing the luminance data of the illuminometer (CL-200A Chroma Meter) and ambient light sensor. A 1 point method is used for the illumination measurements. In the simulation, dimming level of illumination increases from 0% to 100% stepwise.

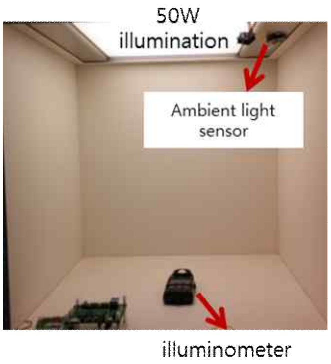
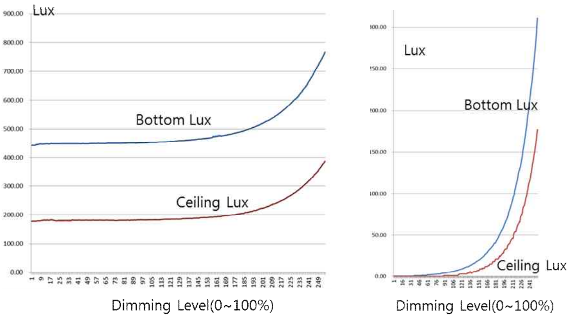


Fig. 2. The Illumination simulation environment.

At the same time, the floor surface data is measured through the illuminometer and the ceiling surface data is collected by ambient light sensor. Equation (1) can be obtained by linear regression analysis through the simulation.

In Figure 3, we see a comparison of the floor surface data with ceiling surface data from the simulation.



(a) When there is some external light source. (b) In the dark room.

Fig. 3. Simmulation results.

$$Y= 1.5599\chi + 169.12. \qquad (1)$$

Y : The floor surface data (Bottom lux)
X : The ceiling surface data (Ceiling lux)

Equation (1) is possible to obtain a formula, such as (2):

$$Y= A\chi + B \qquad (2)$$

Y: The reasoning floor surface data
X: The measured ceiling surface data
A (weight) = (y1-y2)/(s1-s2)
y1: The floor surface data at the time of the minimum dimming level of lighting
y2: The floor surface data at the time of the maximum dimming level of lighting
s1: The ceiling surface sampling data at the time of the minimum dimming level of lighting
s2: the ceiling surface sampling data at the time of the maximum dimming level of lighting
B (Constant) = ((y2-y1) * s2)/ (s1-s2)) + y2

Equation (2) predicts user preference (target) lux value on the floor by referencing measured lux value on the ceiling surface. In Figure 4 shows automation lighting control mechanism and architecture using ambient light sensors.

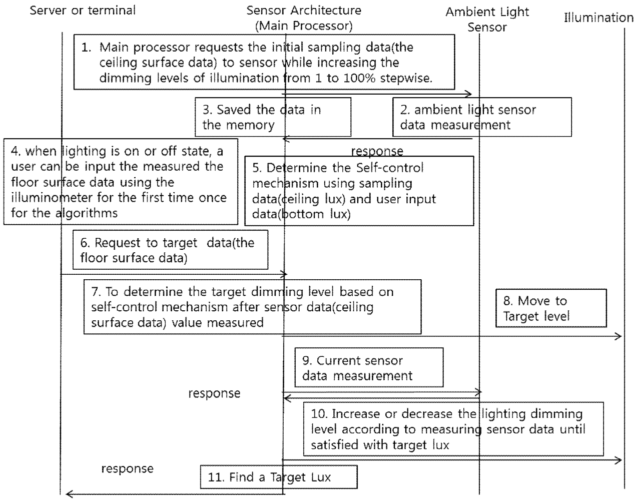

Server or terminal
Sensor Architecture (Main Processor)
Ambient Light Sensor
Illumination
1. Main processor requests the initial sampling data(the ceiling surface data) to sensor while increasing the dimming levels of illumination from 1 to 100% stepwise.
3. Saved the data in the memory
2. ambient light sensor data measurement
response
4. when lighting is on or off state, a user can be input the measured the floor surface data using the illuminometer for the first time once for the algorithms
5. Determine the Self-control mechanism using sampling data(ceiling lux) and user input data(bottom lux)
6. Request to target data(the floor surface data)
7. To determine the target dimming level based on self-control mechanism after sensor data(ceiling surface data) value measured
8. Move to Target level
9. Current sensor data measurement
response
10. Increase or decrease the lighting dimming level according to measuring sensor data until satisfied with target lux
response
11. Find a Target Lux

Fig. 4. The mechanism of the automatic lighting control system.

The user is able to maintain the lighting environment in real time through automation lighting control mechanism.

IV. Conclusion and Future Work

We propose automation lighting control system and architecture using ambient light sensors in the home environments. In this paper, the ceiling lux data can be converted to the floor surface lux data, which is set by user automatically. Also, the dimming level of illumination can be switched to the target dimming level quickly by using an automatic lighting control algorithm. In the future, we will need to study the simulation and light data analysis of algorithms in the various spaces.

Acknowledgment

"This work was supported by R&D program of KEIT, [10049014 The development of MICE-LED convergence content and interactive media-art of tele-Screen], [10042947, A development of LED system lighting engine module with compact sized data communication modules and driver IC/Processor control parts based on multi-sensor],"

References

[1] L. Deng-feng, B. Yun-ting, W. He, and L. Hu, "Design of intelligent lighting control system" Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2011 IEEE International Conference ,pp.134-137, 20-23 March.

[2] F. Kaku, et al. "Construction of Intelligent Lighting System Providing Desired Illuminance Distributions in Actual Office Environment", Artifical Intelligence and Soft Computing Lecture Notes in Computer Science Volume 6114, 2010, pp 451-46

[3] T. P. Huynh, Y. K. Tan, and K. J. Tseng, "Energy-aware wireless sensor network with ambient intelligence for smart LED(Light Emitting Diode) lighting system control", IECON 2011, pp2923 – 2928, Nov. 2011.

Robustness Analysis for Indoor Lighting Systems

An Application of Model Checking in Large-Scale Distributed Control Systems

Richard Doornbos, Jacques Verriet
TNO-ESI
Eindhoven, Netherlands
e-mail:{richard.doornbos, jacques.verriet}@tno.nl

Mark Verberkt
Philips Lighting
Eindhoven, Netherlands
e-mail: mark.verberkt@philips.com

***Abstract*—Modern lighting systems are configurable systems-of-systems that have to operate in an environment that they cannot fully control. These systems have to guarantee the continuation of their functionality regardless of the events in their environment. As testing and simulation are not able to identify all possible interactions of a lighting system and its environment we propose a model checking approach to analyze a lighting system's robustness. To allow easy integration in lighting system development, the approach uses the same configuration options as the lighting systems under study. We apply our approach to an office lighting system and show how model checking can be used to analyze the robustness against network failures and to investigate communication protocols to improve system robustness.**

Keywords-distributed systems; system robustness; model checking; Uppaal; indoor lighting systems.

I. Introduction

Many believe that current trends like Internet of Things (IoT) will only thrive when a well-established industry will take them up. This industry could well be the lighting industry, which is going through a number of paradigm shifts: from traditional light sources (incandescent, fluorescent lamps) to LED (light emitting diode), from simple light sources to intelligent, networked, multi-sensor luminaires, and from individually controlled light points to large-scale distributed control systems. These major changes fit well to the IoT idea of having many internet-connected sensor/actuator nodes distributed over an area of interest, e.g., an office building.

A lighting system for an office building is a prime example of a complex system: it is a large-scale distributed system, containing thousands of sensor and actuator components, which exhibit event-based behavior. The system can have very many configurations, but is comprised of only a limited number of types of components. Typical for modern systems is the complicating factor that it has to cooperate with other systems (heating/cooling, network, power, security), sometimes leading to conflicting requirements.

A. Related Work

In this paper, we present a model checking approach to analyze the robustness of large-scale indoor lighting systems to (erroneous) events in its environment. Robustness is the ability of a system to continue to operate correctly across a wide range of operational conditions, and fail gracefully outside of that range [7].

To the best of our knowledge, model checking has not been applied to large-scale lighting systems. There are many examples of other (industrial) systems that have been analyzed using model checking. Examples include elevator control systems [9] and railway interlockings [12].

In another example, van den Berg et al. [11] use a domain-specific language for specifying medical imaging systems. This domain-specific language is translated into Uppaal [10] for performance analysis. The output of this analysis in translated in information understandable to system designers: analysis results are transformed into lower and upper bounds of system response times. Hendriks et al. [8] have applied a similar approach to create optimal schedules of a wafer scanner.

Similar results have been achieved using MechatronicUML, an Eclipse-based tool suite for the design of cyber-physical systems [2]. It comprises a modeling language and a development process. To validate software correctness, Gerking [3] has developed transformations from MechatronicUML to Uppaal [10] and vice versa: a MechatronicUML design is transformed into an Uppaal model and counterexamples identified by Uppaal are translated back into the MechatronicUML language. This allows system designers to formally validate their system designs without knowledge of Uppaal's timed automata formalism.

Combemale et al. [1] present a formal approach to tracing back analysis results. Their approach requires an input language with an operational semantics definable as finitely-branching transition systems; it transforms analysis results back to the syntax and operational semantics of a domain-specific input language. Input for their approach is a formal relation between the states of the input language and those of the target language. They illustrate their approach using a timed process modeling language as input language and a timed Petri net language as analysis language.

The systems for which formal analysis and back transformations have been used are quite different from the types of systems that we consider in this paper. The basis of our approach is the configurability of lighting systems: a huge variety of lighting systems can be constructed from a few configurable component types. This also holds for logistic control systems. Verriet et al. [13] have shown how

ISBN: 978-1-61208-399-5

warehouse controllers can be configured using domain-specific tooling. A controller can be configured by selecting the appropriate planning and scheduling components and instantiating their behaviors from a library of configurable protocols. This tooling allows warehouse designers to specify a warehouse control system without knowledge of implementation details. A back transformation is not required, because the generated controller provides feedback in terms understandable to the warehouse designer.

B. *Outline*

The paper is organized as follows. Section II explains the robustness challenges in the design of lighting systems. The indoor lighting case is introduced in Section III. Section IV describes the lighting system model, which is used for the robustness analysis described in Section V. Section VI describes the validation of the model by coupling the model to a test setup. In Section VII, we present strategies to improve the robustness of lighting systems against failures in the communication network. Section VIII reflects on the modeling approach and its industrial applicability and describes future work. Section IX summarizes the paper.

II. Problem Statement

The paradigm shifts mentioned in Section I lead to many technical and business challenges. The technical challenges can be summarized in non-functional aspects, such as interoperability (cooperation/collaboration between components from different vendors), availability and robustness (correct lighting behavior at all times), and security (system integrity and user privacy).

A. *Goal*

For a lighting company, it is crucially important to guarantee correct behavior of a lighting system. In this paper, we use a model-based approach to address this challenge. In particular, we apply model checking to assess the robustness of a lighting system's distributed control system.

Our goal to identify robustness issues in a lighting system has led to a two-step approach. The first step focuses on modeling normal system behavior and identifying the system's reactions to events in its environment. These events include normal events like occupancy detection events, but also failures, such as loss or delay of messages. The second step involves finding improvements to make the system less sensitive to undesirable environmental events.

B. *Approach*

Understanding and predicting the behavior of lighting systems is hard without formal modeling. This is due to the complex interaction of a vast number of parallel processes and events. Simulation provides little chance of identifying (rare) robustness flaws. Since model checking allows even the rarest events to be found, we apply formal modeling and model checking as the main direction for our research. We propose a model checking approach that fits the configurability of distributed lighting systems. We require therefore that a large variety of system models can be configured from a small set of model elements in the same manner that many lighting systems can be configured using a small set of component types.

Because formal modeling skills are not commonly available in industry, we propose an approach where a system configuration is converted into a formal model that is used to analyze a lighting system. An important aspect of our work is the translation of the model checking results back to the lighting system domain.

We aim to eventually hide the complexity of the formal modeling tools completely by adaptation of tooling so that it can easily be integrated in an industrial way of working.

III. Case Description

Our robustness analysis is instigated by a real-world application: an office lighting system being developed by Philips Lighting. This office lighting system provides a complete lighting solution in office spaces (cell office, open plan) and central spaces, such as corridors, lounges and entrance areas. The system is typically deployed in an office building and cooperates with other systems, such as HVAC (heating, ventilation, and air conditioning), security, network and power systems. The system comprises networked luminaires with LED-based light sources, a set of sensors (occupancy, luminosity, etc.), and a microcontroller. The number of luminaires in a building is in the order of thousands; typically half of them have a set of sensors.

The office worker, i.e., a lighting system user, has control over the lighting behavior of an area via a button panel (and via a mobile app, but this is not considered in this paper). A button panel allows the selection of a number of predefined lighting settings (relax, concentrate, presentation, etc.), called *presets*. Occupancy sensors provide automatic switching on/off behavior of the lights in an area. Another feature is daylight regulation, which uses the amount of available daylight for setting the light intensity to a constant level. Other features like linking rooms to corridors (keeping the corridor lights on as long as neighboring rooms are occupied, etc.) are not considered in this paper.

System control is distributed in the sense that each luminaire has a controller that exchanges control and synchronization messages with other luminaire controllers. This paper focuses on robustness issues to understand the impact of a change in the technology for message transport. In particular, we investigate the robustness issues when changing from an RS485-based system to an IP-based system, either using Ethernet or wireless. These issues are of extreme importance for Philips Lighting as the possible issues are usually immediately visible to the users. For example, a delayed response to a button press (poor responsiveness) and a single 'dark' luminaire in a 'lit' room (inconsistency) are very noticeable.

The technology transition leads to a more cost-effective solution as the building's existing network can be used. A second important improvement is the use of Power over Ethernet (PoE) technology that not only leads to discarding the power lines for each luminaire, but it also allows installation by less expensive personnel. Using the existing network, however, entails that the network has become part of the lighting system's environment. There are risks involved in migrating from an internal to an external network. In particular, there is no control of the environment: a high network load may cause loss or delay of messages, and the start-up behavior of routers and switches may influence the lighting system behavior.

IV. LIGHTING SYSTEM MODEL

In this section, we present the models that we have used to analyze the robustness of indoor lighting systems. These models are based on the observation that there are a few basic events in a lighting system:

- An *occupancy event* is triggered by the detection of occupancy by a sensor in an area and causes a change of the area's preset.
- A *vacancy event* is triggered by the absence of area occupancy for a certain time period, called the *hold time*, and causes a change of the area's preset.
- A *button event* is triggered by the press of a button in an area and causes a change of the area's preset.

Using these events, one can configure a controller for an area. This configuration involves selecting the appropriate parameters for the events: the presets, the *dwell times* (a period without preset change), and the hold times.

This will be illustrated using a simple example, which includes all three events. Consider a workspace within an open office. When someone enters the workspace area, the lights switch on to a low light level (of 50%). If the workspace is occupied for more than a dwell time of 15 seconds, the light switches to a medium light level (of 75%). A workspace occupant can manually toggle the light level between medium (75%) and high (100%) by pressing a button on a panel. The light switches off (i.e., 0%) automatically if no occupancy has been detected for a hold time of 10 minutes.

The workspace example involves four presets with seven transitions between them. The presets and the transitions are visualized in Figure 1.

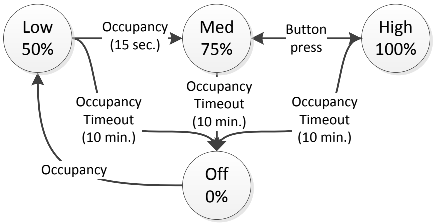


Figure 1. Workspace preset transitions.

Based on the parameterized events, we have created an Uppaal model that can be configured in the same manner as the events. The model consists of two main elements, a controller model and an environment model. These are described in the following subsections.

A. Controller Model

The controller model contains a parameterized timed automaton for each of the events described earlier. The values of the automata's parameters for the workspace example in Figure 1 are given between brackets.

- The *occupancy automaton* has four parameters: a controller id, an active preset (*Off*/*Low*), a dwell time (0/15 seconds), and a new preset (*Low*/*Med*).
- The *vacancy automaton* has four parameters: a controller id, an active preset (*Low*/*Med*/*High*), a hold time (10 minutes), and a new preset (*Off*).
- The *button automaton* has three parameters: a controller id, an active preset (*Med*/*High*), and a new preset (*High*/*Med*).

The occupancy automaton is shown in Figure 2. As explained earlier, the automaton has four parameters: a controller id (*lumId*), an active preset (*p1*), a dwell time (*tDwell*), and a new preset (*p2*). The Uppaal automaton consists of three *locations* (denoted with a circle symbol) connected by five *edges* (denoted with an arrow symbol). Edges are annotated with *selections* (e.g., *p: preset_t*), *guards* (e.g., *sensor[lumId]*), *synchronizations* (e.g., *PB[lumId][p])*, and *updates* (e.g., *t=0*) [10].

The automaton's initial location (indicated with a circle in the location symbol) is the *Off* state. If the corresponding luminaire has a sensor and its power supply gets enabled (channel *powerOn*), it changes to the *On* state. It then resets its internal clock *t*, which determines the time since the last preset change. If occupancy is detected (channel *occupancy*) after the dwell time *tDwell* has elapsed, it communicates a preset change via channel *PLocal* and resets clock *t*. Clock *t* is also reset if a preset change is reported via channel *PB*. If the luminaire loses power (channel *powerOff)*, it changes back to the *Off* state.

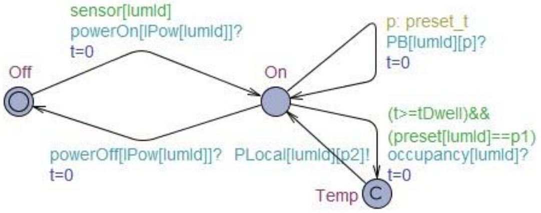


Figure 2. Occupancy automaton.

The event automata receive triggers from the system's environment (see Section IV-B) and update the internal states of the corresponding controllers accordingly. In our model, a controller's internal state includes the active preset and a number of clock values. The clock values are updated by the event automata; the model includes a separate automaton to update a controller's active preset.

We model a lighting system that is controlled in a distributed manner: one area may have several controllers. To avoid undesired behavior, e.g., a controller recalling preset *Off* while another controller has recently detected occupancy, the internal states of the controllers need to be synchronized. This is done by a synchronization event, for which we have created a fourth parameterized automaton. This synchronization automaton informs the other controllers in the same area of its internal state.

B. Environment Model

As our goal is to analyze lighting systems' robustness against events in their environment, we have created automata that model a lighting system's environment. In particular, we have created timed automata for occupancy generation, power supplies, and network communication.

We will focus on the network automata. These automata model loss and delay of messages in a generic manner, i.e., they are independent of the underlying network technology and protocols. The network automata decouple transmission and reception of messages. A network automaton for communicating presets is shown in Figure 3. The automaton uses two channels: an incoming channel *PA* and an outgoing channel *PB*. If the network automaton receives a message via channel *PA*, then it either forwards it via channel *PB* or discards it. The latter models message loss. Message delays have been modeled in a similar manner.

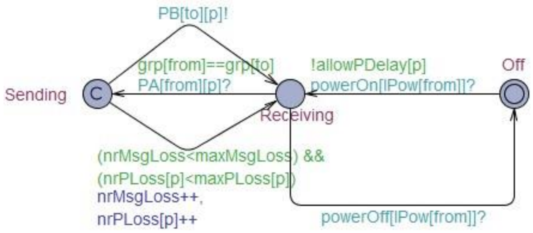


Figure 3. Network automaton.

To fully control loss and delay of messages, the model includes a network automaton for each combination of a source controller, a destination controller, and a preset. This allows a message to be received by any subset of the intended recipients, enabling the analysis of all possible scenarios involving message loss and message delay.

V. ROBUSTNESS ANALYSIS

As explained in Section III, Philips Lighting is considering porting a distributed control concept from an RS485 network that is part of the lighting system, to an IP network that is part of the lighting system's environment. Philips wants to know the influence of an external network on the behavior of their lighting systems.

The CAP theorem shows that a distributed system cannot simultaneously be Consistent, Available, and Partition tolerant [6]. Since high network loads may cause message loss (a form of partitioning), this practically means that inconsistencies between luminaires in one area cannot be avoided or that (part) of the system becomes unavailable, i.e., unresponsive.

This can easily be exemplified; consider the preset transitions of Figure 1 for an area with two luminaires, one of which has an occupancy sensor. Figure 4 shows a Gantt chart for a scenario in which this area ends up in an inconsistent state. The area starts in preset *Off* (black). The occupancy sensor of luminaire 1 detects occupancy (blue) and its controller recalls preset *Low* (cyan event). Fifteen seconds later, this sensor still detects occupancy and preset *Med* is recalled (magenta event). The corresponding message from luminaire 1 to luminaire 2 gets lost. This leads to an inconsistent situation: luminaire 1 is in preset *Med* (light gray) and luminaire 2 in preset *Low* (dark gray). This inconsistency ends when an occupant presses a button and preset *High* (white) is recalled (yellow event).

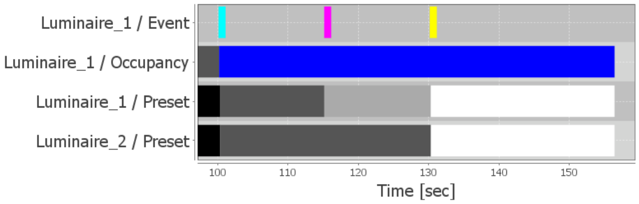


Figure 4. Inconsistency due to message loss.

Section IV has described the automata from which lighting system models can be constructed. The example shown in Figure 1 involves seven event automata per controller. Moreover, all luminaires have an automaton that maintains the active preset. On top of that, there are network automata for each combination of a source luminaire, a destination luminaire, and a preset. For the example area, this involves sixteen network automata.

Using the model elements described in Section IV, we have configured and analyzed a number of lighting systems including a cell office configuration that Philips has used in customer projects. Details regarding these configurations are omitted because of confidentiality.

Uppaal has a requirements specification language in which one can formulate system properties [10]. We have used this language to formulate desired lighting system properties. Combined with a consistency monitoring automaton, we have formulated properties regarding the occurrence and maximum duration of area inconsistencies. With these requirements, we have identified several scenarios in which loss or delay of messages causes an area to end up in an inconsistent situation. The identified message loss scenarios are similar to the one in Figure 4; the inconsistencies caused by message delays are mainly due to out-of-order message reception.

VI. MODEL VALIDATION

As explained in Section V, we have used Uppaal models to identify robustness issues, e.g., scenarios in which an area becomes inconsistent. These formal models have been created from a variety of informal documents and observations of actual systems. To gain confidence in the correctness of the outcome of the robustness analysis, we have coupled our model to a test setup (see Figure 5). This test setup is a realistic, small-scale lighting system composed of commercial products: luminaire controllers, (simulated) sensors, and PoE switches. The simulated sensors allow injecting occupancy events into the system when needed. Message loss and message delays are controlled by an IP bridge. This bridge is realized by two network cards in the PC connecting two separate networks.

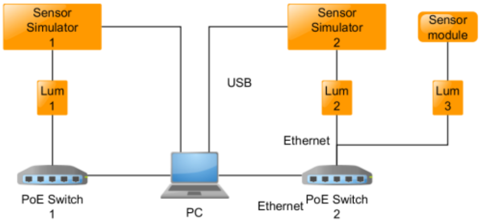


Figure 5. Lighting system test setup.

The Uppaal model and the test setup can both be configured with occupancy, vacancy, button, and synchronization events.

To validate our models, we have created a two-way transformation between the model and the test setup. This transformation is based on a domain-specific language (DSL) that we have created for lighting system scenarios. This DSL has been developed using Xtext technology [4] and the transformations using Xtend [3] and Java.

The test setup allows observing the luminaire controllers and the communication network over time and creates a record expressed in the DSL. A recorded scenario includes occupancy detections, active presets, and output levels. This scenario can be translated into an Uppaal automaton that replays the occupancy detections and checks whether the response of the model corresponds to the response observed in the test setup.

Reversely, the inconsistency scenarios identified using the Uppaal models need to be checked on the test setup. We have developed a transformation that transforms an inconsistency scenario into a script that can be replayed on the test setup to check whether the inconsistency actually occurs. In the same manner, we can check good-weather scenarios.

Using the described two-way transformation, we have successfully validated the Uppaal model. On the one hand, the responses of the model correspond to responses observed on the test setup. On the other hand, all of the inconsistency scenarios identified using our model have been successfully reproduced by filtering or delaying the appropriate network messages.

This gives confidence in the correctness of the model and therefore in a lighting system's behavior in case of loss or delay of messages. The actual probability of message loss and message delay is subject for further study, and involves measurements and estimation of loss rates, and other implementation-specific details.

VII. AUTOMATIC RECOVERY

In Section VI, we have validated that message loss and message delay may lead to periods of inconsistency in a lighting system. Message loss and message delay are rare in practice, but they cannot be avoided by the lighting system, because the communication network is not part of the lighting system. To be robust against message loss, a lighting system needs to cope with these inconsistencies by automatically recovering. This requires the introduction of a recovery strategy. In this section, we analyze such strategies.

We will focus our analysis on lighting systems' recovery capability. In particular, we determine the maximum duration of area inconsistency if a finite number of messages are lost in the communication network. In order to compute this maximum duration, we have extended the model with an area monitoring automaton that continuously checks for inconsistency and starts a clock when it detects area inconsistency.

We have compared three strategies for their recovery capability. All strategies involve repeated synchronization of a controller's internal state. For confidentiality reasons, we omit the details regarding the strategies.

- *Strategy* 1: No additional state synchronization.
- *Strategy* 2: The usage of additional occupancy and vacancy events for state synchronization.
- *Strategy* 3: The usage of new synchronization events for state synchronization.

Table I shows the maximum inconsistency duration for the different strategies, derived using simple argumentation. In this table, N denotes the number of lost messages, T_H equals the maximum hold time of the vacancy events, T_D is the maximum dwell time of the occupancy events, and T_R is the maximum state synchronization time.

TABLE I. MAXIMUM INCONSISTENCY PERIOD DURATION

Strategy	Maximum inconsistency
Strategy 1	$N \cdot \infty$
Strategy 2	$N \cdot (T_H + \max\{T_{R,} T_D\})$
Strategy 3	$N \cdot T_R$

The results in Table I apply to all system configurations that we have analyzed. These include the example in Figure 1 and configurations used by Philips Lighting. The results in Table I show that if no messages are lost, then the lighting systems behave consistently. It also shows that without additional synchronization, message loss can (in theory) lead to infinite periods of inconsistency. Finally, it shows that repeatedly synchronizing the internal state provides a successful manner to recover from area inconsistency; the speed of recovery depends on the strategy.

VIII. DISCUSSION AND FUTURE WORK

In the previous sections, we have presented formal models to analyze the robustness of distributed lighting systems. This section reflects on the modeling approach and its industrial applicability and describes future work.

A. *Industrial Fit*

Application of formal modeling and model checking techniques in industrial practice has always been a challenge, mainly due to the distance between the abstract level of reasoning for formal modeling and the concrete level of reasoning required in product development and realization.

In our work at Philips Lighting, we reduce this distance by matching our models to the actual design and implementation concepts and terminology used by the system architects and engineers. Furthermore, we hide the complexity of the model checking tooling by creating a transformation of configuration information to Uppaal models. This entails the adaptation of our models to the high configurability of the actual system.

The basis of our approach is the configurability of lighting systems: these systems are constructed by selecting and configuring a few component types. In earlier work, we have identified that logistic systems, e.g., warehouses and baggage handling systems, share these characteristics [13]. Our approach can therefore easily be adapted to validate logistic system controllers.

Note that the approach will also have great value in the IoT domain, as there is strong dependency on the good- and bad-weather behavior of the (network) environment.

B. Model and Modeling Experience

The process of creating a formal model from design knowledge and concrete implementations is a challenging one: initially the knowledge has to be acquired from experts or design documentation (which may be hard to find); next the models are created iteratively when the knowledge and insight grows. The large number of features and configurations that are captured in the model quickly leads an unmanageable monolithic model. Therefore, we were forced to decompose the monolithic model into logical parts exhibiting the same behavior. We have created a modular model comprised of parameterized event automata, which fits the configuration tooling of Philips Lighting. This, however, has consequences for scalability. It is a challenge to keep the state space small when having many automata. This scalability problem in a realistic industrial context is a topic for further research.

C. Model Extensions

The current model of the Philips Lighting case captures the control behavior of single-area configurations. The model can be extended with hierarchy of areas (linking the behavior of a corridor to that of connected offices, etc.). We foresee additional scalability challenges in such cases.

Another modeling challenge involves the start-up behavior of system components (including switches, routers, etc.) and the corresponding lighting behavior. This is important in cases of (partial) power failures, or software update scenarios. The challenge lies in the acquisition of detailed knowledge about the component behavior in non-specified situations.

IX. Conclusion

In this paper, we have presented a model checking approach to analyze the robustness of distributed lighting systems. There is a huge variety of lighting systems that can be built from a small set of component types. Our modular and parameterized modeling approach addresses this variety, because it allows the same configuration options as lighting systems. We have shown that our approach is very helpful in identifying robustness issues and in analyzing robustness-improving communication protocols. In particular, we have shown that lighting systems can recover from inconsistent behavior by having their luminaires communicate their internal state periodically. These concrete results are highly appreciated by the development organization of Philips Lighting.

A second benefit of our modular and parameterized approach is its integration in an industrial way of working. Because the models have the same configuration options as the distributed lighting systems, formal models can be generated automatically from the lighting systems' configuration information. We have shown that it is possible to close the loop from formal model to real implementation by having a transformation that translates formal analysis results back into the lighting domain.

References

[1] B. Combemale, L. Gonnord, and V. Rusu: “A generic tool for tracing executions back to a DSML's operational semantics,” in Modelling Foundations and Applications, R.B. France, J.M. Kuester, B. Bordbar and R.F. Paige, Eds. Berlin: Springer, pp. 35-51, 2011.

[2] S. Dziwok et al.: “A tool suite for the model-driven software engineering of cyber-physical systems,” 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE), ACM, November 2014, pp. 715-718, doi:10.1145/2635868.2661665.

[3] Eclipse Foundation: Xtend. [Online]. Available from: http://www.eclipse.org/xtend/ [retrieved: 2015.03.02].

[4] Eclipse Foundation: Xtext. [Online]. Available from: http://www.eclipse.org/Xtext/ [retrieved: 2015.03.02].

[5] C. Gerking: “Transparent Uppaal-based verification of MechatronicUML models,” Master’s thesis, University of Paderborn, May 2013.

[6] S. Gilbert and N. Lynch, “Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services,” ACM SIGACT News, vol. 33, pp. 51-59, June 2002, doi:10.1145/564585.564601.

[7] S.D. Gribble: “Robustness in complex systems,” Eighth Workshop on Hot Topics in Operating Systems, IEEE, May 2001, pp. 21-26, doi:10.1109/HOTOS.2001.990056.

[8] M. Hendriks, B. van den Nieuwelaar, and F. Vaandrager: “Model checker aided design of a controller for a wafer scanner,” International Journal on Software Tools for Technology Transfer, vol. 8, June 2006, pp. 633-647, doi:10.1007/s10009-006-0025-7.

[9] F. Kammüller and S. Preibusch: “An industrial application of symbolic model checking - The TWIN elevator case study,” Informatik - Forschung und Entwicklung, vol. 22, February 2008, pp. 95-108, doi: 10.1007/s00450-007-0032-2.

[10] Uppsala Universitet and Aalborg University: UPPAAL. [Online]. Available from: http://www.uppaal.org/ [retrieved: 2015.03.02].

[11] F. van den Berg, A. Remke, and B.R. Haverkort: “A DSL for performance evaluation of medical imaging systems,” Medical Cyber Physical Systems Workshop 2014, April 2014, pp. 80-93, doi: 10.4230/OASIcs.MCPS.2014.80.

[12] L. van den Berg, P. Strooper, and K. Winter: “Introducing time in an industrial application of model-checking,” in Formal Methods for Industrial Critical Systems, S. Leue and P. Merino, Eds. Berlin: Springer pp. 56-67, 2008.

[13] J. Verriet, H.L. Liang, R. Hamberg, and B. van Wijngaarden: “Model-driven development of logistic systems using domain-specific tooling,” in Complex Systems Design & Management, M. Aiguier, Y. Caseau, D. Krob, and A. Rauzy, Eds. Berlin: Springer, pp. 165-176, 2013.

Smart City Aspects, Services and Application

A communication platform for smart cities

Radovan Novotný, Zdeňka Kuchtová, Radek Kuchta, Jaroslav Kadlec, Radimír Vrba
Central European Institute of Technology
Brno University of Technology
Brno, Czech Republic
e-mail: novotnyr@feec.vutbr.cz

***Abstract*—There are plenty of opportunities for new services by interconnecting physical and virtual worlds with a huge amount of wireless nodes distributed in houses, vehicles, streets, buildings and many other public environments. The first objective of this article was to provide an overview of the smart city concept, and the second goal was to propose a technological solution and communication platform for the development of city services in relation to the improvement of urban systems and services. The diagram of proposed communication platform that is presented could be used as a connection between multiple cities. It is applicable for the development of city services and for the integration and use of rendering and actuating technologies. The proposed communication platform can be used for the improvement of specific urban systems and services and as a connection between multiple cities.**

Keywords* - *smart, city, survey, connection, communication platform, sensor, network, data, service.

I. INTRODUCTION

There is a demand for smarter, effective, efficient and more sustainable cities, pushing the collective intelligence of cities onward, which can improve the ability to forecast and manage urban flows, and integrate the dimensions of the physical, digital and institutional spaces of a regional agglomeration. The expression "smart city" has been used for several years by a number of technology companies and serves as a description for the application of compound systems to integrate the operation of urban infrastructure and services, such as buildings, transportation, electrical and water distribution, and public safety [1]. A smart city can be described as a city that:

- Allows real-world urban data to be collected and analysed by the use of software systems, server substructure, network infrastructure, and client devices [2].
- Implements solutions, with the support of instrumentation and interconnection of sensors, actuators, and mobile devices [3].
- Can combine service production and an intelligent environment, exploits accessible information in its activities and decision making and adopts information flows between the municipality and the urban or business community [4].

The theory of smart cities understood from the perception of technologies and components has some exact properties within the wider cyber, digital, smart, intelligent cities texts [5]. The novelty of this article is summarized in Figure 6 which proposes a connection between multiple cities. The rest of the paper is organized as follows. Section II and III will highlight smart city systems, applications and services, and then in Section IV is presented the role of technology. Section V presents the communication platform that we have proposed. Section VI provides some final conclusions.

II. SMART CITY SYSTEMS AND CITY DATA

As information systems have become prevalent in urban environments they have formed opportunities to capture information that was never previously accessible. Figure 1 provides an overview of the city systems and relevant aspects.

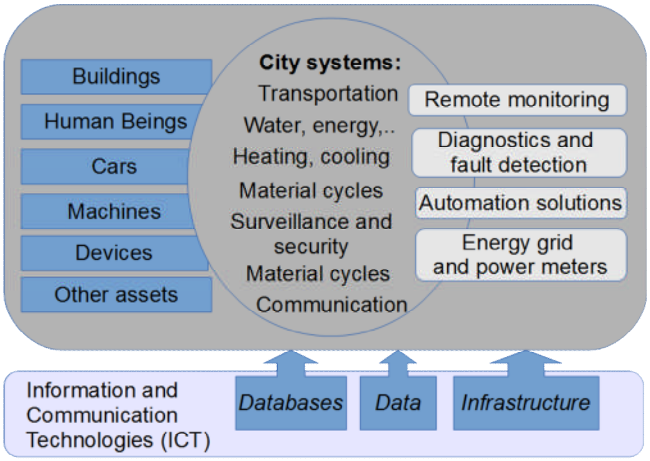


Figure 1. Data and city systems.

This overview is necessary for an understanding of the existing links. Vast amounts of data that describe what happens in the city are available and could be used to create and change intelligent solutions within related areas of e-services application. Knowing what data and what information systems the city has would help with understanding the city systems. But the reality is that "nobody has a comprehensive overall picture of the data and information systems of their city [2].“ Viljanen et al recapitulated that "even the City itself does not have a complete overview on all the information systems it has in its dozens of different departments and public service corporations. [2]" The expansion of both computing power and new algorithms allow this information to be analyzed in near “real-time” in order to provide a base for all developed applications.

The same infrastructure is used 24 hours a day, seven days a week by various stakeholders - citizens, workers, students, researchers, investors or entrepreneurs. Characteristic players in the smart city include municipal leadership, IT and telecommunications companies, utilities, municipality technical services, and grid-infrastructure service providers. Partnerships and strong collaboration strategies and tactics among key stakeholders are required in order to share research and innovation assets, such as emerging Information and Communication Technologies (ICT) tools, methodologies and know-how, experimental technology platforms, and user communities for experimentation on e-service applications and future internet technologies [5].

A. *Smart cities applications and services*

Citizens and other stakeholders expect high quality public services that transform and enhance their daily quality of life. A brief overview of various areas of smart city applications is recapitulated in Figure 2.

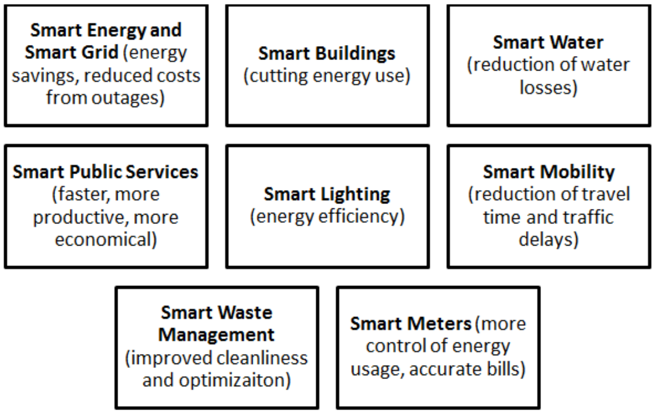

Smart Energy and Smart Grid (energy savings, reduced costs from outages)
Smart Buildings (cutting energy use)
Smart Water (reduction of water losses)
Smart Public Services (faster, more productive, more economical)
Smart Lighting (energy efficiency)
Smart Mobility (reduction of travel time and traffic delays)
Smart Waste Management (improved cleanliness and optimizaiton)
Smart Meters (more control of energy usage, accurate bills)

Figure 2. Areas of smart city applications.

It is clear that the spectrum of application areas is very wide. For example, real-time travel information is essential for applications which let people plan trips on public transportation. The user could have real time information about when the next bus or train is coming. Another example is an application, which collects and distributes real time information about where parking is accessible so drivers can promptly find free spaces. Access to suitable data represents an opportunity for developers to create applications. In this way, stakeholders can access wide online services, with portals for basic information, citizen services, business, and tourism, all based on a common infrastructure.

TABLE I. OVERVIEW OF SMART CITY SERVICES

Application area:	Description:	Examples:
General municipal and business services	Creation of networks between cities and partners, and services realization in order to add value to stakeholders.	- on-line problem solving tools - intelligent shopping - services ordered electronically
Intelligent, sustainable buildings and building management *(smart building)*	Intelligent buildings that contain the advantages that come from integrating communications and building control systems.	- room automation systems - optimized heating, ventilation, and air conditioning
Education, health and social care arena *(smart education)*	Applications that allow improvement of processes undertaken in this area and with better access to services.	- telemedicine monitoring - sharing medical files - tracking systems for elderly people
Energy production and energy efficiency *(smart energy, smart lighting)*	Intelligent electricity system that connects all supply grid (utilities) and demand elements (end users) through an intelligent system.	- lighting controls - smart grid applications - optimize grid performance - provide adherence to environmental rules
Gas, electricity and water smart metering *(smart grid)*	Utility meter that records energy, water or gas usage in real time.	- wireless smart meters -on-line information about consumption
Smart water and waste management *(smart utility)*	Intelligent management of water and sewer system and flow management technology with real time awareness and control.	- intelligent sewer system - rubbish bins real-time monitoring - pressure management
Public safety, security and crime prevention	Anticipate events, respond and preventing them, warn users of dangers, optimize the capacity and response time of emergency services.	- cameras around town - IP video surveillance system emergency signalling
Real-time locating services and geographic information *(smart parking)*	Covering of strategic spatial information needs of people or organizations and realization of service that helps keep track of things.	- location aware applications - identity related services - keep track of cars
Logistics and supply chain *(smart supply chain)*	Synchronizing supply with demand, measuring, monitoring and managing transport and inventory movements or supply chain activities across the city supply chains.	- tracking and inventory control - control and visibility of supply chains - provide adherence to rules and regulations
Mobility and transport *(smart transport)*	Build a real-time and efficient traffic system to optimally use and combine all means of transport (efficiency, delays, and fuel waste and carbon emissions).	- surveillance cameras for transportation - smart parking network - provide adherence to environmental rules and regulations

III. Overview of Smart City Services

There is a wide range of services and applications (See Table 1). These services cover fields, such as transportation (intelligent road networks, connected cars and public transport), public utilities (smart electricity, water and gas distribution), education, health and social care, public safety. Emerging applications and services are extended into diverse fields, such as everyday life of citizens, disaster management, smart buildings, logistics and intelligent procurement.

The applications for this portfolio include implementation for the connected city such as: smart grid, smart home, security, building automation, remote health and wellness monitoring, location aware applications, mobile payments and other machine-to-machine (M2M) applications.

The acronym XaaS (X as a service) refers to any of an increasing number of services provided on-line: everything as a service or anything as a service. The examples of XaaS are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Monitoring as a Service (Maas), Security as a service (SECaaS), Software as a Service (SaaS), and others.

Developed applications are able to supply real-time information and expand the ability to forecast and manage urban flows, and fulfil other functions of the city. Also, they can help to reveal how demands for transportation, water and energy peak in a city and how to take appropriate action and respond. It is important to the city stakeholders that they can collaborate to smooth these peaks and to achieve robustness.

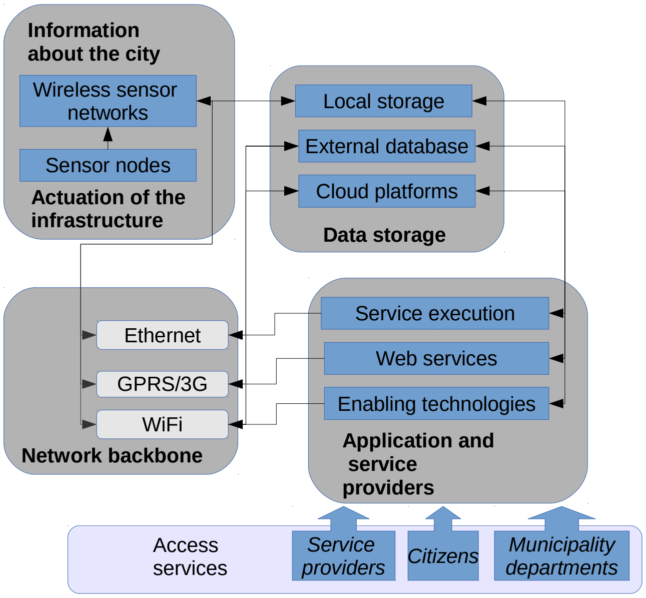


Figure 3. City services and the technological solution.

IV. The Role of Technology

Advances in new technology are employed to improve city applications and services. There are communications, analytical and control technologies that permit transforming the way of doing things while influencing better policy and urban management. It is changing the entire way the service can be solved, combining the ICTs with city infrastructure and shifting the city systems solutions. Thanks to these technologies, there is the capability in the provision of services via digital communication, e.g., interactive services or automating the solution of services. Figure 3 presents city services and the technological solution of services. Data are stored and forwarded by using the network backbone in order for use by service providers and in related applications. Cimmino et al [11] highlighted the role of small cell technology in smart cities – there is the prospect of "increased broadband capabilities, improved flexibility and easy deployment of scalable multi-service network architectures." The article concludes that the integration of broadband personal communications with device-to-device communications and M2M will constitute a significant challenge. The creation of information stuff is not restricted to a particular location, and the resulting products are typically delivered through the network. Smart city services are also available through wireless mobile devices and are enabled by services oriented enterprise architecture including web services, the extensible markup language (XML), and mobilized software applications [15]. Big data to analyze, capture, clean, search, share, store, transfer, and visualize should also be mentioned here.

Smart cities are deploying online services in diverse sectors of cities. An online service, also called Software as a Service (SaaS), is a service delivered by a software application running online and making its facilities accessible to users over the Internet via an interface. The interface could be Hyper Text Markup Language (HTML) obtainable via a standard client, such as web-browser or a web-API (application programming Interface) or by any additional means. It can represent real service that runs on the host (POP, SMTP, HTTP, etc.) or some other kind of metric associated with the host - response to a ping, free disk space, number of logged in users, etc. Eventually, services could be delivered to users through home-based access or mobile access, citywide digital interactive displays, or kiosks. Cloud computing has radically transformed in what way business applications are built and run. Platform as a Service (PaaS) is a way to lease operating systems, storage and network capacity or hardware over the Internet. It is a kind of cloud computing services that deliver a solution stack and a computing platform as a service. Here, online-service users no longer need to own or license the software to run it. Where users need to pay, they are paying for use of the service rather than for owning or licensing the application itself. These innovations have allowed offer more services to more people, to give better access to services with accompanied improvements and innovations.

A. Wireless sensor networks and related technological advances

The aim is to deploy monitoring infrastructure and produce a distributed network of intelligent sensor nodes which can measure many parameters for a more efficient management of the city. Recent advances in wireless sensor networks have been determined by a range of underlying technological advances, primarily progress in MEMS (micro-electro-mechanical system) sensor technology, and innovative ways to manage power consumption. These networks responsible for sensing as well as for the first stages of processing are capable of flexible, low-cost monitoring of a range of environmental parameters and phenomena at very fine levels of spatial and temporal detail.

Smart city solutions proposed for event detection based on Wireless Sensor Networks will be generating important growth in this arena. The wireless sensor network (WSN) consists of a group of heterogeneous and spatially dispersed autonomous sensors deployed in large numbers either inside the phenomenon or very close to it. The WSN is constructed from a large number of "nodes" organized into a cooperative network, where each node is connected to one or several sensors.

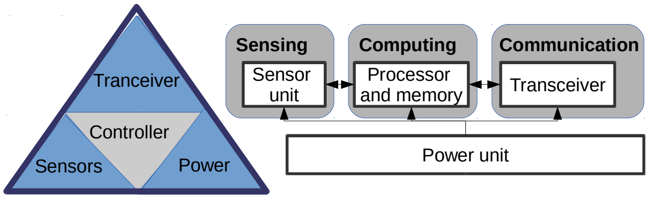


Figure 4. Hardware architecture of a sensor node.

Figure 4 provides a schematic overview of the typical architecture of the sensor node. These sensor nodes have the capability to collect and process data, each node is able to autonomously sense, process, and communicate data about its immediate environment to other nearby nodes and computers. Zheng at al noticed that there are the unique characteristics and constraints for sensor networks: dense node deploymend, battery powered sensor nodes, severe energy, computation, and storage constraints, self-configuration, application specific design requirements, unreliable sensor nodes, frequent topology change, many-to-many traffic pattern, data redundancy, and nonexistence of global addressing scheme [26].

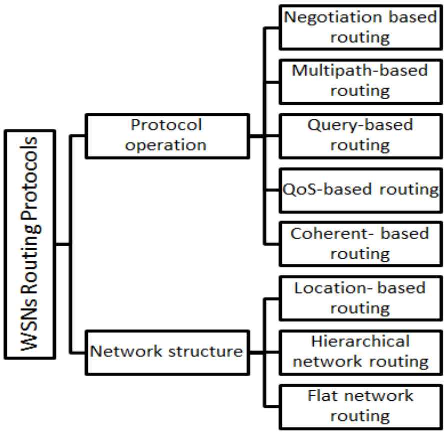


Figure 5. WSNs Routing Protocols [25].

Just as the Internet allows access to digital information anywhere, sensor networks will provide vast arrays of real-time, remote interaction with the physical world (Pinto, 2003). Distributed intelligence from the sensor to the network will become as essential as the Internet - wireless sensor networks give the opportunities for the collation of data which is fit for the purpose supporting the creation of smart cities. Each from a few to several hundreds or even thousands of nodes in the network consists of processing capability given by one or more programmable microcontrollers for controlling node behaviour and processing data.

A lot of research has previously been accompanied into sensor networks, and a comprehensive set of specifications have been completed for the physical layer, link layer, and network layer. The same is true for routing protocols necessary for setting up path or paths from sensor nodes to the data sink. Routing is an important issue and since sensor nodes have limited resources, routing protocols must have a small overhead. As it can be seen in Figure 5 many routing protocols have been developed over the last few years and many innovative routing mechanisms have been proposed [25].

Representative sensor network related communication technologies includes Wireless Fidelity (Wi-Fi), ZigBee, IQRF, Ultra-Wide Band (UWB), and Wireless Hart. Even though it have not been used extensively on a large scale yet, wireless sensor networks (WSNs) offer a substantial technology that helps to cover city conditions monitoring needs. This technology gives the ability to efficiently and quickly detect various spatial events, such as the problems of a region of high pollutant concentration by processing real time data. Air pollution or monitoring of urban environments could be supported by dense WSN of nodes with monitoring capabilities. These advanced real-time systems are wireless, highly distributed, also used in addition to sensors actuators as interfaces deployed across a wide geographic area.

B. *Sensing of the city and cloud computing*

The intelligence of a sensor network is predominantly reflected in provision of real time information and in the fact that the real-time sensor data might be integrated with environmental modelling and control. The primary concept essential for capability of real-time information distribution and use them in city services lies in establishing the digital infrastructure for processing of both WSN and video surveillance data resulting in a more efficient event detection. The growing penetrations of fixed and wireless networks permit that such sensors and systems to be connected to distributed processing centres. The smart city connects the sensors to sense the city systems, and process the sensing information by cloud computing and so on to integrate cyber space and things of internet [15].

Contemporary wireless sensor networks are principally treated as simply a new data source for integration with other conventional spatial and open data information systems. Examples include sensors connecting buildings, infrastructure, transport, networks and utilities, offers a physical space for experimentation and validation of the Internet of Things (IoT) functions. The data is delivered in real-time through the cloud to the service providers, users, and other stakeholders.

V. The Proposed Communication Platform

The diagram of proposed communication platform for the development of city services and for the integration and use of rendering and actuating technologies is shown in Figure 6. The help of instrumentation and interconnection of mobile devices and sensors, which collect and analyse real-world data, creates a dynamic environment with numerous groups of users concerned in different city events. We may use smart travel as an example. The provision of real-time information about urban environments could give real-time travel information for passengers, such as current running times of buses or trains. Traffic monitoring can be supported by monitoring of means of transport, weather and traffic condition.

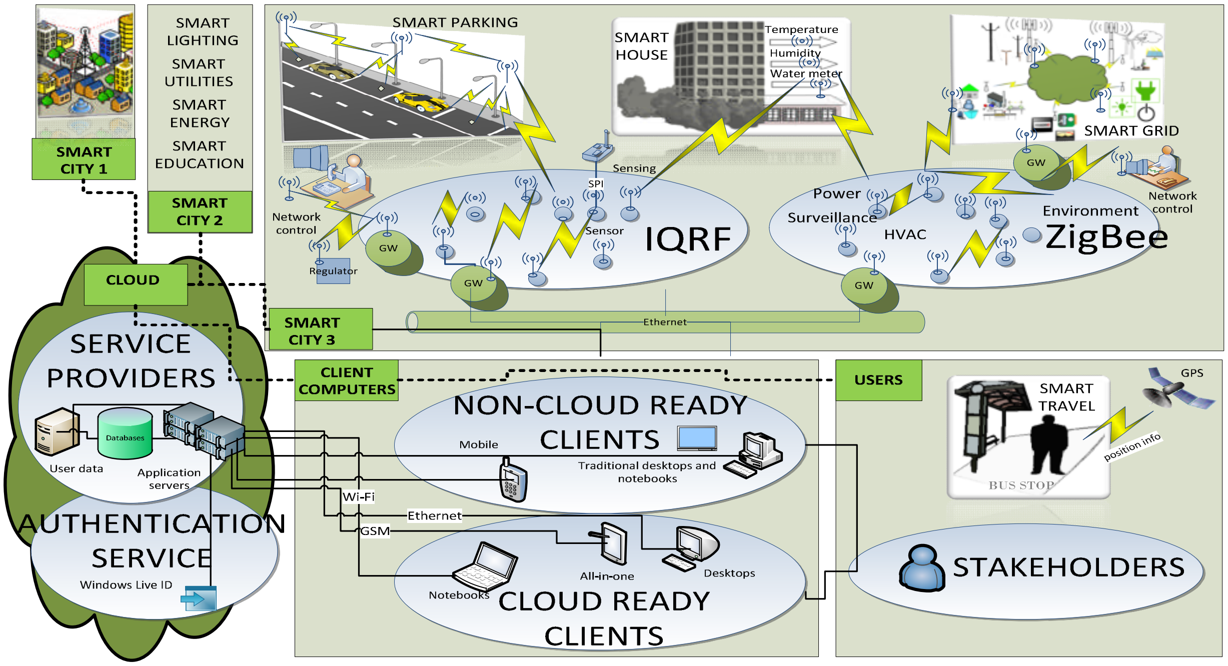


Figure 6. Communication platform for the development of city services and for remote control.

The widespread use of digital sensors and digital control systems for the control and operation of urban infrastructure includes traffic sensors and actuators, building management systems in the smart house solution, digital utility meters in the smart utility area, and so forth. There are plenty of opportunities for new services by interconnecting physical and virtual worlds with a huge amount of wireless nodes distributed in houses, vehicles, streets, buildings and many other public environments. It may be components that will be applied in the fields of vehicle-mounted terminals, wireless meter reading, streetlight (smart lighting), transportation (smart parking), household appliances (smart energy), and industrial cameras (smart house). Ambient spatial intelligence for sustainable cities contains the potential for augmenting such networks with the capabilities to not only capture, but also process, query, and even use spatial data in the network itself.

Thanks to a distributed network of intelligent sensor nodes, a wide collection of parameters can be measured for better management of the city, and data are delivered wirelessly and in real-time to the citizens or the appropriate authorities [1]. By using networks and sensors to measure and control processes, and the cloud for the information sharing, stakeholders can make immediate diagnoses and correct the problem by appropriate action in the event of an accident or another incident. The principle of Smart Earth is that, sensors are embedded everywhere: in the railways, bridges, tunnels, roads, buildings, water systems, dams, commercial equipment and medical equipment, and then physical facilities can be perceived, so information technology extends into the physical world, constructing a "Internet of Things" [15]. A mashup is important to make existing data more useful as a combination of two or more sources to create new services. This can include a wide range of uses from identity and access management to application, web, and portal servers that power stakeholder services and web sites to ensure a view of the citizen and real-time updates of information across city systems. A real time dashboard for monitoring city systems offers solutions to help city authorities manage smart city policies and guarantee the necessary controls and procedures are in place for better governance. The application layer could include interactive modules that notify the users of events or alerts and allow them to trigger further actions.

VI. Conclusions

The first objective of this article was to provide an overview of the smart city concept, and the second goal was to propose a technological solution and communication platform for the development of city services in relation to the improvement of urban systems and services. Monitoring,

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-399-5

measurement and remote control are important areas of a smart city necessary for the full utilization of the opportunities offered by information and communication technologies. Figure 6 included in the previous section gives an overview of the proposed communication platform for the development of city services. This communication platform can be used for the improvement of specific urban systems and services. Directions for our future work include applying the framework for implementation of a proposed communication platform, described in the article in an application for a specific city application.

ACKNOWLEDGMENT

This research has been supported by the European ARTEMIS Industry Association by the project 7H12002 "Interactive Power Devices for Efficiency in Automotive with Increased Reliability and Safety", and by the CZ.1.05/1.1.00/02.0068 project OP RDI "CEITEC - Central European Institute of Technology".

REFERENCES

[1] C. Harrison, , and I. Donnelly, “A Theory of Smart Cities.” Proceedings of the 55th Annual Meeting of the ISSS. Hull: International Society for the Systems Sciences, 2011, p. 11-20.

[2] K. Viljanen., A. Poikola, and P. Koponen. “Information navigation in the city. the City of Helsinki: Forum Virium Helsinki and the Fireball project,” 2012.

[3] *Rifidi - Connect the Internet of Things.* [retrieved: 1, 2015], from http://sourceforge.net/projects/rifidi/.

[4] S. Ahson, S. and M. Ilyas, RFID Handbook: Applications, Technology, Security, and Privacy, Boca Raton: CRC Press, 2010.

[5] H. Schaffers, N. Komninos, B. Pallot, Trousse, M. Nilsson, and A. Oliveira, “Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation.” In J. Domingue, Future Internet Assembly 2011: Achievements and Technological Promises (pp. 431–446). Springer, 2011.

[6] *ASPIRE for SME*, (1, 12 2009). Retrieved 1 1, 2015, from ASPIRE - The EU funded project that brings RFID to SMEs.

[7] *AspireRFID Middleware*, [retrieved: 1, 2015], from http://wiki.aspire.ow2.org/xwiki/bin/view/Main/WebHome.

[8] N. Bartneck, V. Klaas, and H. Schönherr, “Optimizing Processes with RFID and Auto ID”, Erlangen: Publicis Publishing, 2009.

[9] H. Bidgoli, “Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management.” Hoboken, New Jersey: John Wiley & Sons, 2006.

[10] C. Cerrada, I. Abad, I., and J. Cerrada, “DEPCAS: an industrial approach to RFID middleware.” Industrial Technology (ICIT), 2010 IEEE International Conference. Viña del Mar, Chile, 2010, pp. 1394 – 1399.

[11] A. Cimmino, et al., “The Role of Small Cell Technology in Future Smart City.” Transactions on Emerging Telecommunications Technologies, 2013, pp. 1-10.

[12] *CUHK EPCGlobal RFID Middleware 1.0* (Open Source), 2009. [retrieved: 1, 2015], from Mobile Technologies Centre: http://mobitec.ie.cuhk.edu.hk/rfid/middleware/.

[13] *detego® Integration Platforms.* [retrieved: 12, 2013], from http://www.enso-detego.com/en/support/integration-platforms.html.

[14] P. Fuhrer, and D. Guinard, *Building a Smart Hospital using RFID technologies*. [retrieved: 2, 2015], from SAP Research: http://www.gmipsoft.com/unifr/docs/wp_smarthospital.pdf.

[15] L. Hao., X. Lei, Z. Yan., and Y. ChunLi, “The application and implementation research of smart city in China.” System Science and Engineering (ICSSE), 2012, pp. 288 - 292.

[16] D. Hunt, A. Puglia, and M. Puglia. “RFID: A Guide to Radio Frequency Identification. Hoboken”, New Jersey: John Wiley & Sons, 2007.

[17] H. Lehpamer, (2012). “RFID Design Principles.” Norwooe: Artech House.

[18] *OAT Technology* [retrieved: 1, 2015], from http://www.oatsystems.com/technology/.

[19] *Open Source RFID Platform.* [retrieved: 1, 2015], from https://code.google.com/p/fosstrak/.

[20] A. Paikin, *Flash memory.* Retrieved 1, 2015, from http://www.hitequest.com/Kiss/Flash_terms.htm.

[21] J. Pinto, *Wireless sensor networks.* [retrieved: 1, 2015], from Jimpinto.com: http://www.jimpinto.com/writings/sensornetworks.html.

[22] M. Sandvido, F. Chu, and A. Kulkarni, (n.d.). *From NAND Flash Memory and Its Role in Storage Architectures* IEEE Xplore Digital Library. [retrieved: 1, 2015]: http://ieeexplore.ieee.org/ieee_pilot/articles/96jproc11/jproc-MSanvido-20043.

[23] K. Schwartz, *BizTalk RFID: Making RFID Deployments Easy, Simple and Economical*. [retrieved: 1, 2015], from Mircrosoft Developer Network: http://msdn.microsoft.com/en-us/library/aa479354.aspx.

[24] J. Sikander, *Microsoft RFID Technology Overview*, [retrieved: 2, 2014], from Microsoft Developer Network: http://msdn.microsoft.com/en-us/library/aa479362.aspx.

[25] K. Sohraby, D. Minoli, and T. Znati. “Wireless Sensor Networks: Technology, Protocols, and Applications.” New Jersey: John Wiley & Sons, 2007.

[26] J. Zheng, and A. Jamalipour, “Wireless Sensor Networks: A Networking Perspective.” New Jersey: John Wiley & Sons, 2009.

Subspace Identification Methods and Multivariable Control for a Doubly-fed Induction Generator

Danna L. Albarracín Ávila
Electrical Engineering
Universidad Tecnológica de Pereira
Pereira, Colombia
Email: dlalbarracin@utp.edu.co

Eduardo Giraldo
Department of Electrical Engineering
Universidad Tecnológica de Pereira
Pereira, Colombia
Email: egiraldos@utp.edu.co

***Abstract*—Doubly-Fed Induction Generators (DFIG) are being widely used on Wind Turbine Generator System (WTGS), although synchronous generators are being extensively utilized too. Thus, there are different types of identification techniques to obtain an estimated model of system. Orthogonal Decomposition (ORT) and Multivariable Output Error State Space (MOESP) algorithms are two well-known subspace identification techniques, discussed in this paper. These identification techniques are often implemented for multivariable systems. Subspace identification algorithms are attractive since the state space form is highly suitable to estimate, predict, filters, as well as for control design. The Doubly-Fed Induction Generator is widely used in the variable-speed constant-frequency Wind Power Generation System, using the vector control scheme which provides good performance in maximum wind energy capturing. In the traditional vector control scheme, the reduced doubly-fed induction generator model, which neglects the stator flux transients, is employed to simplify the rotor current inner-loop controller using the standard Proportional-Integral (PI) regulation. The approach proposed in this paper must be approached in order to obtain an approximate model influenced by the behaviour of the system with certain functional characteristics and environmental changes.**

Keywords-subspace identification; parameters estimated; wind turbine; multivariable system; integral controller.

I. Introduction

Subspace identification methods have proven to be such a valuable tool in identification area since past few years. This interest is due to the ability of the subspace approach in providing accurate state-space models for multivariable linear systems directly from input-output data [1][2]. In the identification procedure, the main step is to compute the Singular Value Decomposition (SVD) of a block Hankel matrix H [3] constructed with input-output data. The computation of SVD can be done offline or online. The offline can be easily converted to an adaptive online version for a slow time Vaihingen system, such as process control system [3]-[4].

In control systems design with advanced model-based control methods, lots of time and cost are required for system identification to construct accurate models like a variable-speed induction machine. To save the cost for modeling of dynamical systems, it is important to choose proper test signals so that the leading dynamics are stimulated efficiently [5]. A doubly-fed induction generator is a variable-speed induction machine widely utilized in the modern wind power industry. The reasons for using variable speed wind turbines are fourfold: 1) a higher energy yield, 2) a reduction of mechanical loads and a simpler pitch control, 3) an extensive controllability of both active and reactive powers, and 4) less fluctuation in output power [5][6].

There are several vector control strategies for a wind turbine driven DFIG's; one common method is to control the rotor current with Stator Flux Orientation (SFO) [7]. In all the existing control schemes, the reduced DFIG model, which neglects the transient terms in the stator flux and most of them use the traditional PI regulator in their rotor current controllers [8].

In this paper, two subspace approaches are observed in order to perform open-loop system identification. These two approaches are the Orthogonal Decomposition (ORT) method and Multivariable Output Error State Space (MOESP) method. The objective of this paper is to analyse the performance of the models in identifying the state space model of a doubly-fed induction generator (DFIG) and the performance of the Integral multivariable controller applied to the estimated system, as shown in Section 3 and Section 4, respectively.

II. Modeling of DFIG

A. General Descriptions

Figure 1 illustrates the DFIG wind power generation system consisting of a wound rotor induction generator and a back-to-back converter between the rotor slip rings and the grid.

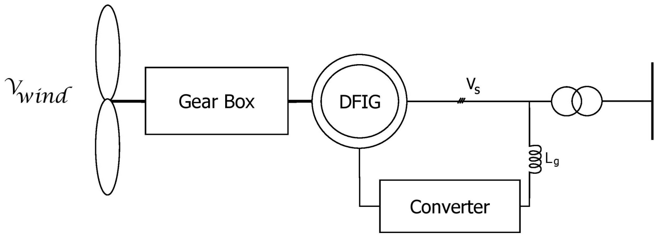


Figure 1. Variable-speed wind turbine system using DFIG.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-399-5

B. Mathematical Model of DFIG

Figure 2 shows the equivalent circuit of a DFIG in the reference frame rotating at the synchronous angular speed ω_1 [7].

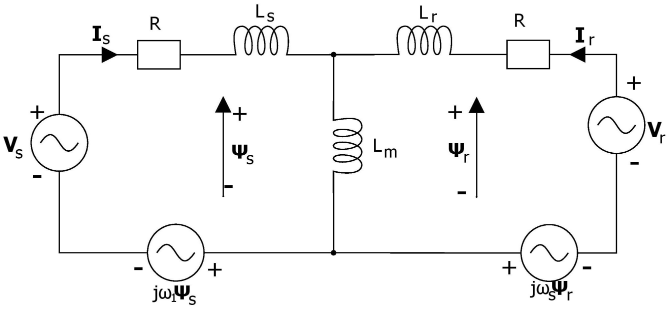


Figure 2. T-equivalent circuit of DFIG in reference frame rotating at ω_1.

According to Figure 2, the stator and rotor space-vector fluxes Ψ_s and Ψ_r are given respectively by

$$\begin{cases} \boldsymbol{\Psi}_s = \mathbf{L}_s\mathbf{I}_s + \mathbf{L}_m\mathbf{I}_r \\ \boldsymbol{\Psi}_r = \mathbf{L}_m\mathbf{I}_s + \mathbf{L}_r\mathbf{I}_r \end{cases} \tag{1}$$

where $L_{\sigma s}$, $L_{\sigma r}$ and L_m are the stator and rotor leakage and mutual inductances, and $L_s = L_{\sigma s} + L_m$ and $L_r = L_{\sigma r} + L_m$ the self inductances of the stator and rotor windings, respectively.

The stator and rotor space-vector voltages $\mathbf{V}_s$ and $\mathbf{V}_r$ in the same reference frame can be expressed as

$$\begin{cases} \mathbf{V}_s = R_s\mathbf{I}_s + \dfrac{d\boldsymbol{\Psi}_s}{dt} + j\omega_1\boldsymbol{\Psi}_s \\ \mathbf{V}_r = R_r\mathbf{I}_r + \dfrac{d\boldsymbol{\Psi}_r}{dt} + j(\omega_1 - \omega_r)\boldsymbol{\Psi}_r \end{cases} \tag{2}$$

where $\mathbf{I}_s$ and $\mathbf{I}_r$ are the stator and rotor current vectors, $\mathbf{R}_s$ and $\mathbf{R}_r$ the stator and rotor resistances, ω_r is the rotor angular speed, and $\omega_s = \omega_1 - \omega_r$ the slip angular speed.

The model of DFIG uses the SVO scheme described in [7], in which we can rewrite the rotor voltage equation in the synchronous $d-q$ reference frame as

$$\begin{cases} \mathbf{V}_{rd} = V'_{rd} + \dfrac{L_m}{L_s}\left(\mathbf{V}_s - \dfrac{R_s}{L_s}\boldsymbol{\Psi}_{sd} + \omega_r\boldsymbol{\Psi}_{sq}\right) \\ \mathbf{V}_{rq} = V'_{rq} - \dfrac{L_m}{L_s}\left(\dfrac{R_s}{L_s}\boldsymbol{\Psi}_{sq} + \omega_r\boldsymbol{\Psi}_{sd}\right) \end{cases} \tag{3}$$

where

$$\begin{cases} \mathbf{V}'_{rd} = R'_r I_{rd} + \sigma L_r \dfrac{dI_{rd}}{dt} - \omega_s\sigma L_r I_{rq} \\ \mathbf{V}'_{rq} = R'_r I_{rq} + \sigma L_r \dfrac{dI_{rq}}{dt} + \omega_s\sigma L_r I_{rd} \end{cases} \tag{4}$$

The system in state space representation is presented as follows:

$$\begin{aligned} \frac{dI_{rd}}{dt} &= \frac{1}{\sigma L_r}\left(V'_{rd} - R'_r I_{rd} + \omega_s\sigma L_r I_{rq}\right) \\ \frac{dI_{rq}}{dt} &= \frac{1}{\sigma L_r}\left(V'_{rq} - R'_r I_{rq} - \omega_s\sigma L_r I_{rd}\right) \\ y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} I_{rd} \\ I_{rq} \end{bmatrix} \end{aligned} \tag{5}$$

III. Subspace identification method

The model presented in Section 2 is nonlinear. The linear model of the DFIG can be obtained by applying an subspace approach as the ORT (Orthogonal Decomposition) or MOESP (Multivariable Output Error State Space) methods to the system around an operating point [3], as follows:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k] \end{aligned} \tag{6}$$

being $\mathbf{x} \in \mathbb{R}^{n\times 1}$, where $n = 6$ is the number of state variables, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, where $p = 2$ is the number of inputs, $\mathbf{u} \in \mathbb{R}^{p\times 1}$, and $\mathbf{C} \in \mathbb{R}^{m\times n}$, where $m = 2$ is the number of outputs.

Suppose that the input-output data $\{u(t), y(t), t = 0, 1, \cdots, N + 2k - 2\}$ are given with N sufficiently large and $k > n$. Based on the input−output data [1], we define as usual block Hankel matrices of past input is defined as

$$U_p = \begin{bmatrix} u(0) & u(1) & \cdots & u(N-1) \\ u(1) & u(2) & \cdots & u(N) \\ \vdots & \vdots & \ddots & \vdots \\ u(k-1) & u(k) & \cdots & u(N+k-2) \end{bmatrix} \tag{7}$$

and future input as

$$U_f = \begin{bmatrix} u(k) & u(k+1) & \cdots & u(k+N-1) \\ u(k+1) & u(k+2) & \cdots & u(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ u(2k-1) & u(2k) & \cdots & u(N+2k-2) \end{bmatrix} \tag{8}$$

where $U_p, U_f \in \mathbb{R}^{km\times N}$. Similarly, we define past and future output $Y_p, Y_f \in \mathbb{R}^{kp\times N}$ respectively. In [9], the constructed input and output are given as

$$U = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, Y = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} \tag{9}$$

The extended observability matrix of order i is defined as

$$\Gamma_i = \begin{bmatrix} C^T & (CA)^T & \cdots & (CA^{i-1})^T \end{bmatrix}^T \tag{10}$$

and the block lower triangular Toeplitz matrix as

$$H_i = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{i-2}B & CA^{i-3}B & \cdots & D \end{bmatrix} \tag{11}$$

we consider a related LQ decomposition

$$\begin{bmatrix} U_f \\ U_p \\ Y_p \\ Y_f \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \end{bmatrix} \tag{12}$$

Step 1: Construct data matrices of U_p, U_f, Y_p and Y_f
Step 2: Perform LQ factorization by (12)
Step 3: Perform SVD to the working matrix

ORT: $G = [L_{42}]$
MOESP: $G = [L_{42}L_{43}]$
where
$$G = \begin{bmatrix} \hat{U} & \bar{U} \end{bmatrix} \begin{bmatrix} \hat{S} & 0 \\ 0 & \bar{S} \end{bmatrix} \begin{bmatrix} \hat{V} \\ \bar{V} \end{bmatrix} \simeq \hat{U}\hat{S}\hat{V}^T$$

Step 4: Compute the estimates of A and C by (10)
$$\Gamma_i = \hat{U}\hat{S}^{1/2}$$
Step 5: Compute the estimates of B and D by (11)
$$\bar{U}^T L_{41} L_{11}^{-1} = \bar{U}^T H_i$$

IV. Multivariable control

A. Integral multivariable controller

In Figure 3, the controller structure is defined by considering accumulated for the error in the signal control [10] is shown.

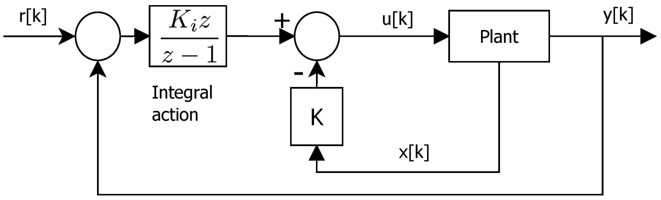


Figure 3. Block diagram of the integral action controller.

In [11], the error in the signal control is defined as follows

$$\mathbf{u}[k] = -\mathbf{K}\mathbf{x}[k] + \mathbf{K}_i\mathbf{v}[k] \tag{13}$$

with

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \mathbf{e}[k] \tag{14}$$

where the error $\mathbf{e}[k]$ is defined as follows

$$\mathbf{e}[k] = \mathbf{r}[k] - \mathbf{y}[k] \tag{15}$$

and $\mathbf{v}[0] = 0$. Therefore, an extended system can be defined as follows

$$\begin{aligned} \begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{v}[k+1] \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{v}[k] \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{B} \\ -\mathbf{D} \end{bmatrix} \mathbf{u}[k] + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} \mathbf{r}[k] \end{aligned} \tag{16}$$

$$\mathbf{y}[k] = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{v}[k] \end{bmatrix} \tag{17}$$

where the control signal is computed as follows

$$\mathbf{u}[k] = -\begin{bmatrix} \mathbf{K} & -\mathbf{K}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{v}[k] \end{bmatrix} \tag{18}$$

V. Results

The parameters of the doubly-fed induction generator used in the simulations are $R_s = 0.01(p.u.)$, $R_r = 0.009(p.u.)$, $L_{\sigma r} = 0.18(p.u.)$, $L_{\sigma s} = 0.07(p.u.)$, $L_m = 3.015(p.u.)$ and $\omega_s = 1.2(p.u.)$.

The wind energy conversion system is identified from behavior shown in Figure 4 and Figure 5. The estimated model by the ORT and MOESP method are shown in (19) and (20), respectively.

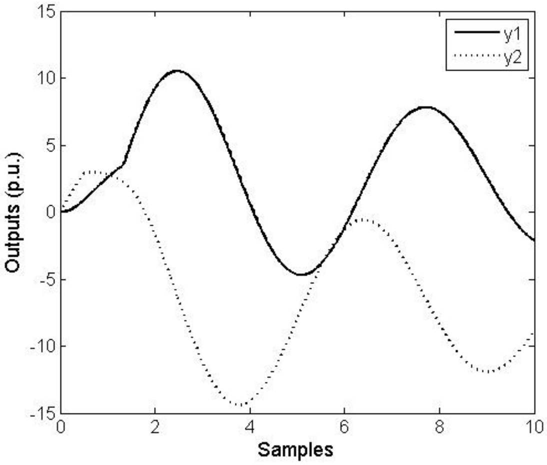


Figure 4. Output behavior.

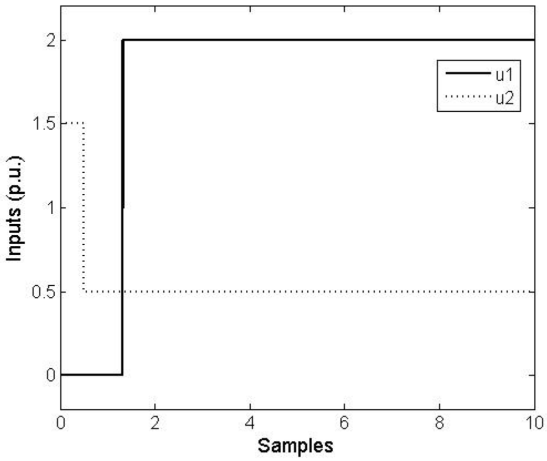


Figure 5. Input behavior.

$$\mathbf{A} = \begin{bmatrix} 0.9992 & 0.0021 \\ -0.0684 & 0.9992 \end{bmatrix} \mathbf{B} = \begin{bmatrix} -0.0179 & 0.0007 \\ -0.0026 & -0.1041 \end{bmatrix}$$
$$\mathbf{C} = \begin{bmatrix} -2.3294 & -0.0199 \\ 0.1133 & -0.4085 \end{bmatrix} \mathbf{D} = \begin{bmatrix} 0.0594 & -0.0790 \\ 0.0054 & 0.0165 \end{bmatrix} \quad (19)$$

$$\mathbf{A} = \begin{bmatrix} 0.9992 & 0.0112 \\ -0.0129 & 0.9992 \end{bmatrix} \mathbf{B} = \begin{bmatrix} -0.0566 & 0.0620 \\ -0.1034 & -0.0494 \end{bmatrix}$$
$$\mathbf{C} = \begin{bmatrix} -0.2080 & -0.2871 \\ 0.3077 & -0.1940 \end{bmatrix} \mathbf{D} = \begin{bmatrix} 0.0467 & -0.2088 \\ 0.0014 & -0.0063 \end{bmatrix} \quad (20)$$

The graph of superimpose between simulated true system data and simulated outputs obtained from ORT and MOESP models are shown in Figure 6 and Figure 7. The input data system is shown in Figure 8. In Figure 6 and Figure 7, the MOESP method gives the best performance overall in open loop compared to the ORT model over the real response of system.

In Figure 9, Figure 10 and Figure 11, the outputs tracking performance and the control inputs are shown using the ORT model.

In Figure 12, Figure 13 and Figure 14, the outputs tracking performance and the control inputs are shown using the MOESP model.

As shown in Figure 14, the results for the control inputs by using the ORT method is better than the obtained results shown in Figure 11, the MOESP method, in terms of control signal amplitude.

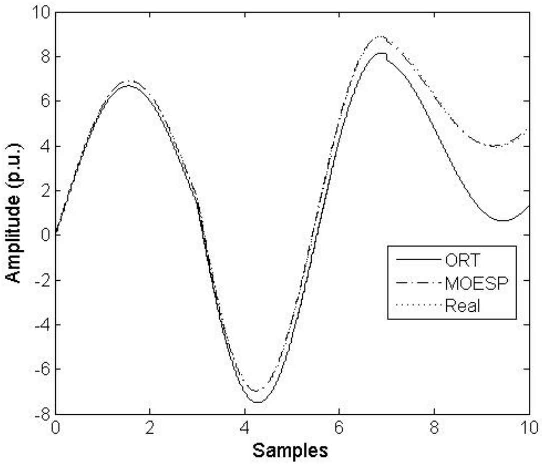


Figure 6. Superimpose between simulated true data system of output 1 and simulated output 1 obtained from estimated system matrices of ORT and MOESP methods.

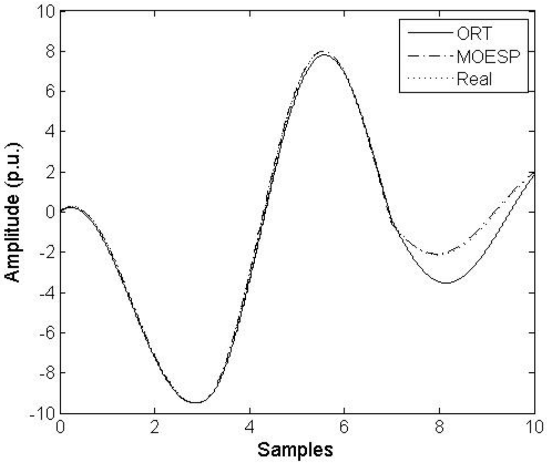


Figure 7. Superimpose between simulated true data system of output 2 and simulated output 2 obtained from estimated system matrices of ORT and MOESP methods.

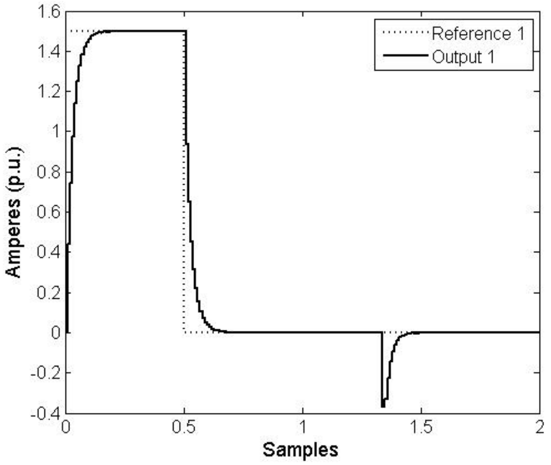


Figure 9. Output 1 tracking performance of ORT model by using integral multivariable controller.

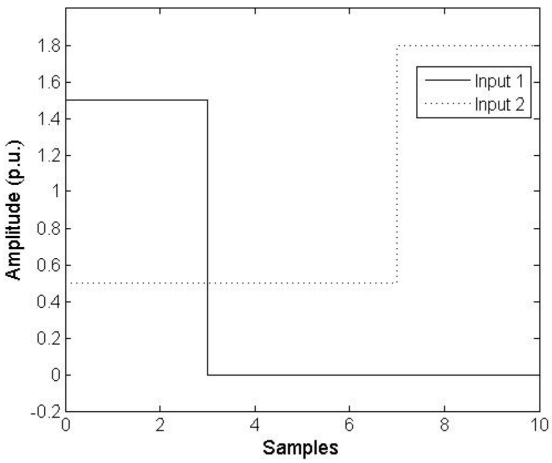


Figure 8. Input data of true system and estimated system matrices of ORT and MOESP methods.

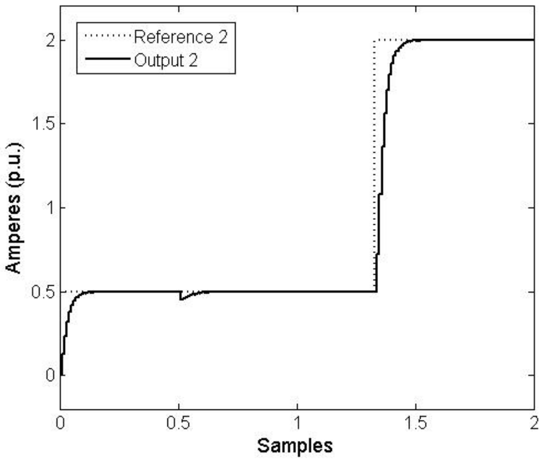


Figure 10. Output 2 tracking performance of ORT model by using integral multivariable controller.

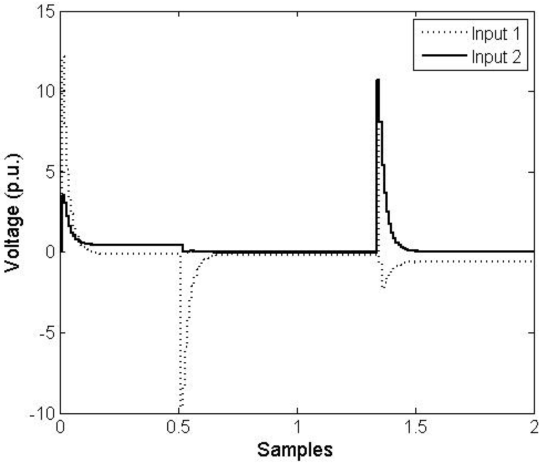


Figure 11. Control signal of ORT model.

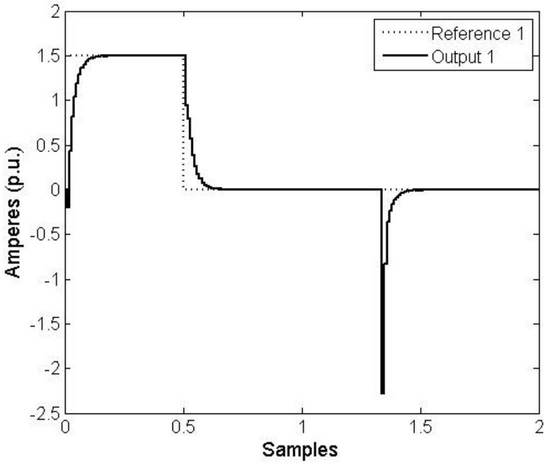


Figure 12. Output 1 tracking performance of MOESP model by using integral multivariable controller.

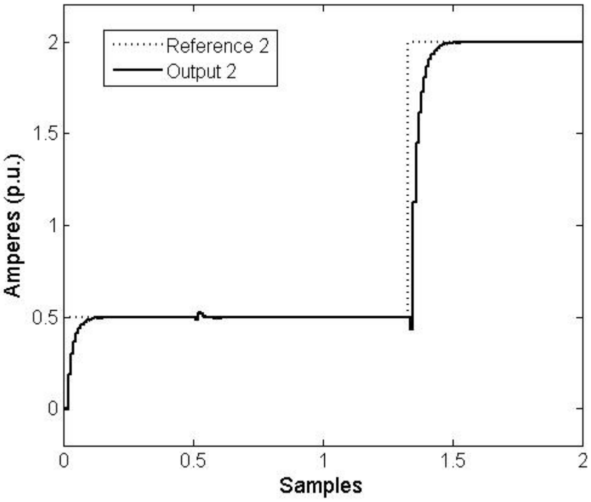


Figure 13. Output 2 tracking performance of MOESP model by using integral multivariable controller.

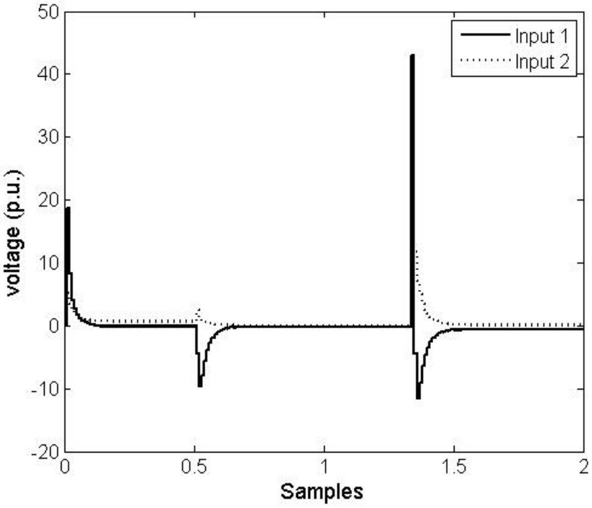


Figure 14. Control signal of MOESP model.

VI. Conclusion and Future Work

In this paper, two subspace identification models of a doubly-fed induction generator are obtained based on two subspace algorithms which are ORT and MOESP; they show an advantages when the parameters of system are unknown and are known only the input-output values, because they make it possible to obtain an estimated model of system. The MOESP method gives the best performance overall in open loop compared to the ORT method with the same inputs of real system. However, the output tracking performance by using the ORT model is better than the obtained results with the MOESP model in terms of overshoot and control signal amplitude when it is applied the integral multivariable controller. The settling time in tracking performance in both models is the same.

Since practically many real-world systems are time-varying, the approach proposed in this paper must be considered in order to obtain an approximate model influenced by the behaviour of the system with certain functional characteristics and environmental changes. These simulations can be obtained with a programming tool, like MATLAB/Simulink [12] for testing the results of identification and control algorithms designed before being applied on real-world systems with time-varying.

The proposed methodologies for identification, the MOESP and ORT methods, shown a state space representation of the system without rearranging the estimated parameters in each vector and obtaining the parameters matrices ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$) with a minor dimensions compared to matrices shown in [13], improving the computational cost with the algorithms.

A future work includes the implementation of algorithms for identification coupled multivariable systems, where couplings are considered between inputs and outputs as possible disturbances present in each subsystem, allow it to obtain an approximate model of system and propose a control strategy of the system.

Acknowledgment

This paper was developed under the research project “Identificación de sistemas multivariables aplicada a generadores eólicos” funded by the Universidad Tecnológica de Pereira with code 6-14-1, and the MSc. thesis “Control óptimo de un sistema multi-variable aplicado a un generador eólico conectado a un sistema de potencia” approved by the “Convocatoria para financiar proyectos de grado de estudiantes de pregrado y posgrado año 2103” funded by the Universidad Tecnológica de Pereira with code E6-14-6.

References

[1] T. Katayama, *Subspace Methods for System Identification*, ser. Communications and Control Engineering, 2006. [retrieved: November, 2014]. [Online]. Available: http://books.google.com.co/books?id=Ge_HTdCBtZAC

[2] G. Tao, “Recursive Subspace Model Identification Based on Orthogonal Projection and Principal Component Analysis,” in *International Conference on Computer Application and System Modeling (ICCASM 2010)*, Oct 2010, pp. V15–422–V15–429.

[3] I. W. Jamaludin and N. A. Wahab, “N4SID and MOESP subspace identification methods,” in *9th International Colloquium on Signal Processing and its Applications (CSPA)*, Mar 2013, pp. 140–145.

[4] D. L. Albarracin-Avila and E. Giraldo, “Adaptive optimal multivariable control of a Permanent Magnet Synchronous Generator,” in *Transmission & Distribution Conference and Exposition - Latin America (PES T& D-LA)*, Sept 2014, pp. 1–5.

[5] L. Shuhui and T. Haskew, “Energy Capture, Conversion, and Control Study of DFIG Wind Turbine under Weibull Wind Distribution,” in *Power and Energy Society General Meeting PES ’09*, Jul 2009, pp. 1–9.

[6] W. Zhi-nong and Y. Xiao-yong, “The Intelligent Control of DFIG-Based Wind Generation ,” in *International Conference on Sustainable Power Generation and Supply (SUPERGEN ’09)*, Apr 2009, pp. 1–5.

[7] H. Jia-bing and Z. Jian Guo, “The internal model current control for wind turbine driven doubly-fed induction generator,” in *Industry Applications Conference, 2006. 41st IAS Annual Meeting. Conference Record of the 2006 IEEE*, Oct 2006, pp. 209–215.

[8] R. Pena, J. Clare, and G. Asher, “Doubly fed induction generator using back-to-back pwm converters and its application to variable-speed wind-energy generation,” vol. 143, no. 3, May 1996, pp. 231 – 241.

[9] M. Aziz and R. Mohd-Mokhtar, “Performance Measure of Some Subspace-Based Methods for Closed-Loop System Identification,” in *2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM)*, Sept 2010, pp. 255–260.

[10] K. H. A.-H. K. Salim, R. and B. Khedjar, “LQR with integral action controller applied to a three-phase three-switch three-level AC/DC converter,” in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, Nov 2010, pp. 550–555.

[11] D. Giraldo and E. Giraldo, Eds., *Teoría de Control Digital*. Productos Editoriales y Audiovisuales-Produmedios, 2012, ISBN: 978-958-722-151-0.

[12] S. Karris, *Introduction to Simulink: With Engineering Applications*. Orchard Publications, 2008. [Online]. Available: https://books.google.com.co/books?id=B8ssmvQmpVoC

[13] D. L. Albarracin-Avila and E. Giraldo, “Identification and multivariable control in state space of a permanent magnet synchronous generator,” *TECCIENCIA*, vol. 9, no. 16, pp. 97–106, 2014.

Dynamic Configuration of Distributed Systems for Disaster Management

Florian Segor, Igor Tchouchenkov, Rainer Schönbein,
Matthias Kollmann, Christian Frey
Fraunhofer Institute of Optronics, System Technologies and
Image Exploitation IOSB
Karlsruhe, Germany
e-mail: {florian.segor, tchouchenkov.igor,
rainer.schoenbein, matthias.kollmann,
christian.frey}@iosb.fraunhofer.de

Stefan Rilling, Rainer Worst
Fraunhofer Institute for Intelligent Analysis and Information
Systems IAIS,
Sankt Augustin, Germany
e-mail: {stefan.rilling, rainer.worst}@iais.fraunhofer.de

***Abstract*—In natural and man-made disasters, it is a necessity for rescue teams to get a quick overview of the situation in place. Robot-supported sensor networks are increasingly used to accelerate surveillance and search operations in complex situations. An appropriate robust system architecture has to support dynamical changes in connectivity as well as in number and type of robots and sensors in action. The proposed solution for a dynamic configuration of a distributed system with heterogeneous sensors and robots for disaster management is based on the Robot Operating System (ROS). The configuration uses an active Information Module with access to the descriptions of the characteristics and capabilities of all relevant system components. The modular descriptions are based on XML standard. Every component has at least one description file with capabilities of the component and their relevant technical characteristics. Descriptions of complex components containing sub-components are hierarchically with references to descriptions of sub parts. Between the system components direct communication links can be established to make the distributed system more robust. External systems may also get information about available capabilities from the Information Module and request needed services directly from the components. The main task of this work is to introduce a dynamic but robust system architecture for controling complex heterogeneous sensor systems to support rescue forces in disaster relive.**

Keywords-Distributed system, disaster management, dynamic configuration, modular, hierarchical, ROS, XML, heterogeneous sensors, robotics

I. INTRODUCTION

Heterogeneous distributed sensor-based surveillance systems have strongly spread during the last years and have grown on account of new demands. A novel concept for such systems for disaster management has been developed by five German Fraunhofer Institutes in the joint research project SENEKA [1][2]. The disaster management in rapidly changing scenarios requires a quick configuration and re-configuration of the system and components with minimal work load on the rescuers. This is a challenge because not only every single component, but normally also the needed communication infrastructure has to be set up and adjusted.

The objective of the SENEKA concept is to network various unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs) and sensor systems usable by first responders. Within the project, possible solutions for a flexible dynamic system reconfiguration during a mission are examined and evaluated [2][3]. In this article, the setup for a dynamic configuration of complex distributed systems for disaster management is described in Section II, as well as a solution concept for automatic configuration with software generation for sensors and sensor modules in Section III. The paper is closed with a final discussion in Section IV.

II. DYNAMIC CONFIGURATION

The concept for the dynamic configuration of SENEKA system is based on a (conceptually redundant) central Information Module (SENEKA-Hub), which has access to data describing the characteristics and capabilities of all relevant system components (see Figure 1). The SENEKA-Hub must always know which components are online, their status and what capabilities and functionalities they can provide.

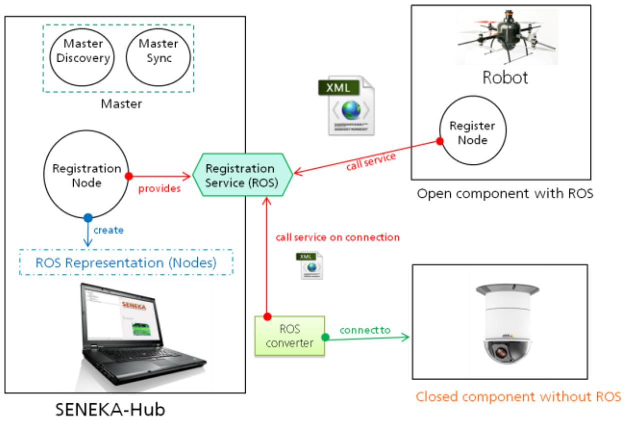


Figure 1: A sample structure of dynamic configurable SENEKA system with heterogeneous components.

To guarantee the automatic "technical" integration of the components in the overall system, the SENEKA-Hub must also "know" relevant technical details about the integration parameters of the abilities (network-addresses, interfaces, protocols, etc.) of all serving components to notify all interested information consumers (human users and technical components or systems). The communication between the system components as well as to external systems is handled not over the SENEKA-Hub, but directly. This makes the system much more robust – especially under poor communication channel quality in disaster areas.

The implementation of such communication architecture is based on the Robot Operating System (ROS) [4]. It is an Open Source project for distributed control systems. For fast communication a Publisher-Subscriber architecture supported by services is used.

The SENEKA-Hub is acting as a ROS Master and collects information about all connected system components. Four types of components can be used:

1) "Open" components with ROS can be programmed to add functions needed for dynamic configuration.

2) "Open" components without ROS have to be adapted (e.g., programmed) to add ROS interfaces and functions needed for dynamic configuration.

3) "Closed" components with ROS supporting some ROS capabilities, but can't be adapted.

4) "Closed" components without ROS and no ROS support that provide other interfaces only (for example a Web-Cam).

Different types of components must be handled in different ways. "Open" components can inform the SENEKA-Hub after their suitable adaptation when connecting to the system or disconnecting from it, providing their own descriptions and report about changing capabilities, if needed. "Closed" components must be "installed" by introducing their abilities to the SENEKA-Hub before the first usage. During registration, "closed" components with ROS only need to transfer their "call sign". The registration of other components must be ensured on the other suitable way – for example, using other "open" components (e.g., ground control station like [5]) to check if an "installed" component is now alive. The "intelligent" components should normally configure themselves. The special "Configurator" program can set parameters of different components and infrastructure (e.g., computers) as needed.

The central problem of such hierarchical heterogeneous systems is the determination of consistent components' descriptions at different hierarchy levels. The components and their interfaces have very different complexity – from easy temperature sensors with RS232 up to distributed command and control centres equipped with many sensors, sensor carriers, servers and workplaces, but all their descriptions must be standardized. Moreover, the same sub-components can be used many times in different parts of the same system or in different systems with variable parameters, so that the descriptions should be easy adaptable. For example, SENEKA contains multiple mobile sensors mounted on light UAVs or UGVs as well as stationary sensors controlled over an ergonomic user interface from an improved AMFIS ground control station [5][6][7] (Figure 2). After the registration of UAVs and UGVs, different control and data channels for direct communication with the control station or with other components can be used as needed.

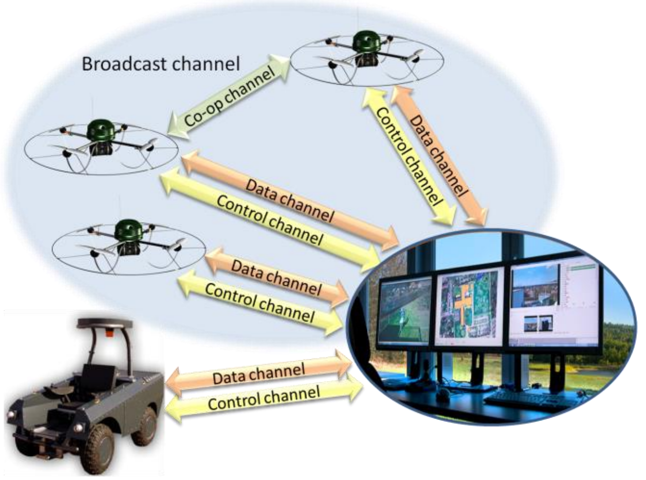
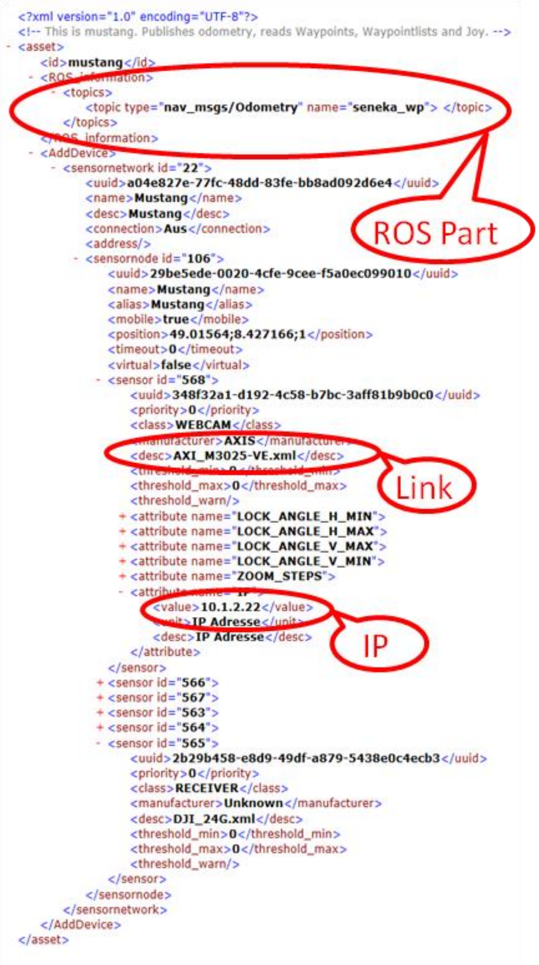


Figure 2: SENEKA sensor network with AMFIS ground control station and mobile sensor carriers (UGV "Mustang" and UAV AirRobot).

The description used is modular and based on XML standard. Every component must have at least one description file that contains two parts: 1) capabilities of the component and 2) their relevant technical characteristics. On account of this information, other subsystems can interoperate directly with this component and use its propagated functions and abilities. The "Configurator" program can also set up the component, if needed. For complex components containing many sub-components and abilities, these descriptions are organized hierarchically and contain references to descriptions of sub parts, e.g., cameras or communication devices. On this occasion, the research about the necessary level of details, which permits an automatic configuration using the descriptions of the single components, is of central relevance. Is the description level too coarse, functions might not be correctly used or actions will lead to unexpected or undesirable reactions. On the other hand, a too detailed description produces abundance in information and needs a lot of expenditure in the generation of the descriptive data.

To explain the configuration concept, a very easy SENEKA system containing only SENEKA-Hub, ground control station and one UGV "Mustang" will be considered. Both, control station and Mustang, are open components and have XML descriptions which can be sent to SENEKA-Hub using the Registration Service in Figure 1. In Figure 3, a very simplified description of UGV Mustang is shown. The UGV contains many different sensors (laser scanner, cameras, magnetometer, GPS, etc.) and WLAN as control/data channel. It can contain additional cameras with additional up/downlinks, so that the actual description file should be sent by each registration.

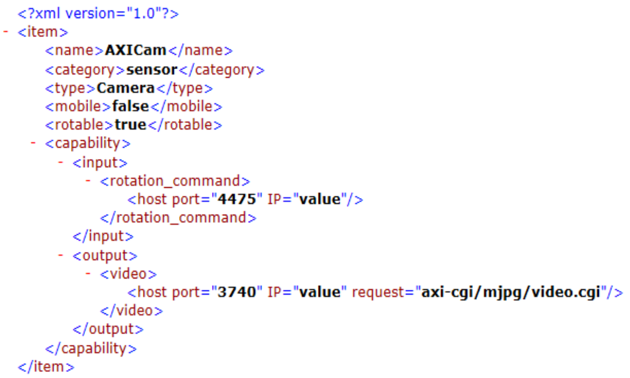


```
  <?xml version="1.0" encoding="UTF-8"?>
  <!-- This is mustang. Publishes odometry, reads Waypoints, Waypointlists and Joy. -->
- <asset>
    <id>mustang</id>
  - <ROS_information>
    - <topics>
        <topic type="nav_msgs/Odometry" name="seneka_wp"> </topic>
      </topics>
    </ROS_information>
  - <AddDevice>
    - <sensornetwork id="22">
        <uuid>a04e827e-77fc-48dd-83fe-bb8ad092d6e4</uuid>
        <name>Mustang</name>
        <desc>Mustang</desc>
        <connection>Aus</connection>
        <address/>
      - <sensornode id="106">
          <uuid>29be5ede-0020-4cfe-9cee-f5a0ec099010</uuid>
          <name>Mustang</name>
          <alias>Mustang</alias>
          <mobile>true</mobile>
          <position>49.01564;8.427166;1</position>
          <timeout>0</timeout>
          <virtual>false</virtual>
        - <sensor id="568">
            <uuid>348f32a1-d192-4c58-b7bc-3aff81b9b0c0</uuid>
            <priority>0</priority>
            <class>WEBCAM</class>
            <manufacturer>AXIS</manufacturer>
            <desc>AXI_M3025-VE.xml</desc>
            <threshold_min>0</threshold_min>
            <threshold_max>0</threshold_max>
            <threshold_warn/>
          + <attribute name="LOCK_ANGLE_H_MIN">
          + <attribute name="LOCK_ANGLE_H_MAX">
          + <attribute name="LOCK_ANGLE_V_MAX">
          + <attribute name="LOCK_ANGLE_V_MIN">
          + <attribute name="ZOOM_STEPS">
          - <attribute name="IP">
              <value>10.1.2.22</value>
              <unit>IP Adresse</unit>
              <desc>IP Adresse</desc>
            </attribute>
          </sensor>
        + <sensor id="566">
        + <sensor id="567">
        + <sensor id="563">
        + <sensor id="564">
        - <sensor id="565">
            <uuid>2b29b458-e8d9-49df-a879-5438e0c4ecb3</uuid>
            <priority>0</priority>
            <class>RECEIVER</class>
            <manufacturer>Unknown</manufacturer>
            <desc>DJI_24G.xml</desc>
            <threshold_min>0</threshold_min>
            <threshold_max>0</threshold_max>
            <threshold_warn/>
          </sensor>
        </sensornode>
      </sensornetwork>
    </AddDevice>
  </asset>
```

Figure 3: Simplified XML description of UGV "Mustang" for SENEKA-Hub.

The detailed descriptions of sub-components are in linked XML files (AXI_M3025-VE.xml for camera and DJI_24G.xml for additional wireless video link), so that the main file is not very complex and contains the information at the appropriate hierarchy level only. The UGV receives movement commands, waypoints or waypoint lists and can deliver a video stream from the on board camera – only capabilities on this level are needed for the ground control station. The files for mobile components can contain start coordinates, if needed (<position> in Figure 3), but actual coordinates will be reported dynamically.

The same AXI cameras can be also used as component of SENEKA on the same level as the UGV. Because their IP-addresses can be different, they must be configured. The IP-address is set with parameter "value" in the main file. The camera position is the same as position of UGV – that's why the parameter <mobile> is "false". The simplified description of the camera is shown in Figure 4.



```
  <?xml version="1.0"?>
- <item>
    <name>AXICam</name>
    <category>sensor</category>
    <type>Camera</type>
    <mobile>false</mobile>
    <rotable>true</rotable>
  - <capability>
    - <input>
      - <rotation_command>
          <host port="4475" IP="value"/>
        </rotation_command>
      </input>
    - <output>
      - <video>
          <host port="3740" IP="value" request="axi-cgi/mjpg/video.cgi"/>
        </video>
      </output>
    </capability>
  </item>
```

Figure 4: XML description of AXI camera (simplified).

The camera receives rotation commands on port 4775 and can deliver video streams from IP "value" (set to "10.1.2.22" in Figure 3) by setting the next request on port 3740: "AXI-cgi/mjpg/video.cgi".

To ensure correct configuration, the XML description must contain all information needed to set up wired and wireless communication. After configuration, system components communicate directly, as needed, on the basis of the information from the SENEKA-Hub – mostly with the ground control station.

External systems (for example, command centres) can get information from the SENEKA-Hub about available capabilities and request needed services directly by the components. They have to implement the access to information describing the capabilities available from the ROS SENEKA-Hub – this service has to follow the standard with appropriate descriptions.

III. AUTOMATIC CONFIGURATION CONCEPT

A particular challenge is the specification of modules' descriptions and reusable software supporting automatic configuration in complex heterogeneous distributed systems. To ensure a generic solution, suitable classifications of all input and output devices as far as software modules are needed. Output devices can be classified on values that they can represent, input devices on generated values. A sample of the sensor classification according to type of their information is shown in Table 1.

TABLE 1: SAMPLE OF SENSORS CLASSIFICATION ACCORDING TO SENSOR OUTPUT.

Dimension	Direction dependent	Sensors/Values
0	No	Temperature, pressure, gas, motion detector, non-directional microphone, voltage,
0	Yes	Distance, velocity, acceleration, force
1	Yes	Line scanning camera, microphone array
2	Yes	Video/IR camera, 2D laser scanner
3	Yes	Radar, 3D laser scanner, PMD, 3D Video/IR camera

To support automatic configuration, the classification of each sensor group must be expanded with possible types of information compression (for camera: uncompressed, MJPEG, MPEG2, MPEG4, H264, etc.), frame rate, resolution and interfaces' classification according to standard/protocol (both input and output, if suitable) like shown in Table 2.

TABLE 2: SAMPLE OF (SIMPLIFIED) 2D SENSORS CLASSIFICATION ACCORDING TO STANDARD/PROTOCOL.

Channel	Wired						Wireless		
Analogue	RCA						Downlink		
Digital	USB		Ethernet				Digital Downlink	WLAN	
	2.0	3.0	UDP	HTTP	RTSP	RTP	...	...	...

Sensor carriers' classification takes into account all possible movement types (position changing/direction changing/combination, degree of freedoms, etc.), but also possible movement control: position, velocity or acceleration for each degree of freedom and their limits; absolute/relative values; correlations between coordinates etc.

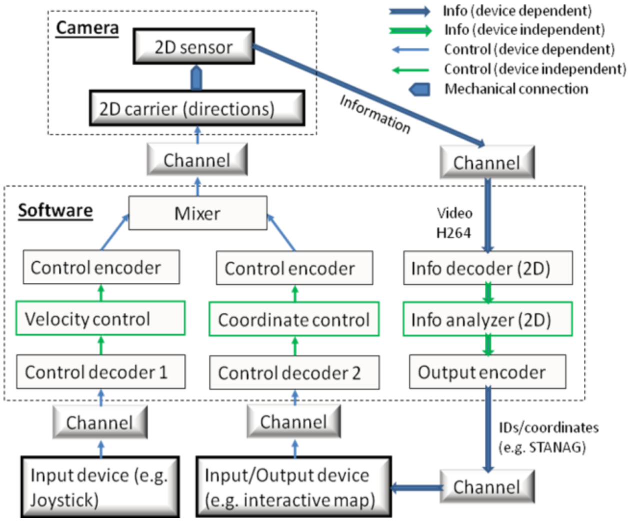


Figure 5: Automatic configurable information analyzing and control system for pan-tilt camera.

Based on the classification, parent classes for each type of components and standard interfaces can be developed. These classes must be device-independent. Device-dependent capabilities can be implemented with decoders and encoders for information, mixers and separators to prepare information for further analysis and transfer. All software modules must be also classified and suitable described with XML. A (simplified) sample of an automatic configurable modular control and information analyzing pan-tilt camera system is shown in Figure 5. Green modules are device-independent.

IV. CONCLUSION

In this article, a solution for dynamic configuration of distributed heterogeneous systems for disaster management has been described. The solution is based on standardized XML descriptions of components containing technical and non-technical details. This concerns a long-term installation, set up and restarting for reconfiguration, especially important in quickly alterable disaster situations where systems must switch fast between certain configurations for different duties or change configuration by exchanging or adding new components.

A sample of an automatic configurable pan-tilt camera control system based on devices' classification and of modular software architecture illustrates the concept of automatic configuration.

To implement adaptive automatic configuration with easy integration of new components and software generation, basic software modules must be defined and implemented. For the automatic software generation, the software has to be described by XML descriptions.

ACKNOWLEDGMENT

The authors would like to thank their colleagues and students, who have contributed to the work presented in this paper.

REFERENCES

[1] H.-B. Kuntze et al., "SENEKA - Sensor Network with Mobile Robots for Disaster Management," The twelfth annual IEEE Conference on Technologies for Homeland Security (HST 2012), Nov. 2012, pp. 406-410, doi:10.1109/THS.2012.6459883.

[2] H.-B. Kuntze et al., "Situation responsive networking of mobile robots for disaster management," Joint Conference of 45th International Symposium on Robotics and 8th German Conference on Robotics (ISR/Robotik 2014), June 2014, pp. 313-320, ISBN:978-3-8007-3601-0.

[3] I. Tchouchenkov, R. Schönbein, F. Segor and M. Kollmann, "Dynamic and Automatic Configuration of Distributed Heterogeneous Surveillance Systems," The 9th Security Research Conference (Future Security 2014), Sept 2014, pp. 164-171, ISBN: 978-3-8396-0778-7.

[4] Robot Operating System. [Online]. Available from: http://wiki.ros.org/ 2015.01.09

[5] A. Bürkle, F. Segor, M. Kollmann and R. Schönbein, "Universal Ground Control Station for Heterogeneous Sensors," Journal On Advances in Telecommunications, IARIA, Volume 3, Numbers 3 & 4, pp. 152 – 161, 2011.

[6] 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," The 6th International Conference on Systems (ICONS 2011), Jan 2011, pp. 72-76, 2011, ISBN: 978-1-61208-002-4.

[7] E. Santamaria, F. Segor, I. Tchouchenkov and R. Schönbein, "Path Planning for Rapid Aerial Mapping with Unmanned Aircraft Systems," The 8th International Conference on Systems (ICONS 2013), Feb 2013, pp. 82-87, ISBN: 978-1-61208-246-2.

A Dynamic FPGA-based Hardware-in-the-Loop Co-simulation and Prototype Testing Platform

Clément Foucher, Alexandre Nketsa

CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France
E-mail: {Clement.Foucher, Alexandre.Nketsa}@laas.fr

Abstract—**The base idea of co-simulation is to couple heterogeneous simulators in a single environment. This allows for choosing the best-suited simulator to represent each part of a complex system. Hardware-in-the-loop co-simulation introduces physical components in a software co-simulation. In this paper, we propose a hardware-in-the-loop co-simulation platform using a dynamically reconfigurable architecture on FPGAs. Main uses for this technology include introducing prototypes of digital systems directly in the simulation environment, and accelerating simulation by using FPGAs to implement models that can take advantage of parallelism. The platform was validated by coupling a C++ application with hardware modules loaded on-demand on a FPGA.**

Keywords*–*Co-simulation; Prototyping; Hardware-in-the-Loop; Reconfigurable Architectures.

I. Introduction

Co-simulation is the process of making multiple simulators collaborate in a larger simulation. In a co-simulation, each simulator is in charge of simulating a part of the system, and has an interface to exchange data and synchronize with its environment, which is constituted of other simulators. Using a co-simulation approach rather that simulating the whole system within a single simulator is an answer to growing simulation needs as systems get increasingly complex.

Indeed, in systems composed of many heterogeneous sub-systems, very different simulation needs can emerge. There are integrated tools, supporting most of these needs, such as COMSOL Multiphysics [1], which integrates a large number of modules to simulate a heterogeneous system. But as it is difficult to design software that answers each and every simulation need, the co-simulation paradigm seems a promising alternative approach. Profession-centric tools are also developed in that way, such as CNES' BASILES [2] co-simulation framework, which integrates different tools to provide a methodology targeting satellite simulation.

The separation of models in a co-simulation allows for easy replacement of a model by another, as long as the interfaces are preserved. This capability is well-suited for model-based approaches [3], in which models are progressively refined from a very high level description to a precise description of the target system. Extending this mechanism to system's prototype by replacing a model by its physical implementation introduces the Hardware-in-the-Loop (HIL) notion. HIL allows for test and validation in a fully managed environment. This notably makes possible replaying simulation scenarios to ensure the implementation behaves as expected and to compare prototype outputs with the one obtained from the higher level model. This is done by wrapping the prototype in an interface that allows simulators to send commands to the physical system through actuators, as well as sensors to provide data the other way.

Hardware implementations can be of many natures, but we focus here on digital hardware systems, e.g., systems on chips, or control-command parts of a system. Thus in the following, the term *hardware* will refer to logic circuits, by opposition to *software*. We use dynamically programmable hardware, taking advantage of System on Programmable Chip (SoPC) architecture introduced by Field-Programmable Gate Array (FPGA) devices.

In this kind of HIL co-simulation, the programmable hardware can be of two uses:

- Implementing a prototype of a digital system to join the simulation, replacing its higher-level software model,
- Implementing a model which can take advantage of parallelism to see its performances improved being executed on hardware rather than software.

These two different applications of SoPC HIL share a large part of the deployment approach.

In the former, we use a SoPC to implement a logic circuit prototype. FPGAs deliver various advantages for this consideration, the first being the reusability of the hardware. Indeed, nothing but development time is lost in case the circuit is faulty, as opposite as an error in a specifically made circuit. Moreover, programmable logic enhances high debug capabilities. It is easy to add observer logic that will not interfere with the actual purpose of the circuit, thanks to the high parallelism offered by the programmable resources.

In the latter, FPGA is just a computing resource, as those that can be found in High Performance Reconfigurable Computers (HPRCs) [4]. Indeed, hardware implementation enhances concurrent execution and parallelism. This generally improves the speed of algorithms, and may even achieve impressive speedups from single-core software execution for some algorithms. Generally, the more parallelizable is the algorithm, the more substantial is the speed gain.

What we propose here is a solution that achieves the following objectives:

- Easy integration and automated deployment of hardware modules in a distributed heterogeneous co-simulation,

- Dynamic loading/unloading of hardware modules thaks to partial reconfiguration,
- Intuitive interface definition for communication between modules,
- Automatic handling of FPGAs partial reconfiguration, turning a single chip in multiple independent programmable resources.

We developed a platform by combining an existing co-simulation solution, CosiMate [5], with a previous work on managing partially reconfigurable resources, the Simple Parallel platform for Reconfigurable Environment (SPoRE) [6].

SPoRE is a tool for handling FPGA-based computing resources. It allows executing computation kernels on distributed reconfigurable resources from a remote workstation. CosiMate is a co-simulation bus supporting various software simulation tools, and providing a communication and synchronization interface between them. By combining the SPoRE platform and the CosiMate environment, we set up a heterogeneous co-simulation environment using both software and dynamically reconfigurable hardware.

In the following, we begin by taking a look at existing solutions in Section II. Then we present our platform building blocks in Section III, the platform itself in Section IV, and how we build a co-simulation for this platform in Section V. Finally, we analyze the platform usages in Section VI and present the perspectives in Section VII.

II. Related Work

Connecting reconfigurable hardware logic to software in order to take advantage of both kinds of computations allows for powerful applications. This process is notably used in the rising generation of HPRCs [4], which are massive computing farms containing FPGAs tightly coupled to processors, the latter delegating intensive computation kernels to the former, acting as application-specific co-processors.

In [7], Liang *et al.* use a combination of MATLAB/Simulink and Xilinx System Generator (XSG) to communicate between software and hardware. Their co-simulation process follow the MDE guidelines [3], beginning by simulating a module, then coding it in HDL and finally executing it on a FPGA as a HIL process. Nevertheless, the communication between software and hardware is relying on proprietary protocols inherent to the tools. Using this solution, there is no control on how the FPGA is handled by XSG, which is done in a static way. This is the major difference with our solution, which notably allows partial reconfiguration of a FPGA, thus allowing for multi-IP design on a single FPGA chip. Moreover, our tool is able to handle multiple FPGA running in parallel on a network, thus multiplying the available resources.

Liao *et al.* present a coupling technique between a HDL simulator and a hardware module running on FPGA [8]. Their solution implements an efficient synchronization technique as hardware clock signal is generated from software, allowing for synchronous operations. Nevertheless, this solution prevents from taking full advantage of the speedup allowed by hardware implementation. Moreover, this is not clear how the communication ports, on both hardware and software, are generated: is this an automatic process or does ports have to be manually tailored. In our solution, hardware ports rely on bus-based registers, while software ports are based on CosiMate formalism.

An important part of introducing a prototype in a HIL co-simulation is to be able to reproduce on the hardware the exact stimuli applied to the corresponding software model. In [9], authors instantiate a hardware module on a FPGA, and link it to a testbench in a simulator. This technique allows for simulating a HDL design in a simulator, and then deport the design itself on a reconfigurable device while preserving the test scenario. They use the SCE-MI API [10] to communicate between a HDL simulator and hardware implemented on FPGA. While SCE-MI is a very interesting approach for heterogeneous communication, it does not handle the hardware deployment, where our solution allows for automatically handling FPGA configuration using partial reconfiguration.

III. HIL Co-Simulation Platform Base Blocks

To build the co-simulation platform, we combined two existing platform. On one hand, we used the CosiMate software [5] from ChiasTek, a co-simulation bus allowing putting together various simulators. On the other hand, we extended the SPoRE platform [6], previously developed by our means, allowing remote control and managed reconfiguration of FPGA-based nodes through a network.

CosiMate, as shown on Figure 1, is a bus on which standard simulators are plugged. CosiMate offers a standard interface through ports and synchronization mechanisms for simulators.

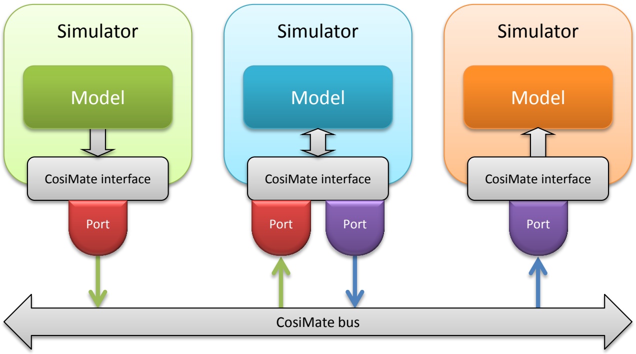


Figure 1. CosiMate bus architecture.

A port is defined with a direction and a data type, e.g., an output integer port on an IP will provide 32-bit data to the bus. Ports can also be defined as arrays to allow transferring blocks of data with larger size than basic types.

A simulator needs an extension library to be compatible with CosiMate. In such a simulator, user can define ports using a specific syntax, which will allow for generation of a XML-based description. A CosiMate project gathers these description files and allows links to be made between ports.

An output port from a simulator can be linked to one or various input ports of the same type and size of another simulator. Then, launching the co-simulation requires all the simulators to be running, and CosiMate automatically handles communication between ports.

Two modes are supported by CosiMate: synchronized and event-driven. In synchronized mode, all simulators wait on a barrier at each simulation step, the barrier being released by CosiMate environment when all simulators have reached it. Communication synchronization is done at each simulation step. This mode enables simulation time to be managed.

In event-driven mode, data flow though ports without any barrier, thus the synchronization process is up to the user. Moreover, event-driven mode also requires user to define a protocol for communication, as there is no time step to indicate when data is available.

The SPoRE platform is a distributed-node platform, which nodes contain FPGAs and are linked by a network, as displayed on Figure 2. The arrows indicate which node initiates the communication.

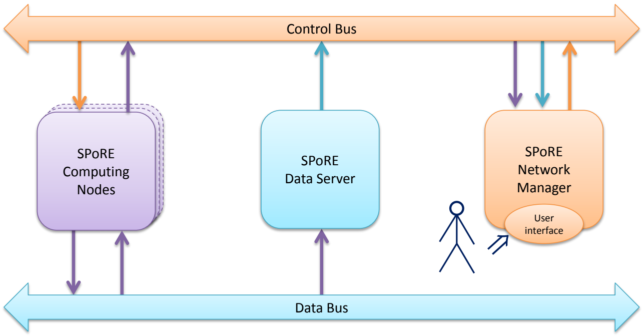


Figure 2. SPoRE buses representation.

As communication between nodes relies on Sockets, the network itself is abstracted, and could take different shapes. In our case, we use an Ethernet-based network. The SPoRE platform contains one or more computing nodes, at least one data server, and exactly one network manager node, from which the user commands the platform.

SPoRE uses a XML description to build an application. SPoRE applications are wrappers that indicate how to use FPGA partial configuration files implementing IPs. SPoRE partial reconfiguration mechanism and node management are discussed in [6] and [11].

When user already has a HDL description of needed computing kernels, a simple vision of SPoRE by user is as follows:

- The user describes computation kernels as black boxes containing in/out ports,
- The user writes an application by describing which kernels are to be used, and for each one, associate ports to application's *message paths*,
- The application description (XML), the kernels descriptions (XML) and the FPGA partial configuration files implementing kernels (binary) are stored on a data server,
- The user launches the application from the network manager,
- The user retrieves the results from the data server.

SPoRE automatically handles application description and bitstream download on computing nodes, does the reconfiguration process, download and compute data, and upload results to the data server.

IV. Building the Platform

The way we chose for linking CosiMate and SPoRE was to use the base representations of each platform to build a bridge. The bridge should then be viewed as a simulator by the CosiMate environment and as a computing node by the SPoRE platform, as presented on Figure 3. CosiMate supports C/C++ written simulators, and SPoRE only needs Socket support to declare a node. We then choose to build the bridge using C++ and Qt, to enhance portability.

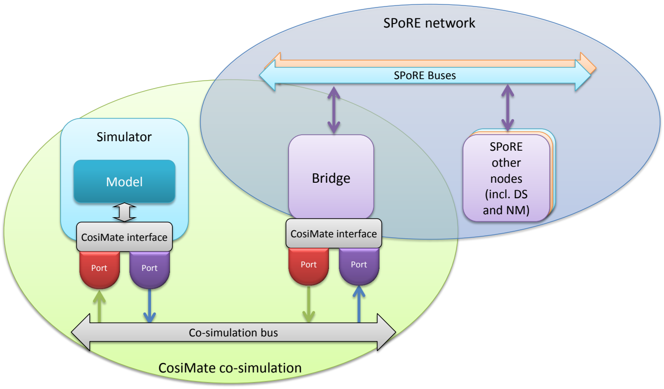


Figure 3. Bridge between CosiMate and SPoRE.

SPoRE platform has been extended to support the bridge. First, we replaced the scheduler to allow IP reconfiguration being done depending on the co-simulation needs. Due to SPoRE modular nature, this replacement does not override the previous scheduler, the scheduler choice is proposed to the user. The scheduler dynamically instanciates the modules when needed in co-simulation, and can erase them when not needed any more. The reconfiguration process itself relies on SPoRE cababilities and is transparent to the user.

In SPoRE application descriptor, we added a reference to a co-simulation descriptor. This reference is ignored by common SPoRE nodes, but can be interpreted by the bridge. The co-simulation descriptor indicates the relation between SPoRE message paths and CosiMate ports. Using this file, the bridge dynamically creates CosiMate ports, allowing for the generation of the CosiMate configuration file. The CosiMate ports are thus dynamic, and depend on the application. There is no need to write a specific XML configuration file for CosiMate, this is done automatically to match SPoRE IP ports.

We use the co-simulation environment in event-driven mode, using a simple request/acknowledge protocol for data handling, as shown on Figure 4. Indeed, when declaring an output (resp. input) CosiMate port, the bridge actually declares two additional single-bit ports, one output (resp. input) for request, and one input (resp. output) for acknowledge.

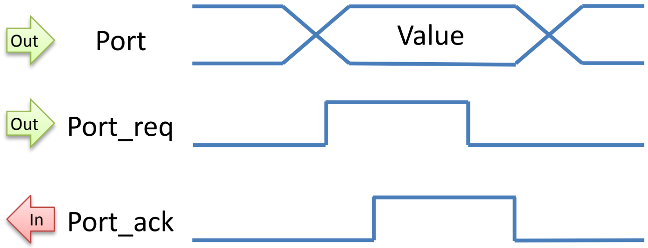


Figure 4. Bridge protocol for an output port.

In SPoRE environment, messages are referenced by a data server. SPoRE uses a *message path* mechanism to link messages to kernels. Message paths are FIFOs containing messages identifiers (IDs) and owners.

In an application description, each kernel port actually used in the application is linked to a message path. All messages produced by an output port will be declared by node to data server in the corresponding message path using a unique message ID. Conversely, when a message is available in a message path linked to an input port, the node will be advised by data server, that will indicate which node hosts the message. Messages are then downloaded directly between nodes. Note that if a node hosts both ends of a message path, the data server will still manage the message path to ensure data coherency, but no download will be necessary. This message path mechanism was initially added to SPoRE while building this bridge, but is now the default behavior for port management.

When an event comes to the bridge from CosiMate, a SPoRE message is dynamically generated, and SPoRE data server is advised of that creation. If a SPoRE node is listening on the according message path, it will then be able to download the message from the bridge. Conversely, the bridge listens on output messages path linked to CosiMate ports. When a message is produced in such a message path, the bridge is advised by SPoRE data server, and will download the message from the producer node, eventually initiating an event on corresponding CosiMate port to transmit the message content.

V. Implementing Models as Hardware

Following the model-based prototyping method [3], the co-simulation process begins by simulating high-level models of the system. The model is then progressively refined, until a HDL implementation is done. This implementation can be done manually by describing the IP core in HDL language, or make use of model-to-text [12] or high-level synthesis tools [13] to generate the code.

Second phase would be to simulate this code using a HDL simulation tool plugged to the co-simulation bus. ModelSim [14] is an example of tool supported by CosiMate. Some minor signal adaptation may have to be done at this phase, as data type representation can differ between software and hardware, notably number of bits used to represent a signal. Afterward the results of this simulation should not differ from the simulation using high-level models, or the error between simulations should be in an acceptable range to validate the implementation [15].

Finally, the synthesis of the hardware module has to be done following SPoRE bus-based architecture. SPoRE implements computing kernels as *cells* plugged to a bus, which notably allows IPs for direct access to data in RAM. This bus-based interface will generally be easy to implement for most IPs. In certain cases however, this mandatory interface will cause some architectural restriction. Typically, this can be the case of some IPs requiring to be fed two or more messages at the same clock cycle. Nevertheless, this can be easily worked around by adding small controllers that will store messages and write them to the IP when all are available.

When correctly wrapped in a SPoRE cell interface, the user describes the IP ports and their protocol using SPoRE descriptor syntax. This allows for use in any SPoRE application, including co-simulation applications by instantiating the bridge.

This integration can be seen on multiple levels. As the bridge concentrates all data exchanges between the two platforms, each platform can be seen by the other as *being* the bridge. This means that we can see the complete platform as a CosiMate co-simulation integrating a FPGA-based simulator, or as a SPoRE platform integrating software simulators. Moreover, due to SPoRE distributed nature and CosiMate allowing multi-host co-simulation, we can easily integrate two or more SPoRE platforms by using as many bridges as necessary, in order to overcome a bottleneck if needed.

For now, the bridge only supports integer transmission between the platforms. If data size between a SPoRE message path and a CosiMate port does not match, a stack is automatically defined. As an example, we tested a FPGA-based AES encryption that requires 128-bit word length, and coupled it with 32-bit integer ports in CosiMate. To do that, the bridge waits for 4 messages from the co-simulation environment before it generate a SPoRE message. Conversely, a SPoRE message will generate 4 successive CosiMate messages when received. Another way to handle this difference is to force data size matching by using CosiMate array mechanism, and treat an array of 4 integers as a single 128-bit message.

The test scenario we built is an AES encoder/decoder prototype testing. The hardware part contains two modules: an encoder and a decoder. The software part consists in a small C++ software that allows for selecting a local file and choosing whether to encrypt or decrypt it. The software module then emits the data words composing the file one after the other as events on the CosiMate bus. The hardware part of the platform then automatically instanciates the required module, reads the inputs, and sends the outputs back to the bridge. The outputs are used by the software application to build a new file. This test application allowed us to validate the hardware/software communication part as well as the dynamic behavior.

VI. Platform Uses

This platform can be used for different purpose. As explained in a previous section, it allows accelerating a simulation by implementing highly parallelizable models in hardware, as well as testing a logic prototype in the same environment its higher-level models were tested.

In both cases, debug features are of matter, as it is important to obtain information from inside a model. This can be easily done by adding Embedded Logic Analyzers (ELA) in the FPGA. But using SPoRE also enables one to add its specific observers inside the IP. Data will then be retrieved from additional ports and uploaded to the data server. Doing so will use Ethernet bandwidth, which may interfere with IP if communication timing is of matter. But this allows observability of the model without needing a specific debug connection such as JTAG. This is especially useful when using multiple FPGAs to deploy models, in which case it is difficult to have specific debug links to all devices.

Compared to other solutions depicted in Section II, this platform adds support for partial reconfiguration. This allows seeing an FPGA as a real SoPC, in which IPs are independent from each other. This means if the simulation needs some model at one point of the simulation, and some other model at a different time, we can use both models on the same device even if there is not enough logic to handle both models at the same time. This is done by reconfiguring the FPGA, replacing the unused model.

Moreover, by implementing a timeslicing scheduling approach, we could do so even when both models are needed at the same time. Timeslicing is a technique used in software to simulate application parallelism by attributing one resource (processor core) to different tasks depending on the time. Here, resources are reconfigurable logic, but this can be done the same way. However, this case needs specific cares. This will be discussed in Section VII.

But the most promising use for this platform is for modelling of dynamic systems. Indeed, by adding SPoRE into the co-simulation environment, we provide support for these systems. As for now, Partial Dynamic Reconfiguration (PDR) is only used for resources management. But we can imagine extending the PDR management to the SPoRE application itself. This can be done by adding explicit reconfiguration directives in the SPoRE application. Using this, we will be able to simulate the behavior of dynamic systems using native FPGA technology.

VII. Conclusion and Future Work

In this paper, we presented a co-simulation platform allowing to put together software and hardware models. The main point of this platform is that is handles partial reconfiguration of FPGAs to turn these in dynamic multi-IP designs. One FPGA can then handle various model implementations at the same time, and the models implemented can vary during the co-simulation. Moreover, various FPGAs can be used at the same time if more reconfigurable logic is needed. All reconfigrations are handled automatically without any need for the user to be aware of the partial reconfiguration mechanism.

The HIL co-simulation platform depicted in this article is only a first step in our research projects. One main extension we would like to do is to support synchronized mode up to the FPGA. This will be the object of further work, and we are exploring several leads on the subject. One idea would be using a clocking process on hardware IPs that is independent of the real time hardware clock, rather being provided by the co-simulation environment, as done by Liao *et. al* [8].

Moreover, we would like to automatize the process of designing SPoRE hardware kernels based on HDL IPs. This could be done using a parsing tool that scans the HDL top level entity interface, and generates both a co-simulation interface for software simulation, and a SPoRE wrapper for integration into HIL co-simulation. Some interface issues should also emerge from this, notably type conversion handling and floating/fixed point values representation. Using standard interface definitions, such as IP-XACT [16] can help automating the integration process.

Support of timeslicing simulation will also be investigated. Indeed, if the user is stuck with a number of FPGAs, which does not allow for implementing all models at the same time, timeslicing can solve this issue. It would consist in instantiating one model on the resources, treat the data related to it, then replace it by another model and do the same. In synchronized mode, this would allow to deal with multiple models on the same resources at each simulated time step. The downside of this approach is the reconfiguration time. Indeed, the reconfiguration time is an uncompressible overhead in an IP lifetime. This overhead gets negligible when the IP use time growth, but is of matter if the reconfiguration is frequent.

This overhead then needs to be taken in consideration if hardware is used for better performances. However, if the hardware is used for implementing prototypes, with no considerations of performances, this can be a useful approach.

Finally, this platform has a potential for the simulation of dynamic systems. If PDR is now used in background by SPoRE, we could make it explicit. This approach will concentrate our efforts, as we see here a promising use of the platform. Indeed, if the user is able to indicate how the reconfiguration should be handled, this opens new perspectives for the simulation of dynamic and auto-adaptive systems. To do so, we need to extend the SPoRE scheduler used for co-simulation to allow direct reconfiguration orders from the application itself.

References

[1] COMSOL Inc. (2015). Comsol multiphysics, [Online]. Available: https://www.comsol.com, [retrieved: March, 2015].

[2] F. Quartier and F. Manon, “Simulation for all components, phases and life-cycles of complex space systems.,” in CSDM (Posters), 2013, pp. 167–174. [Online]. Available: http://ceur-ws.org/Vol-1085/15-paper.pdf, [retrieved: March, 2015].

[3] A. G. Kleppe, J. B. Warmer, and W. Bast, MDA explained: the model driven architecture: practice and promise. Addison-Wesley Professional, 2003, ISBN: 978-0321194428.

[4] T. El-Ghazawi, E. El-Araby, M. Huang, K. Gaj, V. Kindratenko, and D. Buell, "The promise of high-performance reconfigurable computing," IEEE Computer, vol. 41, no. 2, pp. 69–76, Feb. 2008, ISSN: 0018-9162. DOI: 10.1109/MC.2008.65.

[5] ChiasTek. (Feb. 2015). Cosimate, [Online]. Available: http://www.cosimate.com, [retrieved: March, 2015].

[6] C. Foucher, F. Muller, and A. Giulieri, "Online codesign on reconfigurable platform for parallel computing," Microprocessors and Microsystems, vol. 37, no. 4–5, pp. 482–493, 2013, ISSN: 0141-9331. DOI: 10.1016/j.micpro.2011.12.007.

[7] L. Guixuan, H. Danping, J. Portilla, and T. Riesgo, "A hardware in the loop design methodology for fpga system and its application to complex functions," in VLSI Design, Automation, and Test (VLSI-DAT), 2012 International Symposium on, Apr. 2012, pp. 1–4. DOI: 10.1109/VLSI-DAT.2012.6212666.

[8] Y. B. Liao, P. Li, A. W. Ruan, Y. W. Wang, and W. C. Li, "A hw/sw co-verification technique for field programmable gate array (fpga) test," in Testing and Diagnosis, 2009. ICTD 2009. IEEE Circuits and Systems International Conference on, Apr. 2009, pp. 1–4. DOI: 10.1109/CAS-ICTD.2009.4960748.

[9] C.-Y. Huang, Y.-F. Yin, C.-J. Hsu, C.-Y. Huang, and T.-M. Chang, "Soc hw/sw verification and validation," in Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific, Jan. 2011, pp. 297–300. DOI: 10.1109/ASPDAC.2011.5722202.

[10] Accellera Systems Initiative, Standard co-emulation modeling interface (sce-mi) reference manual, 2014. [Online]. Available: http://www.accellera.org/downloads/standards/sce-mi, [retrieved: March, 2015].

[11] C. Foucher, "Méthodologie de conception pour la virtualisation et le déploiement d'applications parallèles sur plateforme reconfigurable matériellement," PhD thesis, Sciences et Technologies de l'Information et de la Communication, Oct. 2012. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00777511, [retrieved: March, 2015].

[12] D. Foures, V. Albert, J.-C. Pascal, and A. Nketsa, "Automation of sysml activity diagram simulation with model-driven engineering approach," in Proceedings of the 2012 Symposium on Theory of Modeling and Simulation-DEVS Integrative M&S Symposium, ser. TMS/DEVS '12, Orlando, Florida: Society for Computer Simulation International, 2012, 11:1–11:6, ISBN: 978-1-61839-786-7. [Online]. Available: https://dl.acm.org/citation.cfm?id=2346616.2346627, [retrieved: March, 2015].

[13] P. Coussy and A. Morawiec, High-level synthesis, From Algorithm to Digital Circuit. Springer Netherlands, 2008, ISBN: 978-1-4020-8587-1. DOI: 10.1007/978-1-4020-8588-8.

[14] Mentor Graphics. (Feb. 2015). Modelsim, [Online]. Available: http://www.mentor.com/products/fpga/model/, [retrieved: March, 2015].

[15] U. Fahrenberg and A. Legay, "Generalized quantitative analysis of metric transition systems," in Programming Languages and Systems, ser. Lecture Notes in Computer Science, C.-c. Shan, Ed., vol. 8301, Springer International Publishing, 2013, pp. 192–208, ISBN: 978-3-319-03541-3. DOI: 10.1007/978-3-319-03542-0_14.

[16] IEEE Computer Society, IEEE standard for IP-XACT, standard structure for packaging, integrating, and reusing IP within tool flows, version 1685-2014, Jun. 2014. [Online]. Available: https://standards.ieee.org/getieee/1685/download/1685-2014.pdf, [retrieved: March, 2015].

A Contribution to the Evaluation of NAND Flash Memory

Jaroslav Kadlec, Radek Kuchta, Radovan Novotný, Zdeňka Kuchtová
Central European Institute of Technology
Brno University of Technology
Brno, Czech Republic
email: jaroslav.kadlec@ceitec.vutbr.cz; radek.kuchta@ceitec.vutbr.cz; novotnyr@feec.vutbr.cz;
xkucht06@stud.feec.vutbr.cz

***Abstract*— NAND flash memories are well known for their uncomplicated structure, low cost, and high capacity. Their typical characteristics include architecture, sequential reading, and high density. NAND flash memory is a non-volatile type of memory and has low power consumption. The erasing of NAND Flash memory is based on a block-wise base. Since cells in a flash chip will fail after a limited number of writes, limited write endurance is a key characteristic of flash memory. There are many noise causes, such as read or program disturbances, retention process, charge leakage, trapping generation, etc. Preferably, all errors in the storage would be adjusted by the ECC algorithm. The conclusion of all mentioned parasitic factors creates a set of external and internal influences which affects variable behavior of memory in time. To prepare an overall analysis of all the important factors that affect the reliability and life-cycle endurance of NAND flash memories and describe the methodology for their evaluation was our main motivation for this paper.**

Keywords- flash memory; non-volatile; bit error rate; error correction code; architecture; reliability.

I. INTRODUCTION

Flash memory has been an important driving force due to the increasing popularity of mobile devices with large storage requirements. Flash memory is respected in many applications as a storage media due to its high access speed, non-volatile type of storage, and low-power consumption. There is a wide range of non-volatile memories, and they all give various characteristics based on the complexity of array organization and structure of the selected cell type [1]. General comparison of NAND and NOR Flash memories is in the Table 1.

Flash memories are becoming widely deployed in many applications, such as solid state drives (SSDs) for embedded controllers and traditional computing storage. NAND Flash memories are becoming more and more popular due to their usage as Solid-State Drives (SSDs) and USB Flash drives which are in general called Flash storage devices.

Another area of application is systems, which allow system reconfiguration, software updates, changing of stored identification codes, or frequent updating of stored information (i.e., smart cards). Electrically erasable and programmable read-only memories (EEPROM’s), which are electrically erasable and programmable, will be produced only for specific applications, because they use larger chip areas and are more expensive.

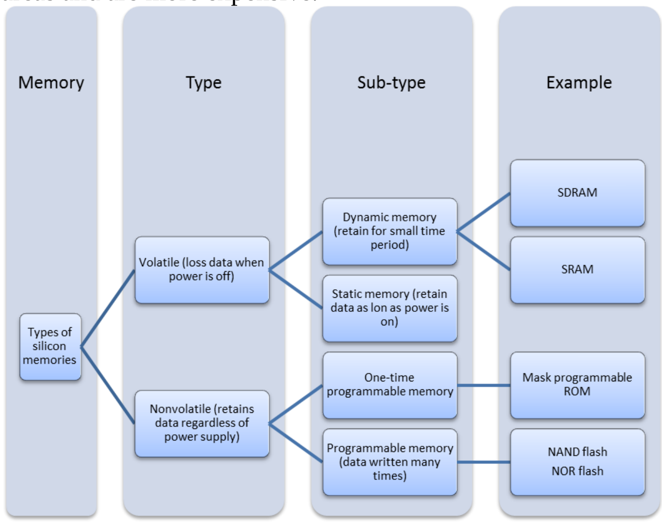

Memory
Type
Sub-type
Example
Types of silicon memories
Volatile (loss data when power is off)
Nonvolatile (retains data regardless of power supply)
Dynamic memory (retain for small time period)
Static memory (retain data as lon as power is on)
One-time programmable memory
Programmable memory (data written many times)
SDRAM
SRAM
Mask programmable ROM
NAND flash
NOR flash

Figure 1. Flash memories as a type of memory device characterized by non-volatility

Following on from these advantages, the manufacturers of memories started to consider the role of flash memories for a new range of applications. These include hard disk caches, solid-state drives, mobile sensor networks, and data-centric computing. Many microcontrollers have integrated flash memory for non-volatile data storage. Flash memory is also used in many applications where data retention in power-off situations and reliability are crucial requirements, such as in embedded computers or wireless communication systems.

Nowadays, flash memory is one of the most popular, reliable, and flexible non-volatile memories to store constant data values and software code. NAND Flash architecture and NOR Flash architecture dominates the non-volatile Flash market [3]. NAND flash requires protracted access time for the data and subsequent access to any non-consecutive location could be problematic. Nevertheless, once a page of memory is released for read, data can be pulled out from the memory fast, but in general. This is the reason why it was rarely used as the main memory of the system. As a result, there must be a controller to access data which is important in order to manage all the essential tasks of accessing NAND Flash device effectively [2]. General division of the Flash memory types is in the Figure 1.

TABLE I. THE MAJOR DIFFERENCES BETWEEN NAND AND NOR FLASH MEMORY

	NAND	NOR
Memory cell arrangements	Cells are arranged in series with the adjacent cells sharing source and drain.	Cells are arranged in parallel with all the source node of the cells connected to the bit line.
Capacity	tens of Gbits	several Gbits
Non-volatile	Yes	Yes
Interface	I/O interface	Full memory interface
High-speed access	Yes	Yes
Access method	Sequential	Random byte level access
Page mode data access	Yes	No
Performance	Fast read (serial access cycle) Fast write Fasted erase	Fast read (random access) Slow write Slow erase
Price	Low	High
Life Span	10^5-10^6	10^4-10^5
Write cycles	10^6	10^6
Advantages	Fast programing and erasing	Random access, possible byte programing
Disadvantages	Slow random access, difficult byte programing	Slow programing, slow erasing
Typical uses and applications	Storage, file (disk) applications, voice, data, video recorder and any large sequential data archiving	Networking device memory, replacement of EPROM, applications executed directly from non-volatile memory

Due to the non-volatile nature of this storage media, there is a high demand for it in the mobile communication industry. Flash memory has become the most popular choice for mobile devices. NAND Flash memory is commonly found in portable or embedded memory for computers, digital cameras, mobile phones, MP3 players and other devices where data is generally written or read sequentially [4].

II. GENERAL NAND FLASH DEVICE ARCHITECTURE

The overall architecture of the NAND flash device is shown in Figure 2. Unlike most memory technologies, NAND flash is ordered in pages which are written and read as a unit. The elementary unit of operation for a NAND Flash device is one page of data with control commands of the whole block (multiple pages) or the whole chip [2]. Therefore, data can be written only to one page at once. A page is defined as cells linked with the same word line. This is the smallest programmable unit physically made up of a row of cells.

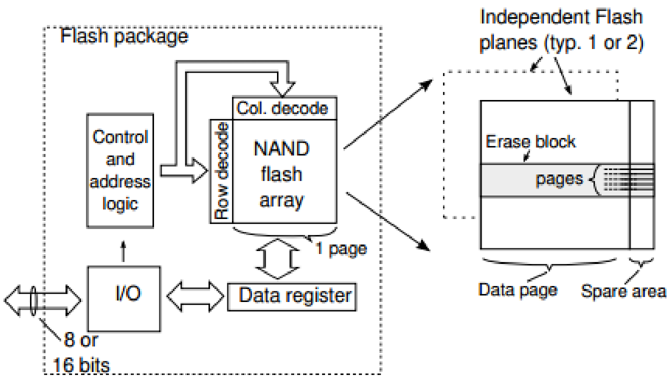


Figure 2. Architecture of the NAND Flash Device [4]

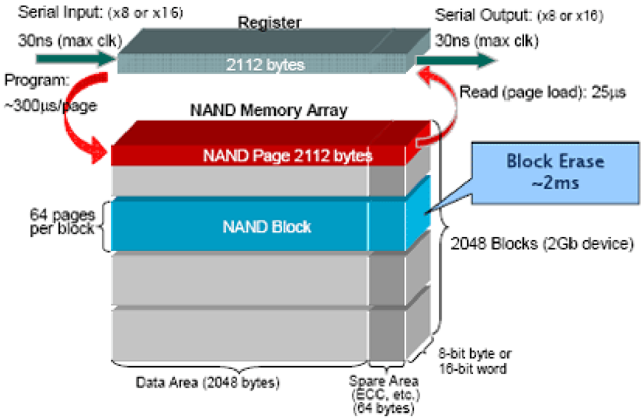


Figure 3. The 2-Gbit NAND device is ordered as 2048 blocks, with 64 pages per block [5].

NAND Flash devices could be considered as large page and small page devices [2]. There are overall 528 bytes (264 words) per small page. For enormous capacities, typically 1 Gbit and more, a large page is used. A large page device usually has 2048 bytes of data and 64 bytes of spare data per page (Figure 3) while a small page device has 512 bytes of data and 16 bytes of spare data per page. The commands sequence for large page and small page devices are different so the controller must be aware of which kind of device is being used.

Cells are organized in pages, and each page is divided into a data area, also named as a “Cell Array” page area, and a redundant area as a spare area for system overhead functions, also named as a “Spare Cell Array” page area. Spare blocks are set apart from the flash storage for remapping bad sectors. This solution prolongs the useful life and reliability of the flash storage device. The spare columns are fully addressable by the user and are typically used for storing Error Correction Code (ECC), wear-leveling, and other organization of information in order to improve data integrity. In operation, bytes from the spare area are equivalent to bytes from the data area and can be used to store the user's data. The spare area is not physically different from the rest of the page.

Before programming, a page must be erased which sets all data bits to “1”. Then, only the value “0” can be programmed into each cell. An erased, blank page of NAND flash has no charges stored in any of its floating gates. Unlike block-oriented disk drives, nevertheless, pages must be erased in units of erase blocks including multiple pages (typically 32 to 128) before being re-written.

III. Error Correction Code in NAND Flash Memories

In digital communication, the quantity of bit errors is the number of received bits of a data stream sent over a communication channel that have been changed due to interference, noise, bit synchronization errors or distortion. The bit error rate or bit error ratio (BER) is the number of bits that have errors divided by the total quantity of transmitted bits throughout a given time interval. BER is a unitless measure, frequently formulated as a percentage. The raw bit error rate relates to the probability of a bit error occurring in an individual bit cell on a flash device [6].

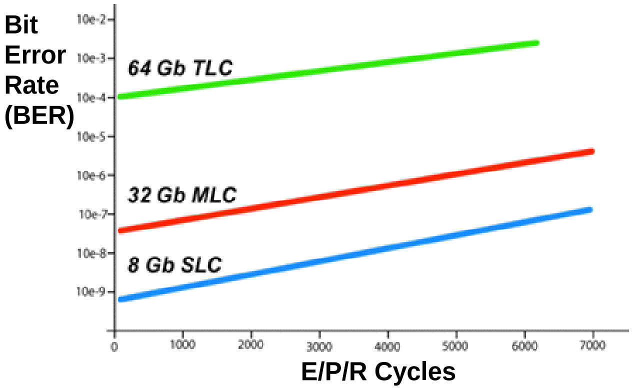


Figure 4. Bit error rate versus Erase/Program/Read cycles for Micron NAND flash [7]

A. Noise sources in NAND flash and the bit error rate (BER)

There are many noise causes existing in NAND flash, such as cell-to-cell interference, read or program disturbances, retention process, random-telegraph noise, background-pattern noise, charge leakage and trapping generation, etc. [8]. Such noise sources considerably shrink the storage reliability of flash memory. Over time the quantity of affected cells increases, see Figure 4. Figure 5 shows that Read Disturbances Error Rate is empirically much worse in devices that have consumed erase, program and read cycles than in uncycled devices [7].

Bit errors are a natural consequence of uncertainty when executing any data storage and must be moderated by software or hardware so that the integrity of the original information is not compromised [6]. For NAND flash, this is implemented by using protecting groups of bits with a higher-level error correction algorithm.

Preferably, all errors in the storage would be adjusted by the ECC algorithm. In reality the algorithm protects against a range of errors that are probable to happen.

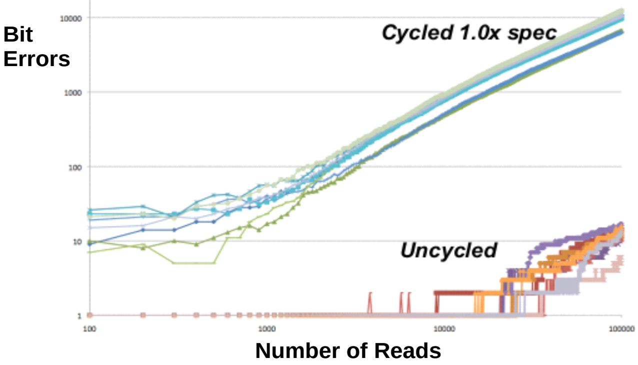


Figure 5. Bit errors versus number of reads [7]

TABLE II. ECC bit correction requirements for SLC and MLC NAND flash memory [9]

	NAND Process	ECC required	Erase Cycle	Data Retention
SLC	70/60 nm	1-bit	100 K	10 years
	50 nm	1-bit	100 K	10 years
	40/30 nm	4-bit	TBD	10 years
MLC	70/60 nm	4-bit	10 K	10 years
	50 nm	4 ~ 8 bit	5 K ~ 10 K	10 years
	40/30 nm	12 ~ 24 bit or more	3 K ~ 5 K	5 years

Over time NAND flash has augmented storage density by storing more bits per cell and moving to smaller geometries. As NAND Flash memory moves towards more progressive process nodes, the cost of devices is decreasing, but the cells become more vulnerable [10]. The quantity of bits kept per cell is increasing, bit values are represented by smaller voltage ranges, generating more uncertainty in the value stored in the bit cell due to more ambiguity in the amount of charge [6]. As the bit cells get smaller, the individual cells are more vulnerable to failure brought by high-voltage stress because fewer electrons can be trapped in the floating gates. The effect is to narrow the valid voltage ranges for a given value, increasing the probability for program and read disturbances. Since this solution requires higher levels of error correction mechanism in order to ensure the integrity of the data on the flash device, the new technology needs more ECC (Deal, Hamming, RS, BCH, LDPC)[11]. Overview of requirements of ECC for SLC and MLC NAND Flash memory is in the Table 2.

The accepted uncertainty upsurges the probability for data to be stored or read incorrectly, requiring higher levels of error correction for MLC flash than for SLC flash [6]. Devices using NAND flash must integrate very high levels of error correction in order to guarantee support for next generation flash devices – see Figure 6.

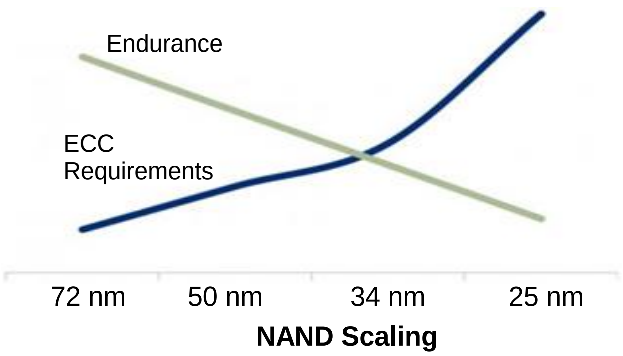


Figure 6. The drawbacks of NAND scaling: decreasing endurance, increasing ECC [12].

A one-bit ECC algorithm is capable of correcting one failure bit per 512 bytes. SLC flash is able to work with single-bit correction over 512 byte sectors because the individual bit error rate is really low.

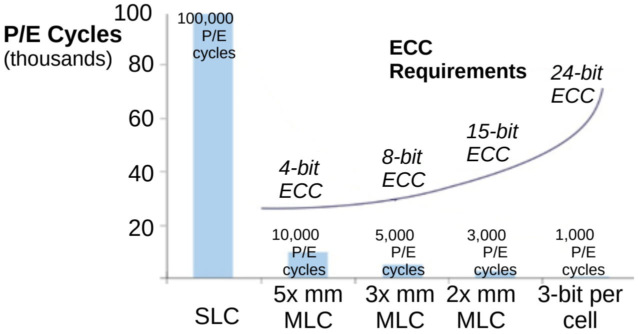


Figure 7. ECC and a life cycle comparison of NAND flash by process node: increase in correction capability is not enough to maintain endurance of the cell [13]

MLC flash has required more powerful correction algorithms capable of correcting four to eight bits to manage the higher bit error rates arising from the greater uncertainty of charging and to detect the various voltage ranges in a single bit cell (see Figure 7) [6].

B. Error detection and correction in NAND Flash Memories

The ECC permits data that is being read or transmitted to be checked for errors and, when necessary, corrected. ECC is a worthy way to recover the incorrect value from the residual good data bits [2]. Error detection and correction or error control includes techniques that permit reliable transfer of digital data by the detection of errors and reconstruction of the original, corrected error-free data. If the ECC cannot correct the error throughout read, it may still detect the error. The application of ECC is used with NAND flash parts to compensate bits that could fail during device operation. On-chip ECC resolves many supposed complications of working with a NAND solution [14]. Currently the error correction is an integral part of the NAND flash that guarantees data integrity.

Up to now, more error correction has been required for MLC NAND technology, whereas SLC NAND has characteristically required only 1-bit ECC for densities up to 4 Gb fabricated at 43 nm [10]. Current trends in the NAND flash market resulting to changes that must be made in the error correction algorithms to preserve the integrity of data stored in next-generation NAND flash devices [6]. The SLC NAND Flash devices, fabricated at 32 nm or 24 nm, require 4-bit or 8-bit ECC, respectively, per 512 bytes [10].

IV. NAND Flash ECC Algorithms

NAND Flash devices need appropriate error correction algorithms to diminish errors that occur during the programming and read operations [6]. The Life span of NAND Flash could be prolonged without more ECC bits due to the especially proposed operation algorithm. Error detection is usually realized using an appropriate hash function or checksum algorithm. A hash function adds a fixed-length tag to a data, which can be whenever recalculated and verified.

The basic system of ECC theory is to enlarge some redundancy for protection. The redundancy permits the receiver to detect a limited number of errors that may happen anywhere in data, and usually to correct these errors without retransmission. Different ECC techniques are necessary in various types of flash memory.

ECCs are typically divided into two classes: block codes and convolution codes. Hamming codes, Bose-Chaudur-Hocquenghem (BCH) codes, Reed-Solomon (RS) codes, and Low-density parity check (LDPC) codes are most notable block codes and have been widely used in communication, optical, and other systems [8]. The choice of the most effective correction code is a compromise between the number of symbol errors that need to be corrected and the additional storage requests for the generated parity data. Early designs implementing SLC NAND used either no error correction or marginally correcting Hamming codes which offer single error correct and double error detect capabilities [6]. Given the low bit error rates of early flash, this was satisfactory to correct the sporadic bit error that arose. As bit error rates enlarged with each successive generation of both SLC and MLC flash, designers progressed to more complex cyclic codes, such as Reed-Solomon (R/S) or Bose-Chaudhur-Hocquenghem (BCH) algorithms to increase the correction capability [6]. While both of the algorithms are similar, R/S codes execute correction over multi-bit symbols while BCH makes correction over single-bit symbols.

Here is how it works for data storage: when any k-bit data is written to flash memory, an encoder circuit makes the parity bits, adds these parity bits to the k-bit data and creates an n-bit code-word [8]. Parity bits form a code that refers to the bit sequence in the word and is stored along with the unit of data. The routinely computed ECC, i.e., the whole code-word, is kept in the spare area of the page to which it relates. Throughout the reading operation, a decoder circuit examines errors in a code-word, and corrects the mistaken bits within its error capability, thereby recovering the code-word [8].

When the unit of data is demanded for reading, a code for the stored and about-to-be-read word is calculated using the algorithm. ECC´s are again calculated, and these values are compared to the ECC values held in the spare area. If the codes match, the data is free of errors. The outcome of this assessment yields an ECC "syndrome" that shows whether errors occurred, how many bits are in error, and, if the errors are recoverable, the bit position of incorrect bits. If the codes do not match, the missing or incorrect bits are determined through the code comparison and the bit or bits are corrected or supplied. The additional information represent redundancy added by the code is recycled by the receiver to recover the original data.

A typical ECC will correct a one-bit error in each 2048 bits (256 bytes) using 22 bits of ECC code, or a one-bit error in each 4096 bits (512 bytes) using 24 bits of ECC code. However, as raw BER increases, 2-bit error correction BCH code becomes a desired level of ECC. Next generation flash devices will move to smaller geometries and increased number of bits per cell, features that will increase the underlying bit error rate [6].

V. SUMMARY

Today, flash memory are one of the most popular, reliable, and flexible non-volatile devices to store data. NAND flash memory has become very popular for usage in various applications where a large amount of data has to be stored. This article discusses important aspects related to the storage reliability and the actual bit error rate.

A NAND Flash device is composed by the memory array, which is separated into several blocks. In general it performs three basic operations: program a page, erase a block, and read a page. There are many noise sources that exist in the NAND flash, which considerably shrink the storage reliability of a flash memory. The paper presents a preliminary study, which was conducted in connection with the preparation of an experiment for evaluating the reliability of a NAND flash memory. The purpose of this study was to summarize the theoretical background. The preliminary aim was to identify factors affecting the reliability for potential usage of the methodology of a planned experiment. However, after considering all aspects, it has been realized that this approach is not possible. Therefore, further research will involve life-cycle and reliability testing using the Weibull analysis method.

ACKNOWLEDGMENT

This research has been supported by the European ARTEMIS Industry Association by the project 7H12002 "Interactive Power Devices for Efficiency in Automotive with Increased Reliability and Safety", by the TA04010476 project TACR "Secure Systems for Electronic Services User Verification" and by the CZ.1.05/1.1.00/02.0068 project OP RDI "CEITEC - Central European Institute of Technology".

REFERENCES

[1] Pavan, P., Bez, R., Olivo, P., and Zanoni, “Flash Memory Cells—An Overview,” Proceedings of the IEEE, VOL. 85, NO. 8, 1997, pp. 1248-1271.

[2] Eureka Technology Inc. (2012). “NAND Flash FAQ”. Retrieved 4. 11. 2014, from http://www.actel.com/ipdocs/apn5_87a_FAQ.pdf.

[3] Gong, B. Y., “Testing Flash Memories”, 2004, from http://www.ictest8.com/base/yuanli/Testing%20Flash%20Memories.pdf.

[4] Paikin, A., “Flash memory”. Retrieved 4. 11. 2014, from http://www.hitequest.com/Kiss/Flash_terms.htm.

[5] Micheloni, R., Marelli, A., and Commodaro, S., “NAND overview: from memory to systems.” Springer Science and Business Media. doi:10.1007/978-90-481-9431-5_2, 2010.

[6] Deal, E., “Trends of NAND Flash Memory Error Correction.” Retrieved 7. 8. 2014 from http://www.cyclicdesign.com/index.php/ecc-trends-in-nand-flash, 2009.

[7] Heidecker, J., “NAND Flash Qualification Guideline”, NEPP Electronic Technology Workshop, 6. 11. 2012

[8] Wang, X., Dong, G., Pan, L., and Zhou, R., “Error Correction Codes and Signal Processing in Flash Memory”, InTech. Retrieved from http://www.intechopen.com/books/flash-memories/error-correction-codes-and-signalprocessing-in-flash-memory, 2011.

[9] Kuo, T.-W., Huang, P.-C., Chang, Y.-H., Ko, C.-L., and Hsueh, C.-W., “An Efficient Fault Detection Algorithm for NAND Flash Memory”. Retrieved from http://www.iis.sinica.edu.tw/~johnson/public_files/FaultDetection.pd.

[10] Toshiba Electronics Europe, “How to handle the increasing ECC requirements of the latest NAND Flash memories in your Industrial Design.” Retrieved 7. 2. 2013, from http://www.toshiba-components.com/memory/data/Whitepaper_BENAND_11_2012.pdf, 2012.

[11] Deal, E., “Hamming, RS, BCH, LDPC - The Alphabet Soup of NAND ECC.” Retrieved 7 2, 2013, from Cyclic Design: http://www.cyclicdesign.com/index.php/parity-bytes/3-nandflash/24-hamming-rs-bch-ldpc-the-alphabet-soup-of-nand-ecc, 2011.

[12] Shimpi, L. A., “Micron's ClearNAND: 25nm + ECC, Combats Increasing Error Rates.” Retrieved 7. 2. 2013 from AnandTech: http://www.anandtech.com/show/4043/micron-announces-clearnand-25nm-with-ecc, 2010.

[13] Naftali, S., “Signal processing and the evolution of NAND flash memory.” Retrieved 7. 2. 2013, from Anobit: http://embedded-computing.com/articles/signal-evolution-nand-flash-memory, 2010.

Multichannel Laboratory Equipment for Measurement of Smart Concrete Material Properties

Ladislav Machan, Pavel Steffan
Department of microelectronics
BUT, FEEC
Brno, Czech Republic
email: machan@feec.vutbr.cz, steffan@feec.vutbr.cz

***Abstract*— “Smart Concrete” materials are cement-matrix composites prepared according to the final application. Strain properties can be used to measure the deformation of concrete structures (bridges, beams, pillars) or for weighing-in-motion of road vehicles. This article describes novel laboratory equipment which is designed for multichannel dynamic and long-period material stability measurements.**

Keywords—smart concrete; dynamic measurement; long-period measurement

I. Introduction

One of the most common structural materials used in engineering construction is cement and its mixtures (concrete and mortar). Cement is slightly conducting material, but its electrical conductance, Electromagnetic Interference (EMI) shielding effectiveness and wave absorbing property are very poor. In order to increase the ability of cement materials to transfer electrons, additional conductive fillings and loadings have to be added. Smart Concrete (SC) could be considering to be a material of the future. Due to its attractive features, SC can be used as a strain-sensing element.

The strain-sensing properties are achieved by a proper volume amount of conductive filler. In this system, the matrix is made of cementitious material with small amount of silica fumes, fly ash, and fine aggregates [1]. Different conductive fillers were tested considering the best strain-sensitivity / material price ratio. Existing research proved carbon black and graphite particles to be the best choice in terms of price. The best strain-sensitivity is achieved near the percolation threshold of filler particles [2].

Strain properties of the composite can be evaluated by impedance changing. The impedance changing sensitivity regarding deformation can be widely affected by a proper choice of concrete admixtures [3]. Generally, in many types of mixtures, the real component is not much affected by the deformation, on the other hand, the imaginary component is, and it can be used to detect the changes [4]. A novel mixture with carbon black filler has a different behavior. The real part of impedance is strongly affected by deformation and can be used for measurements. This property is profitable with regard to the future usage. Measuring principle can be simplified and power supply requirements reduced. Conventional DC techniques for resistance measurements cannot be used. Electrode system of sensing element would be damaged by electrolytic corrosion in this case. Square-wave AC technique with an excitation frequency of 1 kHz and an excitation voltage of 1 V_{p-p} were experimentally set [5]. In the light of new knowledge about materials, a necessity to a simple, relatively inexpensive and portable device has been raised. Considering the needs of laboratory measurements, some requirements on a new device were established:

- AC square-wave measuring principle of resistance.
- Excitation frequency 1 kHz and voltage of 1 V_{p-p}.
- Eight independent measuring channels for multiple element sensing applications.
- Shunt sensing and bridge-based measuring option.
- Four additional channels for sensing temperature and humidity.
- Integrated memory storage device compatible with PC and File Allocation Table (FAT32) system for data logging.
- Battery powered.
- Suitable for a long-period measurements.
- Water and dust proof case.

In Section 2, there is a description of the device. The block diagram is described in Subsection A and B. Subsection C is focused on measuring principle. In Subsection D, there is a mechanical design discussed. The measuring automation and data processing is described in Subsection E. In Subsection F, there are discussed the results of development and final measurement parameters. In Section 3, there is a conclusion which summarizes the results of work.

II. Description of the device

According to the aforementioned requirements, a block diagram of the instrument has been suggested.

A. The block diagram of device

The diagram is shown in Figure 1. The device function is based on 16-bit MSP430F5529 microcontroller, which uses Reduced Instruction Set Computing (RISC) architecture [6]. This microcontroller can be “in circuit” programmed via Joint Test Action Group (JTAG) interface. Interface also allows real-time debugging of firmware, which ensures all device functions.

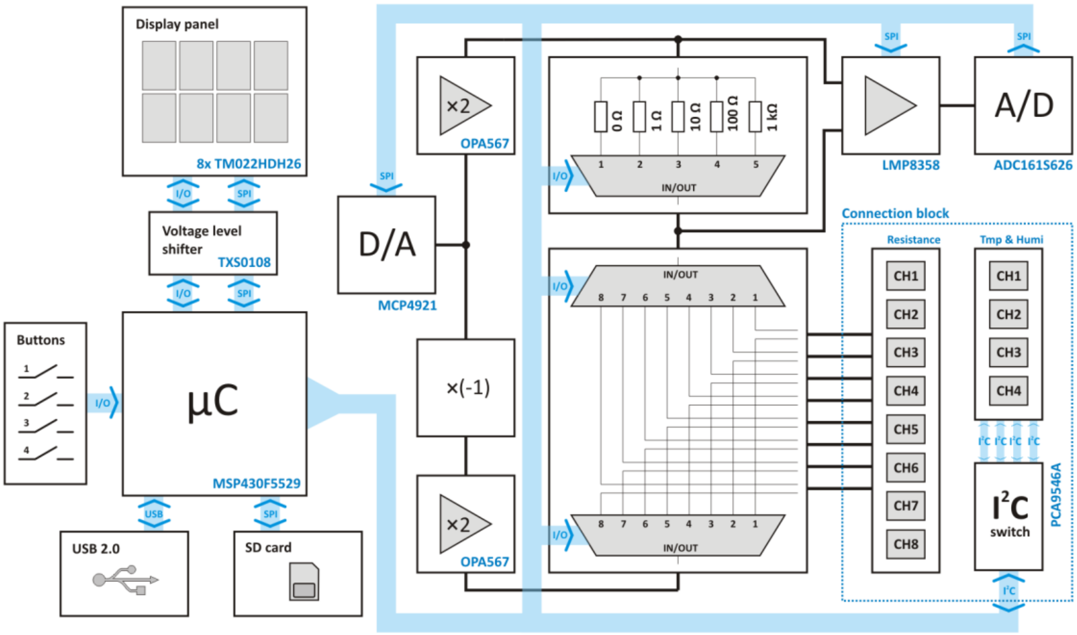


Figure 1. The principial block diagram of proposed device

The device is equipped with a Secure Digital (SD) card, which can be inserted into a side slot, for saving the measured data. SD card communicates with the microcontroller via Serial Peripheral Interface (SPI) interface. A special feature of used microcontroller is built in Universal Serial Bus (USB) interface connection for transferring the measured data into PC. Microcontroller contains complete physical layer of USB communication device. All higher layers of protocol are implemented in the firmware of device. There are two basic regimes of operation. When a measurement is running and measured data are periodically stored into memory, USB device works in Communications Device Class (CDC) mode (virtual COM port emulation in operating system). By this way, it is possible to view a response of measured system in real-time. Specialized software developed together with device is able to receive this data and display it in a graph. It is possible to view a response of measured system real-time by this way. The second mode of USB operation is used when a measurement is stopped and SD card is not used for storing data. This mode is called Mass Storage Class (MSC). SD card is transparently accessible for operational system of PC via USB device in this case. User can download measured data which are stored on SD card in standard FAT32 file system. This behavior is common e.g. for today‘s Smartphones and contributes to user comfort of the device. There is no need to open the housing of device and remove the SD card from the slot for downloading measured data to PC.

B. *Supply voltage supervising*

The aforementioned USB is not used for device supplying and internal accumulator charging. The main supply voltage is obtained from build-in lead-acid battery 6 V, 12 A/h. This large-capacity battery is able to supply the device a long time thereby enabling long-period measurements. USB bus supplying capability is 5 V / 500 mA. The usage of USB bus for charging this type of battery would not be effective. The battery charging is provided by external “fast-charger” through a build-in connector.

The supply voltage supervising is maintained by a group of supply blocks depicted in Figure 2. All supply voltage supervising blocks are based on linear regulation principle. No switch current blocks were used to minimize the noise and ripples in supply strings. This configuration is not such a power effective, but a quiet supply is a big benefit for this application. Supply system of the device consists of a digital supply string, high power analog supply string and low power analog supply string. Digital supply string contains 3.3 V and 5 V voltages. All digital blocks uses 3.3 V voltage levels (microcontroller, SD card, D/A converter). Higher voltage 5 V is used for signal relays, Liquid Crystal Display (LCD) backlight and voltage level shifter block as an option for possibility to connect an older type of display. High-power analog string is used to supply a pair of high-power operational amplifiers OPA567. Noise sensitive analog or mixed-mode parts are supplied via low-power analog string. There are supply and reference voltages for virtual ground definition.

Printed circuit board is designed considering low-noise layout rules. Analog and digital parts of device are separated into two boards. Noisy digital components are spatially separated from noise-sensitive analog blocks in this way. For even better noise properties, it is possible to cover analog board by shielding box. Special attention was paid to the design of ground surfaces. Each supply string has own ground surface shielding. All grounds are connected in one point to avoid noise induction via ground loops.

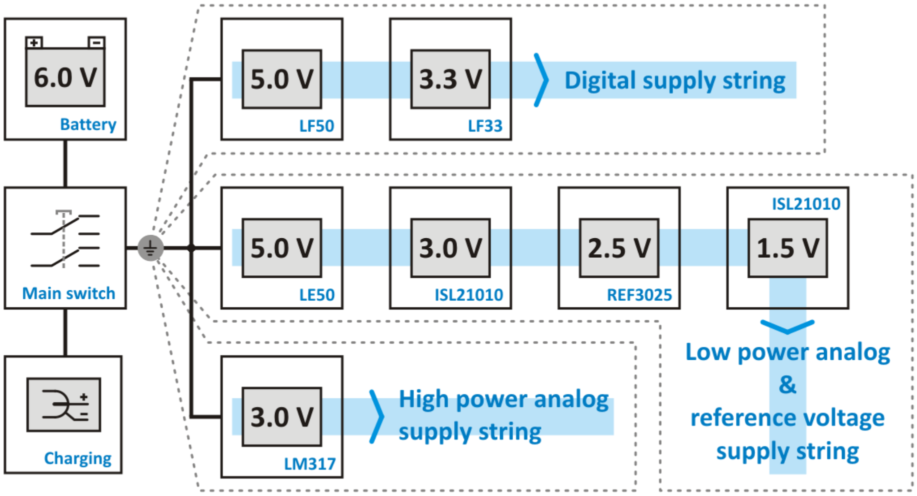


Figure 2. Supply voltage supervising blocks

C. *Measuring principle*

Excitation signal is digitally generated by D/A converter MCP4921. Output signal is amplified by the pair of high-power operational amplifiers OPA567 connected in push-pull configuration. One of them is connected as non-inverting amplifier while the other is inverting. Voltage gain of both amplifiers is 2. Supply voltage of amplifiers is 3 V and virtual ground is shifted to the middle of dynamic range by voltage reference 1.5 V. Output current from amplifiers is internally limited (short circuit protection) and a shift of virtual ground enables the flow in both directions via electrical load. Shunt resistors are in series with load and can be (1 from 5) selected by the program of microcontroller. Shunts are switched by small signal relays, considering low-noise design. Measured sample of material is connected as an electrical load via digitally controlled channel selection switch. Channels are switched (1 from 8) by small signal relays, too. There are two modes of operation which can be selected by proper combination of channel and shunt switches. The first measuring mode uses two-wire load connection. Current flowing through the load causes voltage drop on the shunt resistor. Voltage drop on the shunt resistor is amplified by precision instrumentation amplifier LMP8358 with digitally switchable gain (10, 20, 50, 100, 200, 500 and 1000). Output voltage from this amplifier is sampled by 16-bit successive approximation A/D converter ADC161S626. Samples are taken only in steady states of square-wave excitation signal. Resistance is calculated from a known value of selected shunt resistor, gain, excitation voltage and result from A/D converter. In this mode, eight channels can be used simultaneously one by one.

The second measuring mode offers the possibility of four-wire bridge connection of load. This type of connection helps to reduce temperature and humidity drift of measured material. In an ideal case, temperature and humidity changes act on all four elements of bridge in the same way and the effect on measurement vanishes. This mode does not use a shunt resistor (0 Ω shunt is selected). Proper setting of channel switch allows using four channels for bridge excitation and four channels for differential voltage sensing. Differential voltage is amplified by LMP8358 and processed by the same way as in the case of the first mode. Usage of the second “bridge mode” is under development at this time. A problem arises in the field of technological preparation of measured samples. There is a need to use four material elements with similar value of resistance, in bridge connection. In the meantime, production repeatability is not sufficient considering absolute value of resistance.

In the case of the first “shunt mode”, temperature and humidity cause a drift in measured value. There are four additional digital channels which allow connecting temperature and humidity sensors, in the block diagram (Figure 1). Sensors can be connected via I^{2}C serial interface. For example SHT21 from Sensirion Company is a suitable type. Addressing possibilities of serial bus are extended by hardware I^{2}C switch PCA9546A. Data from sensors can be used for numerical compensation of drift.

D. *Mechanical design*

Dynamic measurements of samples bring problems such as dusty environments, mechanical vibrations, EMI interferences and temperature stress. Mechanical design of a device is based on requirements for battery powered laboratory instrument which must be robust and environmentally resistant. The device is build-in IP68 standard plastic case with transparent cover (Figure 3). Under cover, there is a very well readable display panel. Four control buttons are placed from the front. The opposite side is equipped with eight connectors for measuring probes and four connectors for digital sensors. On the left side, there is the main switch and a connector for charging. The right side includes a water and dust proof USB connector with a rubber plug.

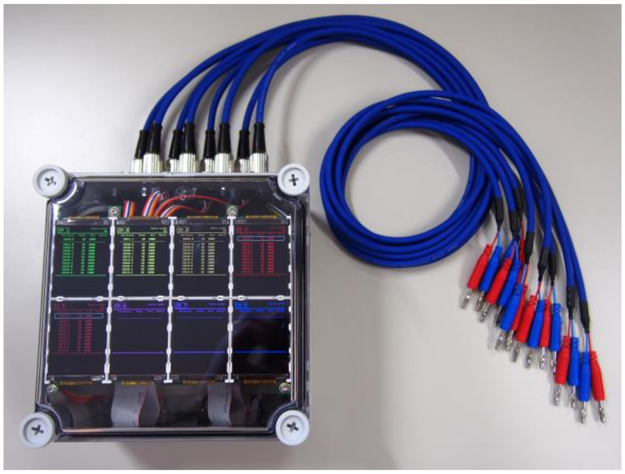


Figure 3. Mechanical design of device

E. *Measurement automation and data processing*

User can start to measure with predefined configuration by pressing the proper button. First, it is possible to choose a measurement mode (shunt or bridge), number of logged channels and measuring period. Measuring range and the gain of instrumentation amplifier is set automatically for each sample. Measuring can be paused and started again or stopped. When measurement is stopped a file with data log is saved with a specific name. During measurements, it is possible to add a marker to log by pressing a button. This feature is useful for better orientation in data log. Output of measurement is in standard ".csv" format. It can be easily processed for example in Microsoft Excel. An example of processed output data log from dynamical measurements is depicted in Figure 4.

F. *Results and final measurement parameters*

An example of processed output data log from dynamical measurements is depicted in Figure 4. This measurement was realized under conditions:

- constant temperature and humidity,
- concrete block size: 300 x 300 x 500 mm,
- automatic range and gain select,
- measurement duration: 2.5 hours,
- three channel mode,
- sample rate: 1s,
- cyclic press loading:
  - 3x 4.4 MPa till 16.6 MPa.
  - 1x 4.4 MPa till 33.3 MPa.
  - 3x 4.4 MPa till 16.6 MPa.
  - 1x 4.4 MPa till 22.2 MPa.
  - 3x 4.4 MPa till 16.6 MPa.
  - 3x 4.4 MPa till 27.8 MPa.
  - 3x 4.4 MPa till 16.6 MPa.
  - 1x 4.4 MPa till 44.4 MPa (destruction).

Sampled data were recalculated from absolute resistance values to relative changes of resistance during loading. Absolute values were compared with the output of professional impedance analyzer Agilent E4980A and absolute error of proposed device is 0.2%. This result is sufficient with regard to the current state of research. There is a possibility for the future improvement. On the basis of known internal temperature, the temperature drift of shunt resistor value can be numerical compensated.

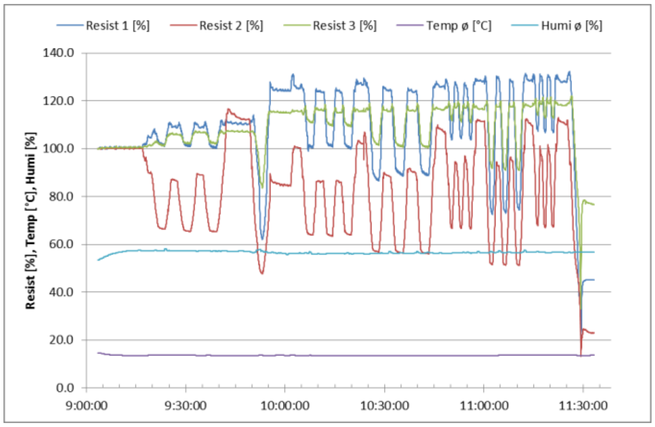


Figure 4. Processed data output – 3 channel dynamical measurement

III. Conclusion

A development of new laboratory instruments was presented in this paper. The overall structure, including suggested block diagram, realization and design of the printed circuit boards, was described. The precision LCR meter E4980, which does not enable the multichannel measurements, can be replaced by this low-cost device.

This instrument is currently used for laboratory measurements and characterization of smart concrete panels at department of microelectronics.

Acknowledgment

This research was supported by the Ministry of Industry and Trade of the Czech Republic under the MPO ČR č. FR-TI3/485 project.

References

[1] N. Xie, X. Shi, D. Feng, B. Kuang, and H. Li, ”Percolation backbone structure analysis in electrically conductive carbon fiber reinforced cement composites”, Composites Part B: Engineering, Volume 43, Issue 8, December 2012, Pages 3270-3275, ISSN 1359-8368.

[2] H. Li, H. Xiao, and J. Ou, "Effect of compressive strain on electrical resistivity of carbon black-filled cement-based composites", Cement and Concrete Composites, Volume 28, Issue 9, October 2006, Pages 824-828, ISSN 0958-9465.

[3] J. Junek, R. Cechmanek, P. Steffan, and P. Barath, "Vliv uhlikovych primesi na elektricke vlastnosti anorganickych kompozitu", XIIIth international conference Ecology and new building materials and products, 2009, Telc, 978-80-254-4447-4.

[4] D. D. L. Chung: Composite Materials - Second Edition, Springer, London, 2010, p. 349 ISBN 978-1-84882-830-8.

[5] P. Steffan, P. Barath, J. Stehlik, and R. Vrba,: "The Multifunction Conducting Materials Base on Cement Concrete with Carbon Fibers". Electronics, 2008, č. b4, p. 82-86. ISSN: 1313- 1842.

[6] Texas Instruments [online]. 2015 [retrieved: March, 2015]. MSP430F5529 datasheet. Available from: <http://www.ti.com/lit/ds/symlink/msp430f5529.pdf>.

The Modified Cement Composite Materials for Electromagnetic Shielding and Stress Detections

Steffan Pavel, Machan Ladislav, Vrba Radimir
Department of microelectronics
FEEC, Brno University of Technology
Brno, Czech Republic
e-mail: steffan@feec.vutbr.cz, machan@feec.vutbr.cz, vrbar@feec.vutbr.cz

Junek Jiri, Cechmanek Rene

Research Institute of Building Materials
Brno, Czech Republic
e-mail: junek@vustah.cz, cechmanek@vustah.cz

***Abstract* — This article deals with the measurement system and the use of special cement composites with carbon particulates. The type of carbon particulates consequently determines the electrical properties of cement composite material. These materials can be used for electric heating, electromagnetic shielding or stress measurement.**

Keywords-smart concrete; shielding efficiency; composite materials; carbon fibers; stress/strain detection.

I. Introduction

Nowadays, characteristics of composite materials based on cement include also other applications of cement materials which could be used, among others, for construction of self-monitoring buildings, bridges, etc. Besides, monitoring the usual properties (mechanical stability over the period of time, environmental resistance, design limits or economic profitability) is beginning to become more important and significant for construction of buildings. Requirements for construction of self-monitoring buildings are consisted of the interconnection of more different fields. In our particular case it is the question of connecting the construction and electrical engineering. At the end of the practical application self-monitoring buildings should facilitate the construction, installation or construction work.

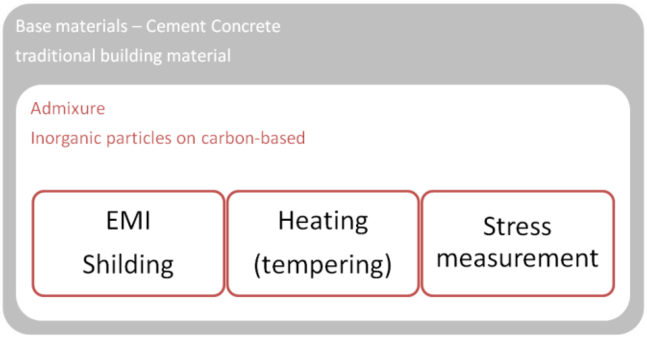

Base materials – Cement Concrete
traditional building material
Admixure
Inorganic particles on carbon-based
EMI
Shilding
Heating
(tempering)
Stress
measurement

Figure 1. Three areas of application of composite materials

During the current research, which is being done, it has been found out that the composite cement materials with carbon fibers allow for monitoring transients that are generated in the material for example due to pressure, tension, temperature and humidity. This is shown in Figure 1. Electrical properties of cement composites materials can be affected by a suitably selected type and amount of carbon particles.

The purpose is to provide complete solutions, including evaluation of the implemented electronic system that allows for obtaining and transmitting information to the user. The invented self-monitoring material can be used for the implementation of an individual measuring object [1][2]. In bridge structures, it is necessary to take into consideration a possibility of the function of the autonomous operation of the system. Major part of the research is focused on solving the problems of sensing impedance of the composite material which plays an important role in issues of aging and degradation of metallic materials due to strong alkaline environments. Corrosion of electrodes can be reduced by means of suitable corrosion inhibitors. For the corrosion diagnosis the accelerated model tests are used.

For the actual application we need to know the relation between accelerated model test and the real environment.

In Section 2, there is a description of measuring electromagnetic fields shielding. In Section 3, there is a description of alternative test method for the testing of shielding effectiveness of composite shielding materials. In Section 4, there is description of theoretical aspects of mechanism of shielding. The results of alternative measurement methods are in Section 5. Next Section 6 describes response of material to stress/strain load and the last Section 7 is conclusions.

II. EMC Measuring of Composite Materials

Measuring the effectiveness of electromagnetic fields shielding is usually done in an electromagnetic chamber. The measurements are used for transmitting and receiving antennas, test and signal generator. To receive the test signal, electromagnetic compatibility (EMC) receiver or

ISBN: 978-1-61208-399-5

spectrum analyzer are normally used. Measurements are generally usually carried out the way that the receiver, the receiving antenna and the necessary cables are placed inside the chamber or inside the test field. Transmitter (signal generator) and the broadcast antennas are located on the outer side of the tested object. The locations of the antennas depend on the EMC chamber or a box. The accuracy of the measurement depends on the correct location and position of the antennas [3][4].

The problem could appear when it is necessary to measure the shielding effectiveness of the material or the chamber or the box from which they are constructed. Especially in the development stage and in order to provide accurate measurements it is not possible to construct the whole chambers or boxes which are too big. This solution is expensive and also time consuming.

Similar problem appears when it is necessary to know the shielding efficiency of the construction materials such as bricks, plasterboard, concrete, etc. These materials could also be called, especially during their development stage, composite materials.

The main problems during the construction of chambers or boxes are caused by using the types of materials which are mentioned above. The chamber or box doors have usually the main influence on the whole shielding, in the other words, the doors always represent the weakest part of these chambers. But the construction of the doors, e.g., made from the concrete, is really complicated, almost impossible [7].

III. COAXIAL FLANGE

Materials for EMI shielding are different from those of magnetic shielding. EMI shielding is a rapidly growing application of carbon materials, especially of short carbon fibers. This review addresses measurement methods from carbon fibers materials usable for EMI shielding, including non-structural and structural composites, colloidal graphite, as well as EMI gasket materials.

The alternative test method for the testing of shielding effectiveness of shielding materials is often stated in literature [3]. The presented coaxial test apparatus is suitable for thin materials like plastic or metallic board, fabric material, etc. This setup is not suitable for the construction materials (concrete, bricks, etc.) because it is very complicated to produce the thin concrete board with the maximal height thickness of around 1 mm. The modified test setup was produced, after analyses of commonly available measurement solutions and setups. Our flange was mainly designed for frequency range from 9 kHz up to 1 GHz. The shape and dimensions of the flange were calculated for the 50 Ω input and output impedances [5].

The design of the flange was done according to the basic mathematical relations [5]

$$Z_M = \frac{60}{\sqrt{\varepsilon_r}} \ln \frac{a_2}{a_1}, \tag{1}$$

where

Z_M is the characteristic impedance of the measurement system (50 Ω);
ε_r is the relative permittivity (in this case is equal 1, air);
a_2, a_1 are the radius of the coaxial line (flange).

The transition from the N-type connector to the opposite end of the flange has the linear shape for both central and external parts. This shape was chosen for the better fabrication. The linear shape could be optimized for the better impedance which especially should be suitable for frequencies over the 1 GHz. The central flange conductor is fabricated from the brass. The rest of the flange is made from the aluminum alloy. The flange was tightened by the torque wrench after the inserting the test composite and it was always tightened by the same value of torque. This setup increases the accuracy of each measurement and also increases the repeatability during the several measurements. The detailed description of the measurement chamber is stated in literature [7].

The measured scattering parameters of the flange itself are given in Figure 2. The S11 and S22 are in the whole range of interest under the -15 dB which refers to the good matching of the both test ports with the measuring system. The insertion losses in both directions (S21 and S12) are in the measuring frequency range less than 1 dB. This data refers to the accurate design of the whole flange. The flange itself will have the insignificant influence on the total dynamic range of the whole measurement setup. The dynamic range will be mainly affected by the used measuring devices (generator and spectral analyzer).

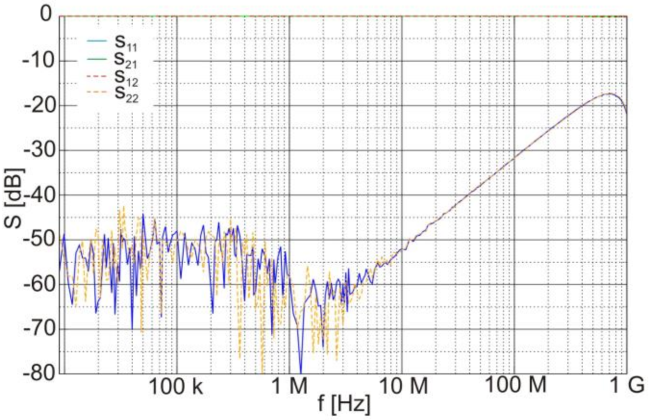


Figure 2. Measured scattering parameters of the realised coaxial flange

IV. Mechanism Of Shielding

The primary mechanism of EMI shielding is usually reflection. For reflection of the radiation by the shield, the shield must have mobile charge carriers (electrons or holes) which interact with the electromagnetic fields in the radiation. As a result, the shield tends to be electrically conducting, although a high conductivity is not required. For example, a volume resistivity of the order of 1 [Ωcm] is typically sufficient. However, electrical conductivity is not the scientific criterion for shielding, as conduction requires connectivity in the conduction path (percolation in case of a composite material containing a conductive filler), whereas shielding does not. Although shielding does not require connectivity, it is enhanced by connectivity. Metals are so far the most common materials for EMI shielding. They operate mainly by means of reflection due to the free electrons in them. Metal sheets are bulky, so metal coatings made by electroplating, electroless plating or vacuum deposition are commonly used for shielding. The coating may be on bulk materials, fibers or particles. Coatings tend to suffer from their poor wear or scratch resistance [6].

Absorption is usually a secondary mechanism of EMI shielding. For significant absorption of the radiation by the shield, the shield should have electric and/or magnetic dipoles which interact with the electromagnetic fields in the radiation. The electric dipoles may be provided by $BaTiO_3$ or other materials having a high value of dielectric constant. The magnetic dipoles may be provided by Fe_3O_4 or other materials having a high value of the magnetic permeability, which may be enhanced by reducing the number of magnetic domain walls through the use of a multilayer of magnetic films. The absorption loss is a function of the product $\sigma_r\mu_r$, whereas the reflection loss is a function of the ratio σ_r/μ_r, where σ_r is the electrical conductivity relative to copper and μ_r is the relative magnetic permeability. Silver, copper, gold and aluminum are excellent materials for reflection, due to their high conductivity. Superpermalloy and mumetal are excellent for absorption, due to their high magnetic permeability. The reflection loss decreases with increasing frequency, whereas the absorption loss increases with increasing frequency [6].

V. EMI Results

The measured scattering parameters refer to the accurate design of the coaxial flange. The next problem will appear with the prefabrication of the concrete ring as the test sample. This ring has to be produced with the high accuracy of its dimension. The example of measured shielding efficiency of the composite concrete material is depicted in the Figure 4. There is also shown the data which was measured with the brass disc. The shielding efficiency of the brass disc is the 115 dB at kHz range and around 70 dB at the GHz range. The shielding efficiency of the composite concrete material is only several dB in the range from 100 MHz up to 1 GHz. So low shielding efficiency of the concrete material is mainly caused by this small thickness of the material (only 8 mm). Figure 3 Measured scattering parameters of the realized coaxial flange.

Due to the skin effect, the composite material having conductive filler with a small unit size of the filler is more effective than one having conductive filler with a large unit size of the filler. For effective use of the entire cross-section of a filler unit for shielding, the unit size of the filler should be the same or smaller than the skin depth. Therefore, the filler of unit size 1 µm or less is typically preferred, though such a small unit size is not commonly available for most fillers and the dispersion of the filler is more difficult when the filler unit size decreases. Figure 4 shows the observed parameters for different types of carbon materials.

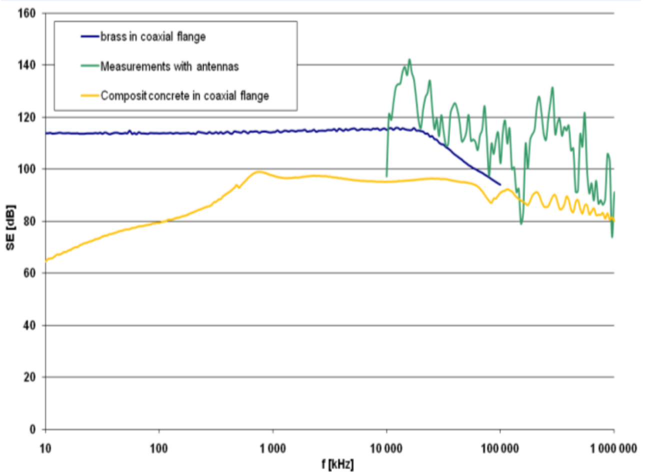


Figure 3. Shielding efficiency of the brass calibration test disc and the composite concrete test sample.

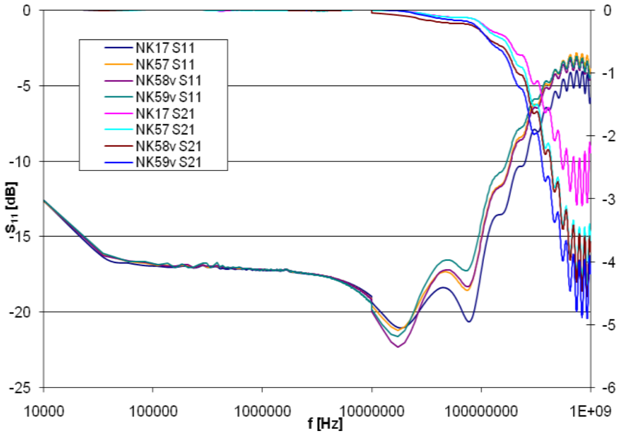


Figure 4. Shielding efficiency of the brass calibration test disc and the composite concrete test sample.

ISBN: 978-1-61208-399-5

VI. STRESS/ STRAIN MEASUREMENT

Another usage of smart concrete is used for sensing the load of concrete elements and structures mainly for measurements which use strain gauges. A strain gauge is a device used to measure strain on the surface part, by means of mechanical stress (tension, compression, etc). In fact, the strain gauge measures the relative deformation. Mechanical stress cannot be measured directly, and thus converted from the measured deformation. To calculate the necessary knowledge of the modulus of elasticity of material under consideration.

Composite material with carbon particles is sensitive to load changes [8]. This sensitivity is manifested by a rapid change of the measured impedance. Measurement was carried out in static mode. Gradually the load was adjusted in range from 0 kN to 1500 kN, after reaching a maximum, the sample was gradually relieved to the minimum load of 200 kN. The characteristic diagram of impedance response for smart concrete measured to stress load are given in Figure 5. Strain properties of the composite can be evaluated by changing impedance. The impedance changing sensitivity regarding the deformation can be widely affected by a proper choice of concrete admixtures [8]. Generally, in all types of carbon admixtures, the impedance of component is affected by the deformation and can be used to detect the changes. For impedance measuring we used an excitation frequency at 1 KHz, and an excitation voltage of 1 V (peak-peak) were experimentally set. Appropriately selected admixture has an effect on the relative size of the impedance changes depending on the pressure. The voltage level of the measuring voltage depends on used metal electrodes. For measuring impedance we used copper electrodes. These electrodes have a high resistance to corrosion in an alkaline environment..

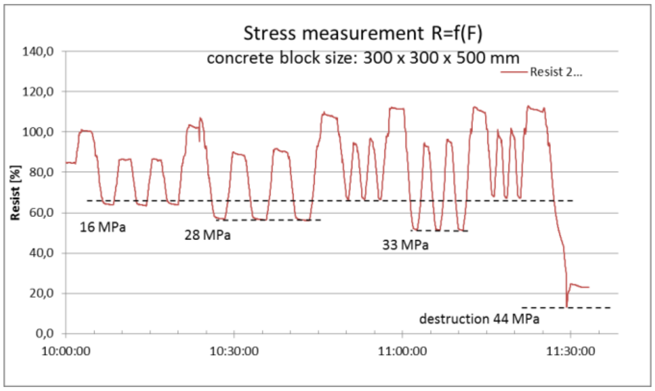


Figure 5. Concrete block - stress measurement

VII. CONCLUSIONS

Composite materials with carbon fibers were produce. Cement composite materials with admixture of carbon particulates, can be really used for electromagnetic shielding or stress sensing. The shielding efficiency of material is composed from several parts. The reflection loss, absorption loss and multiple path reflection losses are the main three parts of the whole electromagnetic shielding. For the accurate classification of the shielding efficiency of composite concrete material it will be necessary to measure each part of the whole electromagnetic shielding effectiveness. This measurement could be done by the vector network analyzer. The dependency of the thickness of the material and shielding efficiency could be determined in the harmony with measured data. The future work on EMI shielding will be focused on these problems and also on the compound of the composite concrete materials.

Some of these materials are stress sensitive and we were measured them under cyclic press loading in range from 0 kN to 1500 kN, after reaching a maximum, the sample was gradually relieved to the minimum load of 200 kN. The future work on stress measurement will be focus to increase the sensitivity of smart concrete materials.

ACKNOWLEDGMENT

This research was supported by the Ministry of Industry and Trade of the Czech Republic under the MPO ČR č. FR-TI3/485 project. The described research was performed in laboratories supported by the SIX project; the registration number CZ.1.05/2.1.00/03.0072, the operational program Research and Development for Innovation.

REFERENCES

[1] D.D.L. Chung, “Composite Materials“ - Second Edition, Springer, London, 2010, p. 349 ISBN 978-1-84882-830-8.

[2] D.D.L. Chung, “Functional Materials“ – Vol.2 Electrical, Dielectric, Electromagnetic, Optical and Magnetic Applications (With Companion Solution Manual), World Scientific Publisher, 2010, p. 345 ISBN 978-981-4287-15-9.

[3] IEEE Standard: Method for Measuring the Effectiveness of Electromagnetic Shielding Enclosures. EMC Society, New York 2006, p. 39.

[4] J. Svačina, “Electromagnetic Compatibility, Principles and Methods. “ (printed in Czech) Brno University of Technology 2001. “Join to European Union”.Volume 2, p. 156 ISBN 80-214-1873-7.

[5] D. Černohosky, Z. Novaček and Z. Raida, “Electromagnetic waves and lines. “ (published in Czech). Brno: Vutium 1999. p. 136 . ISBN 80-214-1261-5.

[6] D.D.L. Chung, “Electromagnetic interference shielding effectiveness of carbon materials.“ Elesevier Science Ltd.: Pergamon (2000), Carbon 279-285.

[7] J. Drinovsky, Z. Kejik, "Electromagnetic Shielding Efficinecy Measurement of Composite Materials" Measurement Science Review, 2009, vol. 9, No. 4, p. 109-112.

[8] R. Cechmánek, J. Junek, B. Nespor and P. Steffan, “Carbon-Based Composites Enable Monitoring of Internal States in Concrete Structures“, World Academy of Science, Engineering and Technology International Journal of Civil, Structural, Construction and Architectural Engineering Vol:8, No:10, 2014.

Software Integration of a Safety-critical ECU: an Experience Report

Ieroklis Symeonidis, Niklas Angebrand, Kostas Beretis, Eric Envall
ArcCore AB
Gothenburg, Sweden
e-mail: {ieroklis.symeonidis, niklas.angebrand, kostas.beretis, eric.envall}@arccore.com

Abstract—**In this paper, we present a software integration methodology in accordance to the automotive software standard AUTOSAR. The case under examination is the active safety electronic control unit (ECU) of the recently developed platform, called Scalable Product Architecture, of the Volvo Car Corporation. Particular emphasis is given in the relationship between the supplier of the ECU and the car manufacturer. Efficient communication between these two parties has been a challenging issue. Therefore, specific workflows regarding the exchange of information and the overall way of working are presented. The need of a dedicated integration team acting as an interface between the two organizations is also highlighted. Finally, concrete guidelines enabling continuous integration throughout the development process are provided. Our approach contributed in decreasing the software development cycles. We strongly believe that the conclusions drawn from our work experience can be generalized up to a certain level, affecting the automotive industry as a whole.**

Keywords- automotive software; AUTOSAR; embedded systems; integration; ISO26262.

I. INTRODUCTION

The development of embedded software in the automotive domain is characterized by its complexity. The reason behind that is not only the increasing functional and safety requirements but also the fact that a significant number of subcontractors are involved in the development process. Traditionally car manufacturers (or OEMs) had the role of integrating different subsystems developed by their suppliers (Tier-1s, Tier-2s, etc.) following a “black-box” approach [1]. These systems used to be limited in scope and usually resided on a single ECU. On the contrary modern automotive systems include distributed functions with strict timing and communication requirements between various ECUs. Such functions can be found in active safety and advanced driver assistance systems of premium cars.

The challenges of software engineering within the automotive industry have been well described by Broy et al in [2]. The increasing role of software as a source for innovation and the multidisciplinary nature of the domain are highlighted. Issues regarding the integration of software components in a distributed system are presented in [3]. In this paper, a general overview of the existing challenges as well as possible solutions to design and analysis issues in automotive systems are presented. AUTOSAR [4] and its implications on the development tool-chain are analyzed in [5]. In addition, [5] also shows how the concepts of the AUTOSAR methodology can be brought together in a common tool-chain leading to a higher degree of automation in the software development. Moreover a case study regarding a way of incorporating AUTOSAR in the development process of an antilock braking system (ABS) is presented in [6]. Finally, an approach for dealing with the complexity of the development process according to specific corporate needs is analyzed in [7]. Emphasis is being given on the need of constructing a tool chain with high degree of reusability and automation. A case study regarding a tool-chain model for AUTOSAR ECU design is also presented.

Software applications produced by different vendors need to be integrated into the final software for the ECU. The development of such software is an iterative process, consisting of multiple releases with parallel lifecycles. The purpose of our work is to present a well-defined software integration process and the way it should be aligned within the development process. These guidelines are derived from our experiences as a software integration partner for Volvo Car Corporation. Detailed information about roles and workflows as well as the flow of information between the OEM and the ECU supplier throughout the whole process will be provided. Our proposed workflow allows the exchange of key information between the two partners while at the same time it protects each side’s intellectual property.

The remainder of the paper is organized as follows. In the following section, we provide the technical background of our paper. Section 3 presents the software development process and section 4 describes our approach towards software integration. Finally, we draw our conclusions and outline future work.

II. TECHNICAL BACKGROUND

In this section, we are going to identify the key stakeholders that were involved in the development process. Also, we are going to present the automotive standards that influence the development process. In this project, there are three interacting parties:

- **OEM:** The main job of the OEM is to develop functions (e.g., Lane Keeping Aid, Collision Warning). A *software architect* defines the structure of the part of the software that will implement these functions. The implementation is assigned to *function developers,* who use Matlab/Simulink as a development tool. The developer of each function is responsible for his distinct part of the code/functionality. The *system testers* are responsible for testing and verifying that the system conforms to a certain behavior.

- **Tier-1:** The responsibilities of the Tier-1 include both the hardware and the final software (including the OEM's advanced functionality). In particular, the Tier-1 designs the ECU hardware, performs OS configuration and implements its own functions. These functions address communication with peripherals (e.g., sensors) and basic functionality such as diagnostics. The same stakeholders (*software architect, function developers* and *system testers*) can be identified within the Tier-1.
- **Software integration team:** ArcCore's software integration team resides inside the OEM. Our team acts as an interface between the two organizations enabling effective communication between all the stakeholders on the appropriate level of abstraction. Our approach increased the efficiency of the development leading to shorter time to delivery and reduced cost. A major challenge of our software integration team was also to establish a work flow and an automated tool-chain. This tool-chain consisted of requirements management tools, code generation tools, AUTOSAR authoring tools, compiler and linking tools as well as general purpose tools like data repositories and build servers. Under this work environment it is clear that the responsibilities of the software integration team can be quite broad, requiring various competencies. The range of activities covered could span from developing glue code or gateway functionality, up to specifying to a component supplier the system functionality to which the component must conform [8].

A. AUTOSAR

AUTomotive Open System Architecture is a software architecture standard developed jointly by automotive manufacturers, OEMs and tool developers. This standard was created to satisfy the need for standardization of basic software and the interfaces to applications/bus systems [9]. The motivation behind that was to reduce system complexity and keep the development cost feasible. Some additional goals of AUTOSAR include the scalability across different vehicles and platforms, maintainability throughout the product lifecycle and the sustainable utilization of natural resources [10].

AUTOSAR follows a layered architecture, where hardware, basic software, runtime environment and application software are separated from each other [11]. The basic concepts of AUTOSAR are the Software Component (SWC), the Runtime Environment (RTE) and the Basic Software (BSW). Each SWC should be assigned to one ECU and encapsulates part of the functionality of the application [12]. The implementation of the SWC is independent of the underlying platform, following the basic design concept of separation between layers. The RTE provides a communication abstraction to the SWCs connected to it, providing the same interface and services both for inter and intra ECU communication. Since the requirements of SWCs running on RTE may vary, different ECUs may have different RTEs. The BSW is essential to run the functional part of the software. It is the standardized software layer, which provides services to the SWCs [11]. It contains both standard and ECU specific components.

Although this model based approach is a step forward in reducing the complexity of the development process, there are certain limitations. AUTOSAR methodology does not include topics like requirements management, hardware development and build management. Therefore, it does not cover the complete development process lifecycle [5]. Furthermore AUTOSAR does not standardize test procedures [5]. Finally, AUTOSAR neither defines concrete guidelines and procedures for development strategies to be followed nor separates distinctly the activities in the various development phases [13]. From the above it becomes clear, that there is not a universal way of working with the standard but only case specific implementations like the one addressed in this paper. The way of working can be subjective and highly dependent on the developer's work experience and interpretation of the standard. Therefore, a well-defined development process is of great importance.

B. ISO26262

ISO26262 [14] is the standard for functional safety management of electrical and or electronic systems within the automotive industry. It applies to all development activities of safety-related systems (electrical, electronic and software) and addresses possible hazards caused by malfunctioning behavior of such systems, including their interaction. It consists of 10 parts, each one dealing with a specific development activity. Parts 6- "Product development at the software level" and 9-"Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses" are of great importance for our work. Part 6 highlights the importance of performing safety analysis at the software level and suggests some mechanisms for error handling and detection at a generic level. However, no clear guidelines are provided leading to subjective implementation in industrial practice [15]. Part 9 provides a classification mechanism for hazards according to ASIL. The ASILs can have the following values "QM, A, B, C, D" where D requires the most attention and QM the least due to a combination of potential severity, controllability and exposure of hazards [16].

ISO26262 provides guidance to identify the level of effort required to achieve the desired level of functional safety [17]. It can also be viewed as a defense against liability claims and it is not a certification requirement [17]. Furthermore AUTOSAR only provides mechanisms to support functional safety on a software level and does not guarantee any functional safety properties of the final system [16]. The key notion that brings together the two standards is "freedom of interference". By partitioning the system into safety related and non-safety related components it has to be assured that there is no interference between the safety related ones and the rest of the software, or that it is reliably detected. Memory partitioning provides spatial freedom of interference, while other techniques like implementation of a watchdog manager provides temporal freedom of interference. Finally, a way of guaranteeing correct exchange of information is through end-to-end communication protection mechanisms. However, IS026262 does not explicitly address AUTOSAR. Therefore, the selection and implementation of any safety mechanism is a responsibility of the AUTOSAR vendor.

III. Software Development Process

The case under examination is an active-safety ECU with two parties involved in the development of the software. Since both parties need to protect their intellectual property, only the compiled version of the source code is exchanged between them (i.e., object-code). Along with the object code a definition of the software structure and its interfaces is supplied in the form of ARXML files. This highlights the importance of a dedicated software integration team, able to combine object code from both sides with a common system extract into unified software. The system extract contains the software structure and the interfaces as well as all service and integration information needed by the software integration team, as defined by the AUTOSAR standard.

An automotive software development project consists of several internal iterations/releases. In our case, each release was divided into the following phases: contract, function integration, testing and verification and short-loop phases (Figure 1).

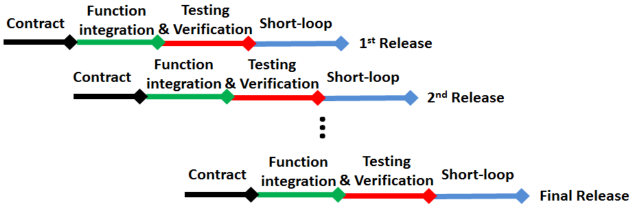


Figure 1. Software development process.

A. *Contract*

AUTOSAR defines a Composition SoftWare Component (C-SWC) that contains one or several Atomic SoftWare Components (A-SWCs), which we will refer to simply as SWCs. The system has a root composition, which contains one composition for the OEM SWCs and one for the Tier1 SWCs (Figure 2). Both sides need to define the interfaces between their compositions and to the external signal busses available for the ECU. This is done by exchanging contracts in form of a preliminary system extract (Figure 3). At this phase software architects on both sides need to provide an initial software structure for the SWCs containing interfaces for sending and receiving signals as well as interfaces towards the diagnostic services. In the contract phase not all details about the final SWC need to be defined. It is possible to add more information in an iterative manner. Usually in the automotive domain, external bus interfaces need to be defined early in the design process and remain unchanged (frozen) until the next release of the software. At a later stage the interfaces towards services like diagnostics need to be also frozen. In order for several parties to be able to work in parallel it is important to freeze the composition interfaces and the service interfaces at the same time.

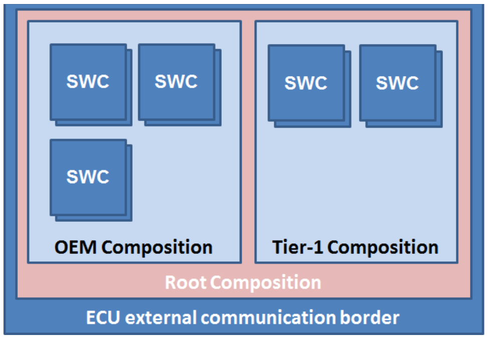


Figure 2. Software composition.

In order to validate the initial software structure, an RTE generation is performed by each party. RTE contracts are generated for each SWC during the contract phase. At this point the SWCs consist of a basic structure with no functionality, we call them SWC shells. To be able to make an RTE generation a preliminary BSW configuration is needed. The purpose is to validate and identify incompatibility issues in the initial structure. Depending on the completeness of the BSW configuration there might be errors/warnings at this phase. The cause of these errors/warnings must be identified by the software integration team and reported to the Tier-1. As an additional validation step the software containing only SWC shells is compiled. This helps to identify errors related to the source code.

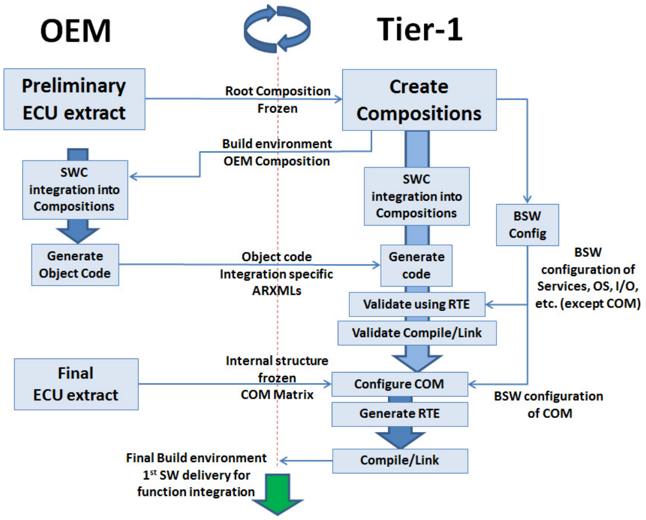

OEM
Tier-1
Preliminary ECU extract
Root Composition Frozen
Create Compositions
Build environment OEM Composition
SWC integration into Compositions
SWC integration into Compositions
BSW Config
Generate Object Code
Object code Integration specific ARXMLs
Generate code
BSW configuration of Services, OS, I/O, etc. (except COM)
Validate using RTE
Validate Compile/Link
Final ECU extract
Internal structure frozen COM Matrix
Configure COM
BSW configuration of COM
Generate RTE
Final Build environment 1st SW delivery for function integration
Compile/Link

Figure 3. Contract phase.

After the successful generation of the RTE, the SWC shells are delivered to the function development team. In order to enable continuous integration we produce a dummy

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-399-5

function for each SWC. This marks the beginning of the function integration phase (Figure 4).

B. *Function Integration*

At this stage the function developers introduce their functionality in the SWC shells. Any integration issue that might occur is resolved by the software integration team and a new shell is generated. To be able to deliver object code, the integration team needs a properly configured build environment, which is the responsibility of the Tier-1. More specifically RTE generation needs to be error-free, as well as BSW modules like OS and COM need to be configured properly. The key for continuous integration is that the code always builds. This is guaranteed by the initial dummy functions, which enable the software integration team to successfully build software regardless of the development state of a specific function. Functions are integrated gradually. Successful integration of a function is indicated by a successful build of the software. Once all the functions of the specific release are integrated, the produced software is delivered for testing on target. It is of great importance to verify that both sides use the same build environment to produce code. Therefore, the Tier-1 delivers the build environment at the end of this phase, having incorporated all the possible changes introduced during function integration. An example of such a change could be the mismatch of the linker script due to changes introduced in the memory sections.

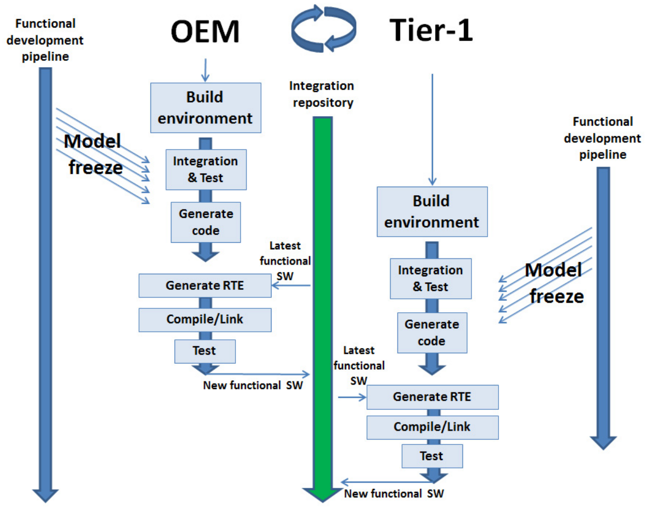


Figure 4. Function integration phase.

C. *Testing and Verification*

The testing and verification phase follows. This is not in the main scope of the software integration team and therefore it will not be analyzed in full detail. Dedicated teams on both sides perform testing and verification on system level, based on specific requirements. Prior to the system level tests it has to be mentioned that the function developers test their functions in simulated environments. The software integration team performs unit tests of the SWC shells and various configuration tests on the system extract. Also upon the official delivery in the form of a binary (from the Tier-1 to the OEM), a series of acceptance tests are performed. If the outcome is successful and the proper documentation is approved, then the software is available for the test vehicles.

D. *Short-loop*

Due to the relative long lead time from function freeze until the function is available in test vehicles there is a great need for having internal engineering releases (i.e., short-loops). This allows the OEM to speed up the function development process and detect possible bugs at an early stage. A short-loop can be performed once a build environment is setup, including the Tier-1 object code. In a short-loop build, new source code from the function team is introduced in order to build complete new software. As long as the internal structure changes and the border of the compositions remains the same software with the new functionality is produced. Any changes introduced must be compatible with the given BSW configuration.

IV. Integration Approach in the Context of AUTOSAR

The abstraction of AUTOSAR can, with great benefits, also be extended into the function development domain. This is done by supporting the function development with SWCs that encapsulate the pure functionality into a functional library and adding AUTOSAR helper components that take care of the AUTOSAR properties (Figure 5). Depending on the implemented functionality, each SWC may require different helper components.

This workflow comes with multiple gains. The functions can be developed and verified in a different environment (for example Matlab/Simulink) without any AUTOSAR dependencies. As mentioned earlier, the SWCs can always be provided with a dummy function, which ensures that the system always builds. Another aspect is that the function developers do not need to know the AUTOSAR details and can keep their focus on function development.

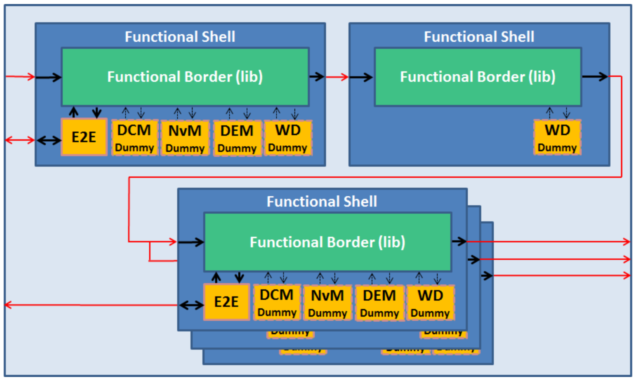


Figure 5. Functional composition under AUTOSAR context.

The usage of a dedicated database for the needs of the OEM's software architect was also introduced. All the system design related information (e.g., signal interfaces, diagnostic services) can be stored in this database. This

ISBN: 978-1-61208-399-5

enables the software architect to model the system in a lightweight fashion, thus providing a higher abstraction level regarding AUTOSAR. Using the information stored in this database, the software integration team can generate the AUTOSAR definition of the system in the form of ARXML files.

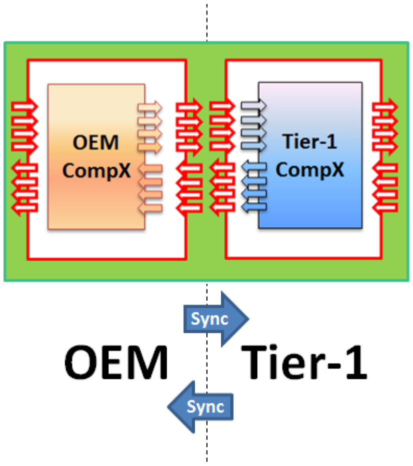


Figure 6. Exchange of information.

For the successful integration, the following rules were established for the exchange of information between the two sides. This is essential in order for both sides to have a synchronized view of the overall software structure (Figure 6). The OEM defines and owns the borders of both compositions (red boxes). Each side defines its composition and internal SWC structure including intra connections. The Tier-1 must make sure that its composition matches the defined border. This information combined with the signal database (defined by the OEM) leads to complete system extract that can be used for the development process. The supplier's border can only be changed by mutual agreement (change request).

V. Conclusions and Future Work

Throughout this case study we illustrated our approach for reducing the complexity involved in the development process of an automotive embedded system. The main challenges in such a process are the interaction between the development partners, the variety and sometimes incompatibility of the tools involved, as well as the subjective implementation of the dominant automotive standards such as AUTOSAR and ISO26262. We presented a proven-in-practice software development framework according to the needs of the AUTOSAR standard. The interaction between the OEM and the Tier-1 becomes much more efficient and at the same time intellectual property is protected. The key element for the successful interaction is the software integration team, which has a broad variety of responsibilities as described earlier. This team may be part of the OEM or could alternatively be a third partner working for the OEM like in our case.

Furthermore concrete guidelines enabling continuous integration in the context of AUTOSAR were provided. In this way, the function development gets decoupled from any AUTOSAR constraints. This leads to shorter development cycles and consequently to a faster time-to-market for the final vehicle. According to the “Driver Support and Software Integration” manager of the Volvo Cars Corporation, the time for producing a short-loop has decreased “from several days to about an hour”. He also stated that, “the AUTOSAR interface specification time has decreased from three months to less than two hours”. Previously this process was manual, involving several engineers, while now it is fully automated.

However, there is still room for improvement. Certain adaptations of the existing tool-chain are needed, in order to deal with incompatibilities between different tools. Ideally this tool-chain should fit into any automotive development environment. Finally, we also plan to implement an AUTOSAR-compliant testing framework for function performance measurement and debugging on target, based on actual log data from test vehicles. In this way possible bugs related to actual implementation that were not detected through simulations can be recreated on a development board with the same microprocessor. With this approach we reduce the need to utilize test vehicles, which are limited in number and might not be available.

Acknowledgments

We would like to express our appreciation to all people at the Active Safety department of the Volvo Car Corporation and especially to Daniel Levin, for their collaboration throughout this project. Furthermore we are also sincerely grateful to Postdoc researcher Risat Pathan from Chalmers University of Technology for his insightful advice.

References

[1] H. Heinecke, et al., "Software Components for Reliable Automotive Systems," in Design, Automation and Test in Europe, 2008. DATE '08, 2008, pp. 549-554.

[2] M. Broy, I. H. Kruger, A. Pretschner, and C. Salzmann, "Engineering Automotive Software," Proceedings of the IEEE, vol. 95 , 2007, pp. 356-373.

[3] M. Di Natale and A. L. Sangiovanni-Vincentelli, "Moving From Federated to Integrated Architectures in Automotive: The Role of Standards, Methods and Tools," Proceedings of the IEEE, vol. 98, 2010, pp. 603-620.

[4] AUTOSAR development partnership, "AUTomotive Open System ARchitecture". [retrieved: March, 2015]. Available: http://www.autosar.org

[5] S. Voget, "AUTOSAR and the automotive tool chain," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010, 2010, pp. 259-262.

[6] T. Hermans, P. Ramaekers, J. Denil, P. D. Meulenaere, and J. Anthonis, "Incorporation of AUTOSAR in an Embedded Systems Development Process: A Case Study," in Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on, 2011, pp. 247-250.

[7] M. Biehl, J. El-khoury, and M. Törngren, "Automated Tailoring of Application Lifecycle Management Systems to Existing Development Processes," International Journal on Advances in Software, vol. 6, 2013, pp.104-116.

[8] P. Wallin, J. Froberg, and J. Axelsson, "Making Decisions in Integration of Automotive Software and Electronics: A Method Based on ATAM and AHP," in Software Engineering for Automotive Systems, 2007. ICSE Workshops SEAS '07. Fourth International Workshop on, 2007, pp. 5-12.

[9] S. Furst, "Challenges in the design of automotive software," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010, 2010, pp. 256-258.

[10] AUTOSAR development partnership, "AUTOSAR-Technical Overview". [retrieved: March, 2015] . Available: http://www.autosar.org/about/technical-overview/

[11] D. Diekhoff, "AUTOSAR basic software for complex control units," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010, 2010, pp. 263-266.

[12] B. Huang, H. Dong, D. Wang, and G. Zhao, "Basic Concepts on AUTOSAR Development," in Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on, 2010, pp. 871-873.

[13] C. JungEun, L. DongSun, and L. Chaedeok, "Process-Based Approach for Developing Automotive Embeded Software Supporting Tool," in Software Engineering Advances, 2009. ICSEA '09. Fourth International Conference on, 2009, pp. 353-358.

[14] International Organization for Standardization, "ISO 26262-10:2012," ed, 2012. [retrieved: March, 2015]. Available: https://www.iso.org/obp/ui/#iso:std:iso:26262:-10:ed-1:v1:en

[15] V. Bonfiglio, L. Montecchi, F. Rossi, and A. Bondavalli, "On the Need of a Methodological Approach for the Assessment of Software Architectures within ISO26262," presented at the SAFECOMP 2013 - Workshop CARS (2nd Workshop on Critical Automotive applications : Robustness \& Safety) of the 32nd International Conference on Computer Safety, Reliability and Security, Toulouse, France, 2013, pp. 51-56.

[16] K. Hyungju, R. Itabashi-Campbell, and K. McLaughlin, "ISO26262 application to electric steering development with a focus on Hazard Analysis," in Systems Conference (SysCon), 2013 IEEE International, 2013, pp. 655-661.

[17] B. Böddeker and R. Zalman, "AUTOSAR at the Cutting Edge of Automotive Technology," in 7th International Conference on High-Performance and Embedded Architectures and Compilers, 2012. [retrieved: March, 2015]. Available: http://www.hipeac.net/system/files/zalman-keynote.pdf

Natural Language Processing of Textual Requirements

Andres Arellano
Government of Chile,
Santiago, Chile
Email: andres.arellano@gmail.com

Edward Carney
Lockheed Martin,
College Park, MD 20742, USA
Email: edward.carney@lmco.com

Mark A. Austin
Department of Civil Engineering,
University of Maryland,
College Park, MD 20742, USA
Email: austin@isr.umd.edu

***Abstract*—Natural language processing (NLP) is the application of automated parsing and machine learning techniques to analyze standard text. Applications of NLP to requirements engineering include extraction of ontologies from a requirements specification, and use of NLP to verify the consistency and/or completion of a requirements specification. This work-in-progress paper describes a new approach to the interpretation, organization and management of textual requirements through the use of application-specific ontologies and natural language processing. We also design and exercise a prototype software tool that implements the new framework on a simplified model of an aircraft.**

Keywords-Systems Engineering; Ontologies; Natural Language Processing; Requirements; Rule Checking.

I. Introduction

Model-based systems engineering development is an approach to systems-level development in which the focus and primary artifacts of development are models, as opposed to documents. As engineering systems become increasingly complex the need for automation arises [1]. A key element of required capability is an ability to identify and manage requirements during the early phases of the system design process, where errors are cheapest and easiest to correct. While engineers are looking for semi-formal and formal models to work with, the reality remains that many large-scale projects begin with hundreds – sometimes thousands – of pages of textual requirements, which may be inadequate because they are incomplete, under specified, or perhaps ambiguous. State-of-the art practice involves the manual translation of text into a semi-formal format (suitable for representation in a requirements database). A second key problem is one of completeness. For projects defined by hundreds/thousands of textual requirements, how do we know a system description is complete and consistent? The motivating tenet of our research is that supporting tools that make use of computer processing could significantly help software engineers to validate the completeness of system requirements. Given a set of textual descriptions of system requirements, we could analyze them making use of natural language processing tools, extracting the objects or properties that are referenced within the requirements. Then, we could match these properties against a defined ontology model corresponding to the domain of this particular requirement. This would throw alerts in case of lacking requirements for some properties.

II. Project Objectives

Significant work has been done to apply natural language processing (NLP) to the domain of requirements engineering [2] [3] [4]. Applications range from using NLP to extract ontologies from a requirements specification, to using NLP to verify the consistency and/or completion of a requirements specification. This work-in-progress paper outlines a framework for using NLP to assist in the requirements decomposition process. Our research objectives are to use modern language processing tools to scan and tag a set of requirements, and offer support to systems engineers in their task of defining and maintaining a comprehensive, valid and accurate body of requirements. Section III describes two aspects of our work in progress: (1) Working with NLTK, and (2) Integration of NLP with ontologies and textual requirements. A simple aircraft application is presented in Section IV. Section V covers the conclusions and directions for future work.

III. Work in Progress

Topic 1. Working with NLTK. The Natural Language Toolkit (NLTK) is a mature open source platform for building Python programs to work with human language data [5].

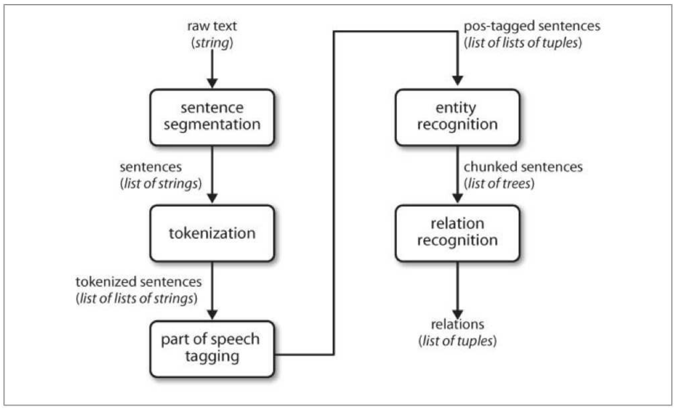


Figure 1. Information extraction system pipeline architecture.

Figure 1 shows the five-step processing pipeline. NLTK provides the basic pieces to accomplish those steps, each one with different options and degrees of freedom. Starting with an unstructured body of words (i.e., raw text), we want to obtain sentences (the first step of abstraction on top of simple

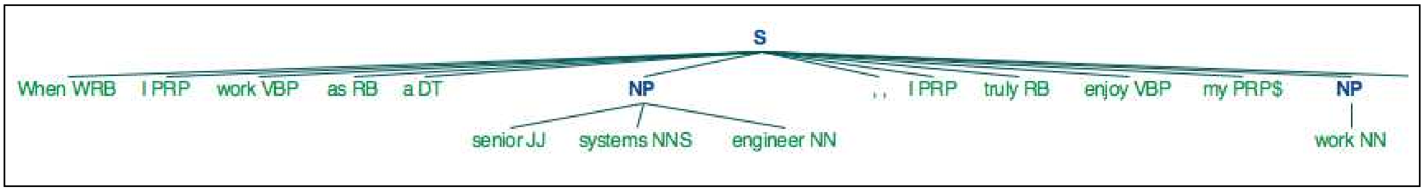


Figure 2. Output from building a chunking grammar.

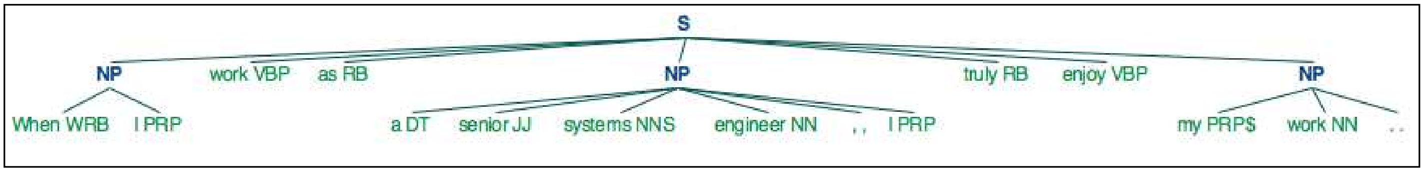


Figure 3. Output from the example on chinking.

words) and have access to each word independently (without loosing its context or relative positioning to its sentence). This process is known as *tokenization* and it is complicated by the possibility of a single word being associated with multiple token types. Consider, for example, the sentence: “These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule.” The abbreviated script of Python code is as follows:

```
text = "These prerequisites are known as (computer)
        system requirements and are often used as a
        guideline as opposed to an absolute rule."
tokens = nltk.word_tokenize(my_string)
print tokens
=>
[’These’, ’prerequisites’, ’are’, ’known’, ’as’,
 ’(’, ’computer’, ’)’, ’system’, ’requirements’,
 ’and’, ’are’, ’often’, ’used’, ’as’, ’a’,
 ’guideline’, ’as’, ’opposed’, ’to’, ’an’,
 ’absolute’, ’rule’, ’.’]
```

The result of this script is an array that contains all the text’s tokens, each token being a word or a punctuation character. After we have obtained an array with each token (i.e., word) from the original text, we may want to normalize these tokens. This means: (1) Converting all letters to lower case, (2) Making all plural words singular ones, (3) Removing *ing* endings from verbs, (4) Making all verbs be in present tense, and (5) Other similar actions to remove meaningless differences between words. In NLP jargon, the latter is known as *stemming*, in reference to a process that strips off affixes and leaves you with a stem [6]. NLTK provides us with higher level *stemmers* that incorporate complex rules to deal with the difficult problem of stemming. The Porter stemmer that uses the algorithm presented in [7], the Lancaster stemmer, based on [8], or the built in lemmatizer – Stemming is also known as *lemmatization*, referencing the search of the *lemma* of which one is looking an inflected form [6] – found in WordNet. Wordnet is an open lexical database of English maintained by Princeton University [9]. The latter is considerably slower than all the other ones, since it has to look for the potential stem into its database for each token.

The next step is to identify what role each word plays on the sentence: a noun, a verb, an adjective, a pronoun, preposition, conjunction, numeral, article and interjection [10]. This process is known as *part of speech tagging*, or simply *POS tagging* [11]. On top of POS tagging we can identify the *entities*. We can think of these *entities* as “multiple word nouns” or objects that are present in the text. NLTK provides an interface for tagging each token in a sentence with supplementary information such as its part of speech. Several taggers are included, but an *off-the-shelf* one is available, based on the Penn Treebank tagset [12]. The following listing shows how simple is to perform a basic part of speech tagging.

```
my_string = "When I work as a senior systems
             engineer, I truly enjoy my work."
tokens = nltk.word_tokenize(my_string)
print tokens

tagged_tokens = nltk.pos_tag(tokens)
print tagged_tokens
=>
[(’When’, ’WRB’), (’I’, ’PRP’), (’work’, ’VBP’),
 (’as’, ’RB’), (’a’, ’DT’), (’senior’, ’JJ’),
 (’systems’, ’NNS’), (’engineer’, ’NN’), (’,’, ’,’),
 (’I’, ’PRP’), (’truly’, ’RB’), (’enjoy’, ’VBP’),
 (’my’, ’PRP$’), (’work’, ’NN’), (’.’, ’.’)]
```

The first thing to notice from the output is that the tags are two or three letter codes. Each one represent a lexical category or part of speech. For instance, WRB stands for *Wh-adverb*, including *how*, *where*, *why*, etc. PRP stands for *Personal pronoun*; *RB* for *Adverb*; *JJ* for *Adjective*, *VBP* for *Present verb tense*, and so forth [13]. These categories are more detailed than presented in [10], but they can all be traced back to those ten major categories. It is important to note the the possibility of one-to-many relationships between a word and the tags that are possible. For our test example, the word *work* is first classified as a verb, and then at the end of the sentence, is classified as a noun, as expected. Moreover, we found two nouns (i.e. objects), so we can affirm that the text is saying something about *systems*, *an engineer* and *a work*. But we know more than that. We are not only referring to *an engineer*, but to a *systems engineer*, and not only a *systems engineer*, but a *senior systems engineer*. This is our *entity* and we need to *recognize* it from the text (thus the section

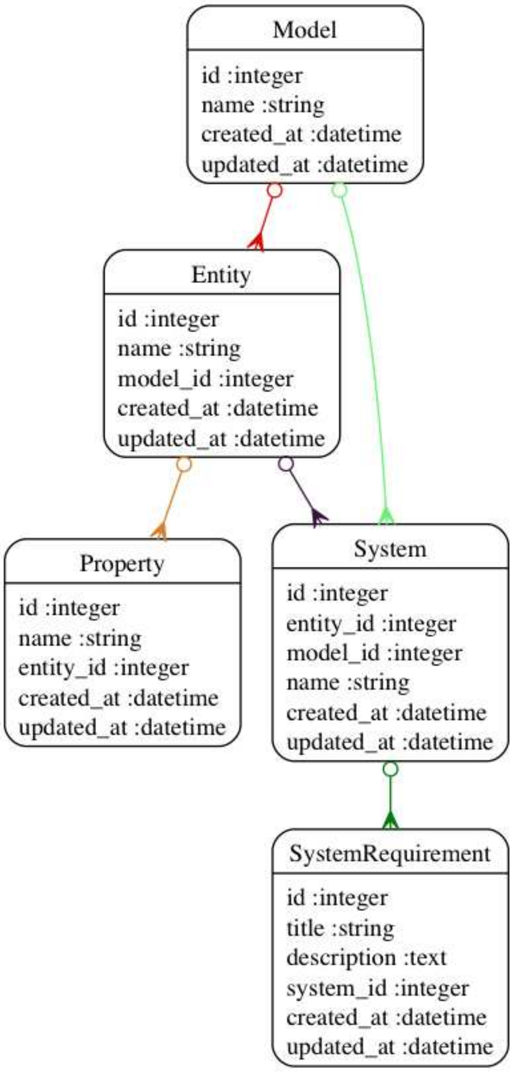


Figure 4. UML diagram of the application models.

name). In order to do this, we need to somehow tag groups of words that represent an entity (e.g., sets of nouns that appear in succession: *('systems', 'NNS'), ('engineer', 'NN')*). NLTK offers regular expression processing support for identifying groups of tokens, specifically noun phrases, in the text. The rules for the parser are specified defining *grammars*, including patterns, known as *chunking*, or excluding patterns, known as *chinking*. As a case in point, Figures 2 and 3 show the tree structures that are generated when chunking and chinking are applied to our test sentence.

Topic 2. Integration of NLP with Ontologies and Textual Requirements. In order to provide a platform for the integration of natural language processing, ontologies and systems requirements, and to give form to our project, we built *TextReq Validation*, a web based software that serves as a proof of concept for our objectives. The software stores ontology models in a relational database (i.e., tables), as well as a system with its requirements. It can do a basic analysis on these requirements and match them against the model's properties, showing which ones are covered and which ones are not. The software has two main components: The web application that provides the user interfaces, handles the business logic, and manages the storage of models and systems. This component was built using Ruby on Rails (RoR), a framework to create web applications following the Model View Controller pattern [14]. The views and layouts are supported by the front-end framework Bootstrap [15]; These scripts are written using Python. Figure 4 is a UML diagram showing all the models. The *models* corresponding to the MVC architecture of the web application, reveal the simple design used to represent an Ontology and a System. The first one consists of a Model – named after an Ontology Model, and not because it is a MVC model – that has many Entities. The Entities, in turn, have many Properties. The latter is even simpler, consisting of only a *System* that has many *System Requirements*. Most of the business logic resides in the models. Notice, in particular, system-level interpretation of results from the natural language processing.

IV. Simple Aircraft Application

We have exercised our ideas in a prototype application, step-by-step development of a simplified aircraft ontology model and a couple of associated textual requirements. The software system requires two inputs: (1) An ontology model that defines what we are designing, and (2) A system defined by its requirements. We manage a flattened (i.e., tabular) version of a simplified aircraft ontology. Figure 5 shows the aircraft model we are going to use.

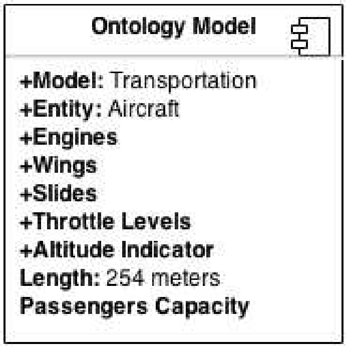
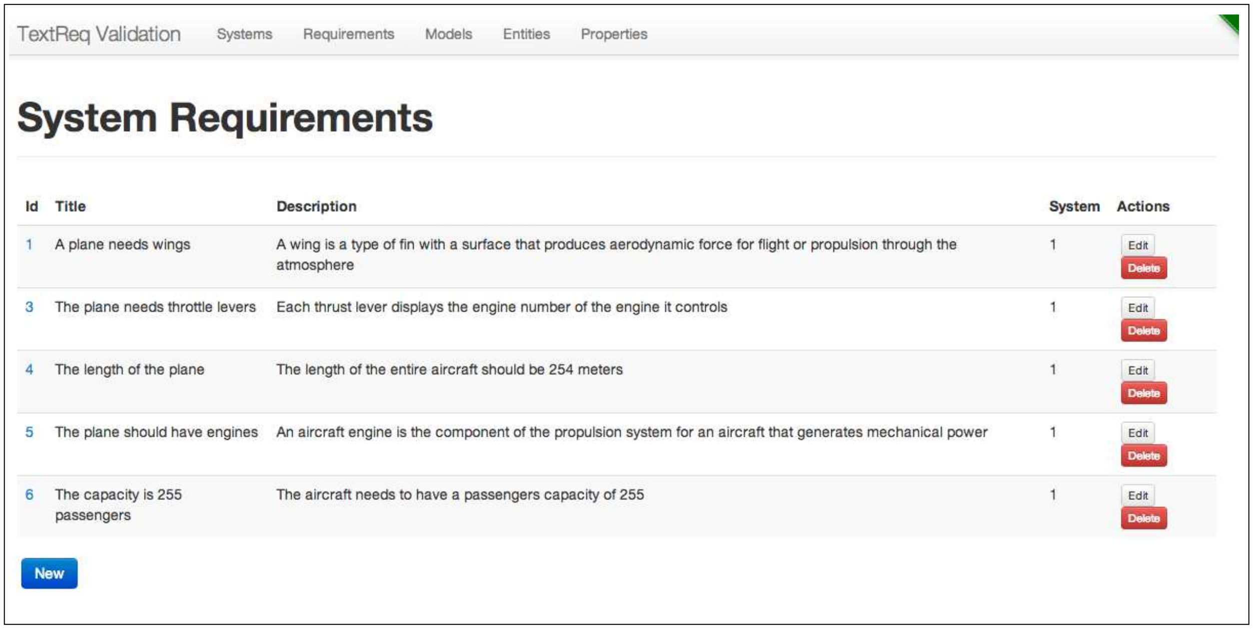


Figure 5. Simplified ontology model for an aircraft.

This simple ontology suggests usage of a hierarchical model structure, with aircraft properties also being represented by their own specialized ontology models. Second, it makes sense to include a property in the model even if its value isn't set. Naturally, this lacks valuable information, but it does give us the knowledge that that particular property is part of the model, so we can check for its presence. The next step is to create a system model and link it to the ontology. We propose a one-to-one association relationship between the system and an ontology, with more complex relationships handled through hierarchical structures in ontologies. This assumption simplifies development because when we are creating a system we only need to refer to one ontology model and one entity. The design of the system is specified through *textual system requirements*. To enter them we need a system, a title and a description. Figure 6 shows, for example, all the system Requirements for the system *UMDBus 787*. Notice that each



System Requirements

Id	Title	Description	System	Actions
1	A plane needs wings	A wing is a type of fin with a surface that produces aerodynamic force for flight or propulsion through the atmosphere	1	Edit Delete
3	The plane needs throttle levers	Each thrust lever displays the engine number of the engine it controls	1	Edit Delete
4	The length of the plane	The length of the entire aircraft should be 254 meters	1	Edit Delete
5	The plane should have engines	An aircraft engine is the component of the propulsion system for an aircraft that generates mechanical power	1	Edit Delete
6	The capacity is 255 passengers	The aircraft needs to have a passengers capacity of 255	1	Edit Delete

Figure 6. Panel showing all the requirements for the system *UMDBus 787*.

requirement has a title and a description, and it belongs to a specific system. The prototype software has views (details not provided here) to highlight connectivity relationships between the requirements, system model (in this case, a simplified model of a UMDBus 787), and various aircraft ontolology models. The analysis and validation actions match the system's properties taken from its ontology model against information provided in the requirements. The output from these actions is shown in Figures 7 and 8, respectively.

V. Conclusions and Future Work

When a system is prescribed by a large number of (non formal) textual requirements, the combination of previously defined ontology models and natural language processing techniques can play an important role in validating and verifying a system design. Future work will include formal analysis on the attributes of each property coupled with use of NLP to extract ontology information from a set of requirements. Rigorous automatic domain ontology extraction requires a deep understanding of input text, and so it is fair to say that these techniques are still relatively immature. A second opportunity is the use of NLP techniques in conjunction with a repository of acceptable "template sentence structures" for writing requirements [16]. Finally, there is a strong need for techniques that use the different levels of detail in the requirements specification, and bring ontology models from different domains to validate that the requirements belongs to the supposed domain. This challenge belongs to the NLP area of *classification*.

References

[1] M.A. Austin, and J.S. Baras, "An Introduction to Information-Centric Systems Engineering". Toulouse, France: Tutorial F06, INCOSE, June 2004.

[2] V. Ambriola and V. Gervasi, "Processing Natural Language Requirements," in Proceedings 12th IEEE International Conference Automated Software Engineering. IEEE Comput. Soc, 1997, pp. 36–45. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=632822

[3] C. Rolland and C. Proix, "A Natural Language Approach for Requirements Engineering," in Advanced Information Systems Engineering. Springer, 1992, pp. 257–277.

[4] K. Ryan, "The Role of Natural Language in Requirements Engineering," in [1993] Proceedings of the IEEE International Symposium on Requirements Engineering. IEEE Comput. Soc. Press, 1993, pp. 240–242. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=324852

[5] NLTK Project, "Natural Language Toolkit NLTK 3.0 documentation." [Online]. Available: http://www.nltk.org/

[6] C. Manning and H. Schuetze, Foundations of Statistical Natural Language Processing. The MIT Press, 2012. [Online]. Available: http://www.amazon.com/Foundations-Statistical-Natural-Language-Processing-ebook/dp/B007L7LUKO

[7] M. Porter, "An Algorithm for Suffix Stripping," Program: electronic library and information systems, vol. 14, no. 3, Dec. 1980, pp. 130–137. [Online]. Available: http://www.emeraldinsight.com/journals.htm?issn=0033-0337&volume=14&issue=3&articleid=1670983&show=html

[8] C. D. Paice, "Another Stemmer," ACM SIGIR Forum, vol. 24, no. 3, Nov. 1990, pp. 56–61. [Online]. Available: http://dl.acm.org/citation.cfm?id=101306.101310

[9] Princeton University, "About WordNet - WordNet - About WordNet." [Online]. Available: http://wordnet.princeton.edu/

[10] M. Haspelmath, "Word Classes and Parts of Speech," 2001. [Online]. Available: http://philpapers.org/rec/HASWCA

[11] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, Inc., 2009.

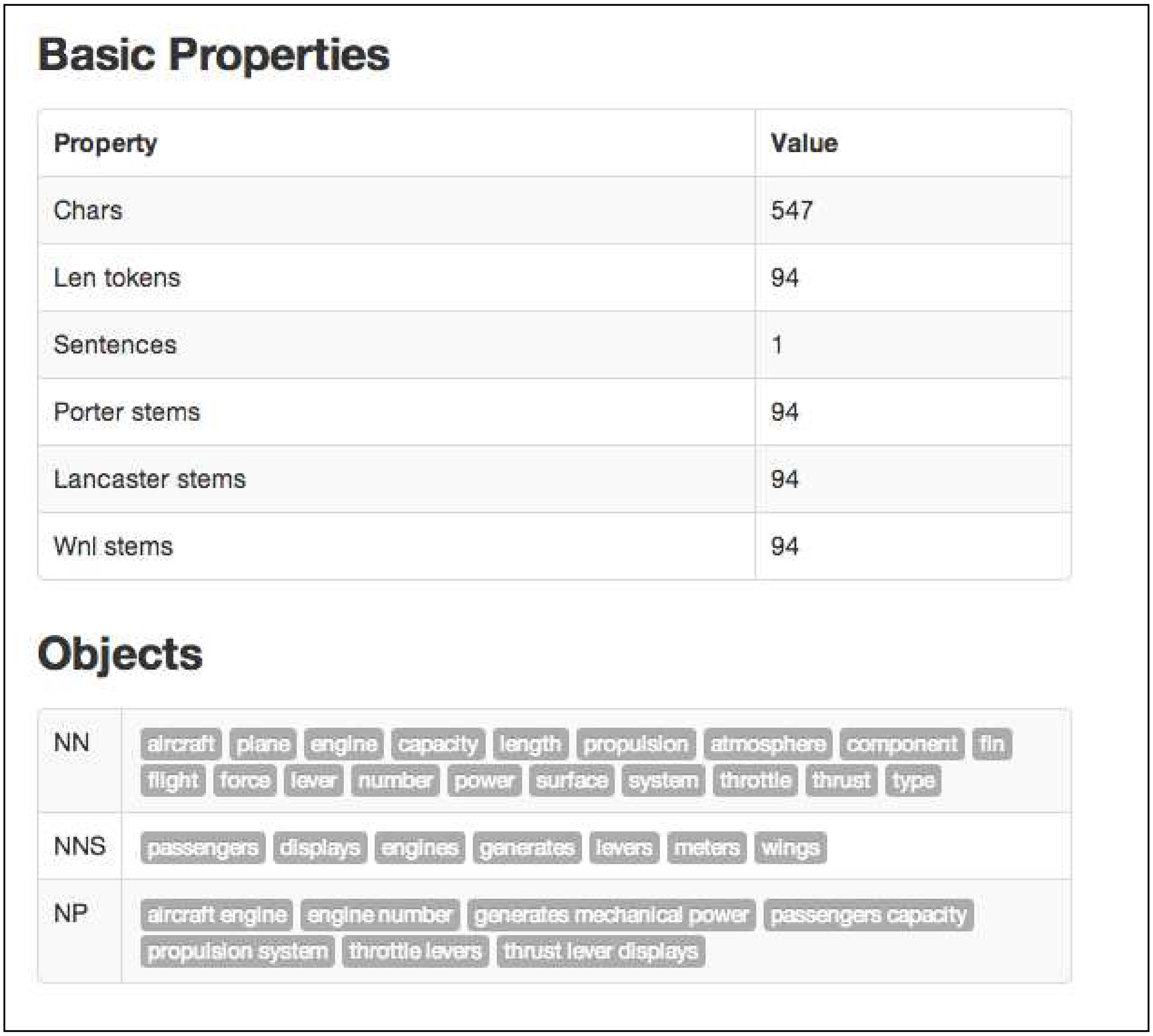
[12] University of Pennsylvania, "Penn Treebank Project." [Online]. Available: http://www.cis.upenn.edu/ treebank/

[13] B. Santorini, "Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision)," 1990. [Online]. Available: http://repository.upenn.edu/cis_reports/570

[14] Ruby on Rails. See http://rubyonrails.org/ (Accessed, March 2015).

[15] Bootstrap. See http://getbootstrap.com/2.3.2/ (Accessed, March 2015).

[16] E. Hull, K. Jackson, and J. Dick, Requirements Engineering. Springer, 2002. [Online]. Available: http://www.amazon.com/Requirements-Engineering-Elizabeth-Hull-ebook/dp/B000PY41OW



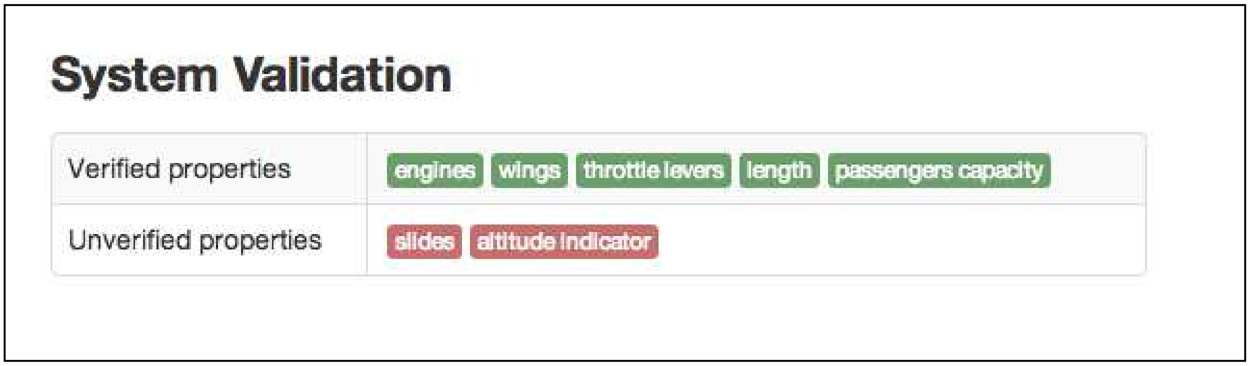
Basic Properties

Property	Value
Chars	547
Len tokens	94
Sentences	1
Porter stems	94
Lancaster stems	94
Wnl stems	94

Objects

NN	aircraft plane engine capacity length propulsion atmosphere component fin flight force lever number power surface system throttle thrust type
NNS	passengers displays engines generates levers meters wings
NP	aircraft engine engine number generates mechanical power passengers capacity propulsion system throttle levers thrust lever displays

Figure 7. Basic stats from the text, and a list of the entities recognized in it.



System Validation

Verified properties	engines wings throttle levers length passengers capacity
Unverified properties	slides altitude indicator

Figure 8. This is the final output from the application workflow. It shows what properties are verified (i.e., are present in the system requirements) and which ones are not.

A Practical Approach to Software Continuous Delivery

Everton Gomede and Rodolfo M. Barros
Department of Computer Science
State University of Londrina
Londrina, Paraná, Brazil
e-mail: evertongomede@gmail.com, rodolfo@uel.br

***Abstract*—To deliver quality software continuously is a challenge for many organizations. It is due to factors such as configuration management, source code control, peer-review, delivery planning, audits, compliance, continuous integration, testing, deployments, dependency management, databases migration, creation and management of testing and production environments, among others. To overcome these challenges, this work-in-progress paper presents a continuous delivery process that promotes artefacts produced by developers, in a managed fashion, to production environment, allowing bidirectional traceability between requirements and executables. As a preliminary result, we obtained an ecosystem of tools and techniques evaluated, tested and put into production in order to support this process.**

Keywords–Continuous Delivery; Process Quality.

I. Introduction

Software Delivery Process (SDP) consists of several tasks in order to promote artifacts created in the production environment (server(s) where an executable is installed to delivery features to the users) [1]. These ones can occur in either environment, producer or consumer. Due to the unique characteristics of each software product, a general process to various contexts probably cannot be set. Therefore, we should interpret a SDP as a *framework* to be customized according to the requirements and characteristics of each product (Software Delivery Process, in this context, is a part of Software Development Process).

This customization usually causes a *manual* execution of SDP [2]. Production environment is configured in a manual way by the infrastructure team using terminals and/or third-party tools. Artifacts are copied from a continuous integration server to a production environment and possibly some data and/or metadata are adjusted before software is released for operation.

However, this process has some weaknesses. Predictability is the first one, because it increases risk and downtime whether any unexpected situation occurs [3]. Additionally, the repeatability factor may compromise the diagnosis of post-deployment problems [2]. Finally, this process is not auditable and it does not allow the recovery of information about all events that were held to deliver a version.

There is a growing interest in practices to overcome these problems [4]. Such practices are known as *Software Continuous Delivery* (SCD), defined as the ability to publish software whenever necessary. This publication may be weekly, daily or every change sent to the code repository. The frequency is not important, but the ability to deliver when it is necessary [2].

This approach has great importance in software development because it helps who is in charge of delivering to understand better their process and, consequently, improve it. Such improvements can be in terms of automation, decrease delivery time, rework reduction, risk reduction, or others. Among them, the main is the ability to have a version of software, ready for delivery, each new code added to the repository.

In this context, this paper presents a practical approach to address the problems of software continuous delivery. The main objective is to contribute with a setup of servers, process, techniques and tools that assist to deliver software continuously. In addition, some recommendations and further work are discussed. Architecture issues, project management and other dimensions related to software development were omitted. We collect these data through of a case study.

Thus, this article was divided into five sections, including this introduction. In Section II, we present fundamental concepts and related works. In Section III, we present an approach to Software Continuous Delivery. In Section IV, we present preliminary results. Finally, in Section V, we present conclusions, recommendations and suggestions for future work.

II. Fundamental Concepts and Related Works

There is a relation between quality of software products and quality of the process used to build them. Implementation of a process aims to reduce rework, delivery time and increase product quality, productivity, traceability, predictability and accuracy of estimates [2]. In general, a software development process contains activities shown in Figure 1.

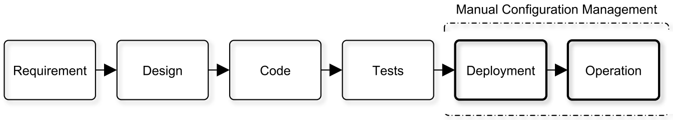


Figure 1. A simplified software development process [1] [2].

Configuration management tasks of deployment and operation activities, highlighted in Figure 1, are usually performed manually [2]. This practice, according to Humble and Farley [2], is accompanied by anti-patterns:

- Deploying software manually: there should be only two tasks to perform manually; (1) choose a version and (2) choose the environment. These are goals to be achieved in process like in [5].
- Deploying after development (requirement, design, code and tests) was complete: it is necessary to integrate all activities of the development process and

put stakeholders working together since the beginning of the project.

- Manual configuration management of production environments: All aspects of configured environments should be applied from a version control in an automated process.

In this context, some Software Continuous Delivery Practices arises. It is a developing discipline, which builds up software that can be released into production at any time, by minimizing lead-time [3].

To assist this type of software delivery approach, from construction to operation, Humble and Farley presents the *Deployment Pipeline* (DP), a standard to automate the process of SCD. Despite each organization may have an implementation of this standard, in general terms it consists of activities shown in Figure 2.

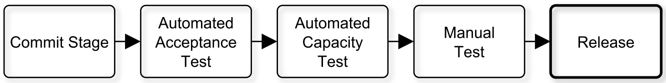


Figure 2. The deployment pipeline [2].

With each change, artifacts are promoted to next instance of pipeline through a series of automated tasks. The first step of the pipeline is to create executables and installers from the code repository, in a process known as Continuous Integration (CI). Other activities perform a series of tests to ensure that the executable can be published. If the release candidate passes all tests and criteria, then it can be published [2].

To implement this pipeline, some approaches were presented. Among them, Krusche and Alperowitz [5] described the implementation of a SCD process for multiple project. Their goal was to obtain the ability to publish software to their clients with just a few clicks. The main contribution of this work was to show that developers who have worked on projects with SCD, understood and applied concepts being convinced from the benefits of it.

Bellomo et al. [6] presented an architectural framework together with tactics to projects that address SCD. The main contribution of this work is a collection of SCD tactics in order to get software products performing with a higher level of reliability and monitoring into production environment.

Fitzgerald and Stol [4] published trends and challenges related to what the authors called ”Continuous *”, which is, all topics related to software delivery that can be classified as continuous. The authors addressed issues such as; Continuous Integration (CI), Continuous Publication (PC), Continuous Testing (CT), Continuous Compliance (CC), Continuous Security (SC), Continuous Delivery (EC), among others. An important point of this paper is the distinction between the Continuous Delivery and Continuous Publication. According to the authors, Continuous Publication is ability to put into production software products in an automated manner. This definition is complementary to the software continuous delivery definition given above.

Although all these works have a practical nature, none of them showed which tools were used, which recommendations for similar scenarios and which were the techniques used during deployment. Therefore, the work presented in this paper seeks to fill these gaps.

III. A Practical Approach

A. Main Proposal

To provide an infrastructure that allows the Software Continuous Delivery is the main goal of setup shown in Figure 3. It has 4 areas: Commit Stage (CS), Quality Assurance (QA), Staging (ST) and Production (PD).

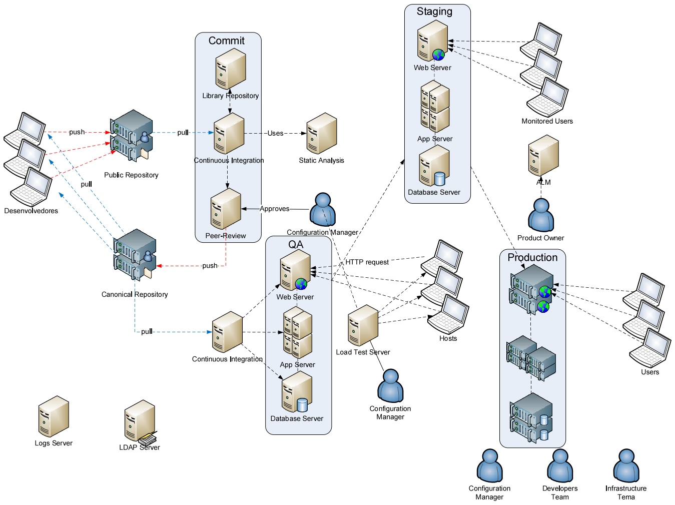


Figure 3. An overview of a setup of servers and areas.

B. Areas

Commit Stage (CS) has primary responsibility to implement continuous integration of all code reviews sent to the repository. This area consists of the following services:

- Public Code Repository
  - ○ Purpose: to get code reviews that have not been approved.
  - ○ Tool: Git (git-scm.com).
  - ○ Technique: it has a single branch, called master, which receives revisions of all developers.
- Continuous Integration
  - ○ Purpose: to integrate all code reviews sent to the server.
  - ○ Tool: Jenkins (jenkins-ci.org) and Maven (maven.apache.org)
  - ○ Technique: it does integration performing unit testing and adding first acceptance step in peer-review server.
- Static Analysis
  - ○ Purpose: to make code analysis generating quality reports.
  - ○ Tool: SonarQube (sonarqube.org).
  - ○ Technique: each integration performs a series of tests, such as size metrics, complexity, test coverage, dependency calculation, among others. Creates a baseline quality of the project.
- Peer-Review
  - ○ Purpose: to enable promotion/rejection of codes from public to canonical repository.

    - Tool: Gerrit (code.google.com/p/gerrit).
    - Technique: approval of two steps, the first being carried out by continuous integration server and the second by the configuration manager. If the review through both sides, code is promoted to canonical repository.
- Canonical Repository
    - Purpose: to receive approved code reviews.
    - Tool: Git (git-scm.com).
    - Technique: it has a single branch, called master, which receives revisions of peer-review server.
- Repository Libraries.
    - Purpose: to store libraries and components used in integration.
    - Tool: Nexus (sonatype.org/nexus).
    - Technique: libraries and components are installed automatically or manually on the server being available for use at the time of integration.

Layout and operation of Commit Area are shown in Figure 4.

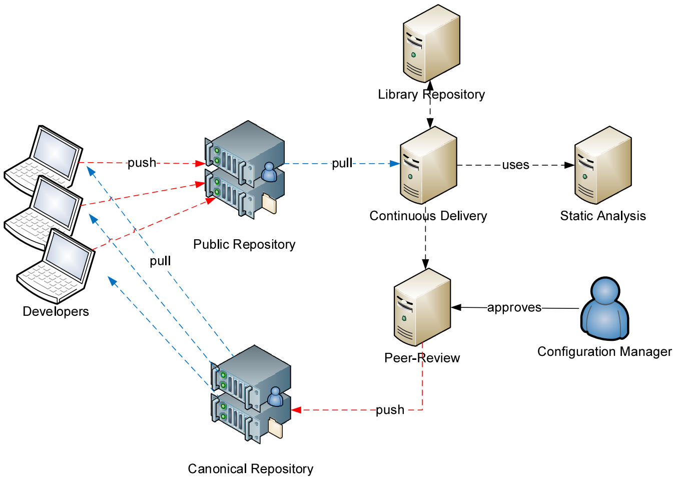


Figure 4. Commit Stage (CS).

Quality Assurance Area (QA) has the main purpose of performing all automated tests and allow Quality Manager perform manual tests, such as exploratory testing [2]. This area consists of the following services:

- Continuous Integration
    - Purpose: to obtain a copy of the code and perform integration, functional and automated load tests.
    - Tool: Jenkins and Maven (maven.apache.org).
    - Technique: get a copy of canonical repository to generate executable, install them into library server, application servers and database server. After that, execute integration, functional and load tests.
- Page Servers, Application and Database
    - Purpose: to host application to test
    - Tools: may vary according to the technology used; Wildfly and MSSQL are some examples.
    - Technique: can vary depending on the technology used (to install and configure, basically).
- Load Test
    - Purpose: to perform a load test against the page servers, application and database.
    - Tool: Jmeter (jmeter.apache.org) and Vagrant (vagrantup.com).
    - Technique: it performs script created by quality manager allocating hosts as required to test. It generates a supported load from baseline.

Operation of Quality Assurance area is shown in Figure 5.

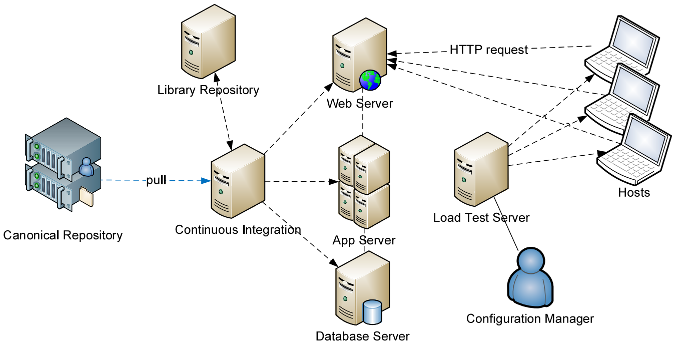


Figure 5. Quality Assurance (QA).

Staging Area aims to provide for monitoring users and product owners an environment as close as possible to production environment, so they perform approval tests. These ones are related to user experience and their perception regarding how software product meets specified requirements. This area has a copy of operating environments, both in terms of operating systems, tools and settings, and in terms of data. Monitored users are the ones chosen to perform approval tests in a monitoring way. Occasionally, they are in the product owner role. Figure 6 shows this area.

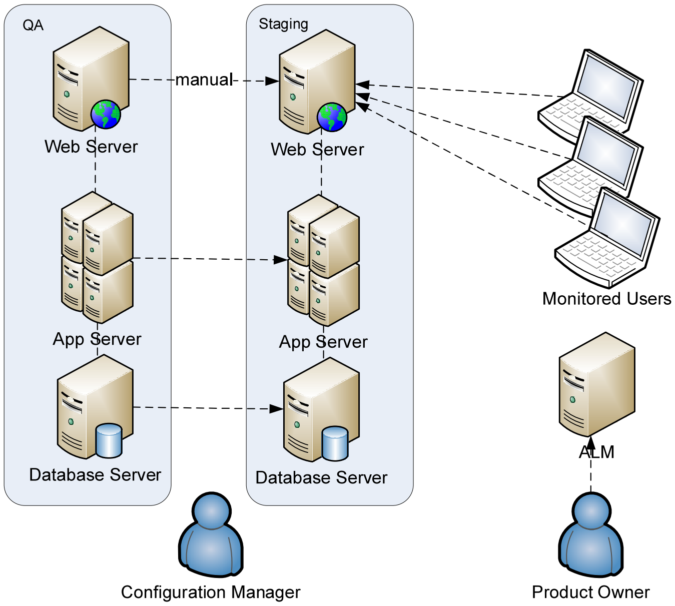


Figure 6. Staging (ST).

ISBN: 978-1-61208-399-5

Finally, configuration manager makes promotion from Staging Area artifacts to Production Area manually by Configuration Manager. However, developers and infrastructure staff are present to perform this task. Figure 7 shows this area.

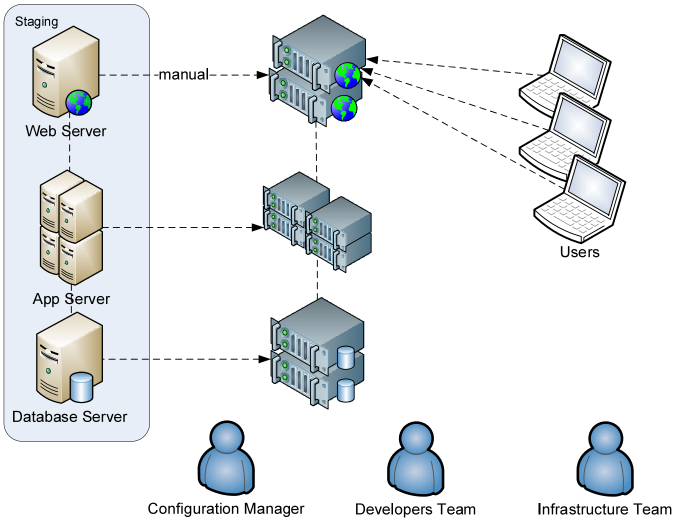


Figure 7. Production (PD).

Also, the following servers were used: (1) Log Server and (2) LDAP Server. The first has a very important function in the setup; to get all events occurred by indexing logs. This assists the diagnosis, providing information for reporting, alerts and dashboard. The tool used in this case is Splunk. The second server has a function to allow authentication and authorization for all setup servers. This is necessary because it is costly to maintain users across all the servers involved in an individualized way, in addition this increase security flaws. The tool used in this case is OpenLDAP (openldap.org).

C. Tools

Table I summarizes all tools used with its URL. These tools are used to Configuration Management (Git, Gerrit, Nexus, Flywaydb and Vagrant), Continuous Integration (Jenkins and Maven), Quality Assurance (SonarQube and Jmeter), Application Lifecycle Management (Redmine) and infrastructure (Splunk and OpenLDAP).

TABLE I. TOOLS USED.

Goal	*Name*	*URL*
Continuous Integration	Jenkins	jenkins-ci.org
Source Repository	Git	git-scm.com
Build	Maven	maven.apache.org
Gathering Logs	Splunk	splunk.com
Peer-Review	Gerrit	code.google.com/p/gerrit
Static Analysis	SonarQube	sonarqube.org
Load Test	Jmeter	jmeter.apache.org
Library Repository	Nexus	sonatype.org/nexus
Application Lifecycle Management	Redmine	redmine.org
Database Migration	Flywaydb	flywaydb.org
Automated Installation	Vagrant	vagrantup.com
Authentication and authorization	OpenLDAP	openldap.org

These tools were used because they are open/free software.

IV. RESULTS

Preliminary results about this approach are related to automation of many delivery tasks, resulting in a more predictable and managed process. Another aspect, related to collaboration, is due to communication between developers and infrastructure team was increased in all aspects of the process, since planning of a feature until its publication. These results are classified in a process maturity level [2], as shown in Figure 8.

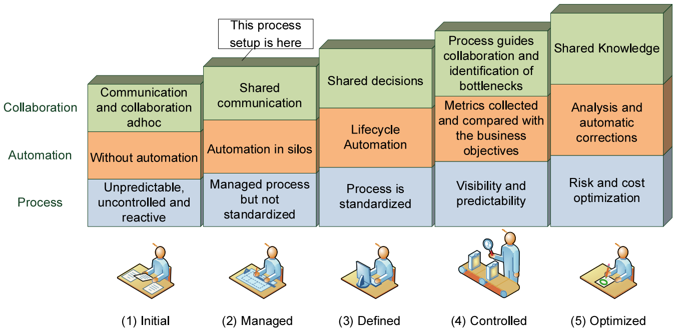

This process setup is here
Collaboration
Automation
Process
Communication and collaboration adhoc
Without automation
Unpredictable, uncontrolled and reactive
Shared communication
Automation in silos
Managed process but not standardized
Shared decisions
Lifecycle Automation
Process is standardized
Process guides collaboration and identification of bottlenecks
Metrics collected and compared with the business objectives
Visibility and predictability
Shared Knowledge
Analysis and automatic corrections
Risk and cost optimization
(1) Initial
(2) Managed
(3) Defined
(4) Controlled
(5) Optimized

Figure 8. Process maturity level [2].

V. CONCLUSION

This work presents a practical approach that can be used in similar processes. Additionally, among the contributions can be mentioned (1) a set of tools evaluated and (2) a set of techniques, that can be used for organizations that do not use this type of approach, as for those which already have a higher level of maturity.

Moreover, some further work may be developed to improve setup provided in this article. The first one aims to get a strategy for publication with less impact in terms of unavailability of software products, including deployment across different timezones. The second one is linked with multiple projects scenarios. We can analyze how the artifacts, from several projects, are promoted to production by the same team.

Finally, this article has a practical purpose. However, to implement continuous delivery involves more than installing some tools and automate some tasks. It depends on effective collaboration among all of those involved in the delivery process, senior management support and especially the desire of stakeholders in become the changes a reality.

REFERENCES

[1] M. V. Mantyla and J. Vanhanen, “Software Deployment Activities and Challenges - A Case Study of Four Software Product Companies,” 2011 15th European Conference on Software Maintenance and Reengineering, Mar. 2011, pp. 131–140.

[2] J. Humble and F. David, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. ser. Addison-Wesley Signature Series. Pearson Education, 2010.

[3] T. Ernawati and D. R. Nugroho, “IT Risk Management Framework Based on ISO 31000:2009,” International Conference on System Engineering and Technology, vol. 11, 2012, pp. 1–8.

[4] B. Fitzgerald, “Continuous Software Engineering and Beyond : Trends and Challenges Categories and Subject Descriptors,” RCoSE 14, 2014, pp. 1–9.

[5] S. Krusche and L. Alperowitz, “Introduction of Continuous Delivery in Multi-Customer Project Courses Categories and Subject Descriptors,” ICSE Companion 14, 2014, pp. 335–343.

[6] S. Bellomo, N. Ernst, R. Nord, and R. Kazman, “Toward Design Decisions to Enable Deployability: Empirical Study of Three Projects Reaching for the Continuous Delivery Holy Grail,” 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Jun. 2014, pp. 702–707.

Meta-Theory and Machine-Intelligent Modeling of Systemic Changes for the Resilience of a Complex System

Roberto Legaspi
Transdisciplinary Research Integration Center
The Institute of Statistical Mathematics
10-3 Midori-cho, Tachikawa, Tokyo, Japan
e-mail: legaspi.roberto@ism.ac.jp

Hiroshi Maruyama
Department of Statistical Modeling
The Institute of Statistical Mathematics
10-3 Midori-cho, Tachikawa, Tokyo, Japan
e-mail: hm2@ism.ac.jp

***Abstract*— Resilience is the ability of a complex system to persist in, adapt to, or transform from dramatically changing circumstances. Our objective is to characterize the resilience of a complex system in depth by looking at what fundamentally constitutes and leads to system changes and how the system can be resilient to these changes. Our characterization is by a two-fold framework, i.e., with a meta-theory that integrates long-standing foundational theories of systemic change and a two-part machine-intelligent computational modeling, specifically, using network analysis and machine learning models, to realize our meta-theory. By starting with a meta-theory as background knowledge to guide our modeling, we avoid irrelevant, scattered and loosely knitted paradigms. Complementary, any truth presented by the inferred models that are not accommodated in the meta-theory may correct flaws in the meta-theory. To our knowledge, our framework that uses this linking of meta-theory and machine-intelligent modeling to characterize resilience is novel. The results we obtained from our simulations show that our framework is a systematic and pragmatic way of inferring predictive models of the contextual interaction behaviors of a resilient system.**

Keywords-dynamic system theories; resilience theories; system evolution theories; intelligent systems.

I. INTRODUCTION

We have witnessed in the past 10 years unprecedented massive devastations in terms of human lives, livelihoods and infrastructures brought about by strong natural hazards, such as Hurricane Katrina in 2005 that is considered to be one of the deadliest hurricanes in U.S. history, the Haiti earthquake of 2010 with its catastrophic magnitude of 7.0M$_w$, the 9.0M$_w$ undersea megathrust earthquake off the Pacific coast of Tōhoku Japan and the Fukushima nuclear power plant disaster that happened in its wake, and Haiyan in 2013 that is one of the strongest trophical cyclones ever recorded. To say, however, that our reality is mostly a series of mild and insignificant events punctuated by only a handful of massive devastations is inaccurate. The reality is that the occurrences of car, train and airplane crashes, sinking ships, oil, chemical and radiation spills and leaks, terrorist attacks, and spread of viruses, among others, are more frequent than we think. These so-called *normal accidents* [42] dictate the quick, frequent and incremental critical adaptations of our systems [27]. Then we can add to these the catastrophic events that are difficult to model and predict given their ill-defined and non-computable nature, or the so-called *Black Swans* and *X-events* (citations in [26][32]), which compel our systems to carry out dramatic and novel adaptations in order to survive and sustain their existence.

In other words, accidents and disasters are actually common and inevitable [27][47], hence, our systems that keep us, our way of life, and our world existing and flourishing must be *resilient*, i.e., able to withstand even large perturbations and dramatically changing circumstances and preserve its core purpose and integrity [53], and achieve generalized recovery once failure due to perturbation is inevitable [32]. While resilience theory has been adopted in various fields including ecology, biology, economics, finance, engineering, social science, and of course, human development (noteworthy surveys can be found in [31][32]), we argue for deepening further the analysis of what makes a system resilient through a deeper understanding of what *fundamentally* (foundational) constitutes and leads to systemic changes and how the system can be resilient through undesirable changes. To be resilient also means to embrace change [31]. We position our argument with the long-standing theories of system complexity, chaos, self-organization, and criticality, all of which are interesting emergent properties shared by complex systems and have been used to explain biological evolution [23][24], capacity for computation in physical systems [39][25][36], evolution of natural and socio-ecological systems [2][21][40][41], and the collapse of social systems [46][35][13][9]. We integrate essential concepts of these theories in varying grains of analyses and view this integration as a *meta-theory*.

We also argue for the use of a two-part machine-intelligent modeling, specifically, using network analysis and machine learning approaches, to automatically discover the hidden rules of contextual interaction behaviors of a complex system. By starting with a meta-theory as background knowledge to guide our modeling, we avoid scattered and loosely knitted paradigms. Complementary, any truth present in the inferred models that is not accommodated in the meta-theory shall correct flaws in the meta-theory. Our meta-theory and machine-intelligent models can evolve together with increasing predictive isomorphism [34] to accurately represent the phenomena present *in*, i.e., endogenous (e.g., emergent properties, complexity, chaos, adaptation, and transformation, among others) and *with*, i.e, exogenous (e.g., disturbances, stress, and shocks), a complex system. To our knowledge, this linking of a meta-theory and two-part intelligent modeling to automatically characterize the contextual interaction behaviors of a resilient system is novel.

Our paper is structured as follows. We elucidate in detail our meta-theory in Section II and discuss in length our machine-intelligent modeling approaches and simulation results in Section III. We make a final defense of our framework in Section IV and then conclude in Section V.

II. Our Meta-Theory

Figure 1 shows our meta-theory that cohesively puts together theories on complexity, chaos, self-organization, critical transition and resilience. The complex system evolution cycle in our meta-theory involves three regimes, namely, order, critical, and chaos. The second ordered regime, however, may be novel in the sense that it required the system to transform when adaptation back to the previous state was no longer attainable. The moving line indicates system "fitness", i.e., the changing state of the system in terms of its capacity to satisfy constraints, its efficiency and effectiveness in performing tasks, its response rate (time to respond after experiencing the stimuli), returns on its invested resources or capital, and/or its level of control.

It is through our meta-theory that we can view a complex system as *open*, i.e., always in the process of change and actively integrating from, and disseminating new information to, changing contexts, as well as *open-ended*, i.e., it has the potential to continuously evolve, and evolve ways of understanding and manipulating the contexts (endogenous and exogenous) that embed it [48]. Both characteristics are vital for the complex system to be resilient.

Our succeeding elucidation of our meta-theory, and the references that accompany our elucidation, would attest to the fact that the individual components, i.e., theories, which comprise our meta-theory are neither from a vacuum nor just mere speculations as they are evident in physics, ecology, biology, and system dynamics. What we are presenting here, however, is a plausible integration of these theories.

While complexity theory focuses on how systems consisting of many diverse elements give rise to well-organized, predictable behavior, chaos theory concerns itself with how simple systems pave the way for complicated nonlinear unpredictable behavior. Self-organization holds that structures, functions, and associations emerge from the interactions between system components and their contexts. The critical regime, which we pay special attention to due to its importance, holds significant paradoxes – it may herald an unwanted collapse or become a harbinger of positive change [44], and while it may signal hidden fragilities [12], it is also theorized to facilitate complex computations, maximize information storage and flow, and be a natural target for selection because of its hidden characteristics to adapt [25][23][36]. While we adopt the terms order, critical, and chaos from dynamical systems theory [49], to persist, adapt, and transform is resilience thinking [14][8][10].

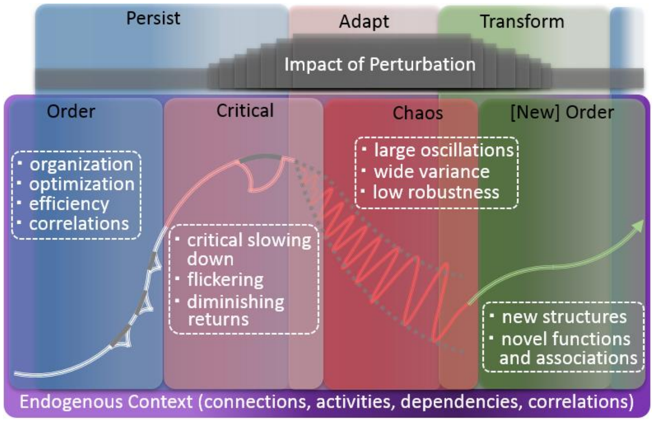


Figure 1. We integrate in varying grains of analyses how the different theories are plausibly related – hence, a *meta-theory*.

In the ordered regime, connections, interdependencies, and correlations begin to emerge. In this stage, the system will control and manage change. It will always attempt to re-establish equilibrium in order to persist in its ordered state each time it is perturbed (indicated by the dents in fitness). When it encounters a perturbation, it should readily bounce back and recover. System adaptations, however, will only be small, moderate, segmented and gradual, which are sufficient to handle only the manageable perturbations. The system changing or becoming permanently damaged from shock is not a major concern in this phase. To resume normal operations immediately and distort less in the face of minor perturbations is an increasing trait.

The ordered regime is a slow process characterized by increasing system efficiency and optimization of processes. What is also increasing, however, is the connectedness or tight coupling of the system components. Furthermore, the system's self-regulation becomes more finely tuned to the set of perturbations and responses it became familiar with. These tight coupling and rigidity only make the impact of any perturbation, regardless of its magnitude, to also increase. All this build-up is like an accident in the wings waiting to happen. Eventually, the system shall converge to a state that makes itself less adaptive to perturbations and therefore brings itself to the critical regime, which is at times also called the "edge of chaos" [25][23][36].

Scheffer et al. [44] elucidate the behaviors displayed by the system in the critical regime. One is a critical slowing down, i.e., the rate at which a system recovers from small perturbations becomes slow. Flickering may also be observed wherein a highly stochastic system flips to an alternative basin of attraction when exposed to strong perturbations. Page [40] added diminishing returns, which is the decrease in some system performance measure such as efficiency, robustness, or accuracy.

Comes a point when complexity can no longer be sustained and persistence and small adaptations are no longer possible, and so the system enters the chaotic regime. The building up of complexity becomes a constraint to adaptation and eventually leads to chaos. In the chaotic phase, the system will need larger adaptations, otherwise, it will need to transform to a new ordered regime to survive – one that will require dramatic change of structure and function.

Systems that demonstrate a transformative capacity can generate novel ways of operating or novel systemic associations and can recover from extreme perturbations [31]. Such systems learn to embrace change [31], and instead of bouncing back to specification, which is proved vulnerable and led to chaos, they bounce *forward* to a new form [29].

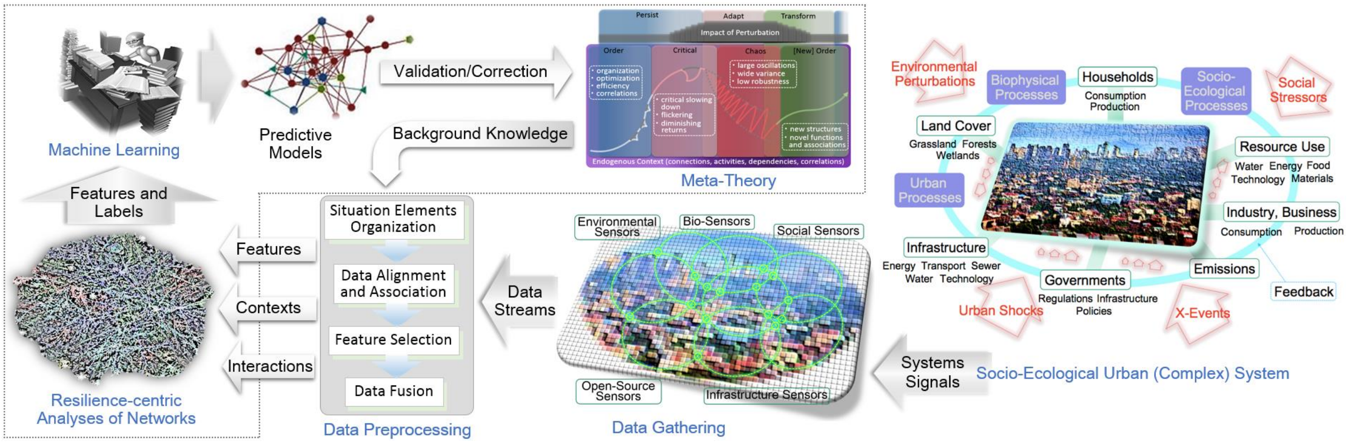


Figure 2. Our entire complex systems resilience modeling architecture, which includes our two-fold framework (enclosed in dotted lines).

III. Machine-Intelligent Modeling of the Resilience of a Complex System

When we speak of complex system properties, we speak of system-wide behaviors emerging from the interaction and interdependencies of diverse system components. To be more concrete, our long-term objective is to model a *socio-ecological urban system* (SEUS), as shown at the right side of Figure 2, where diverse components, which can be systems in themselves [5][50], are intricately connected and may at times display rather extreme interdependencies. This SEUS contains continuous flow of resources, information, energy, capital, commerce, and people. The sustainability of a component will critically depend on its place in the system and how it, and the entire SEUS, can withstand perturbations.

The meta-theory shall be the by-product of integrated transdisciplinary perceptions of what characterizes systems resilience. We can develop social computing platforms for the collaboration and integration of expert and experiential knowledge (e.g., of aborigines and natives whose breadth and depth of experiential knowledge make their lack of formal education insignificant) [26]. We can also develop knowledge extraction and integration technologies that can infer relationships that exist among knowledge from largely varying domains and can synthesize individualized, micro-level, and domain-dependent knowledge towards contextual systemic knowledge that can lead to actionable information for resilience. Such actionable information, for example, can be in the form of a repository of evidences of what works (predictive) and may work (innovative) in a situation (e.g., disaster prevention, mitigation and management).

To gather large amount of data to model the SEUS, ubiquitous smart and interacting daily-living objects can offer a wide range of possibilities [6][43]. Urban services, such as vehicular traffic, banking, purchasing, personal security for citizens, social services, and tourism support, among others, have been recently enhanced by digital networks and mobile technology. Digital networks may also control sanitation and waste, water, traffic, communication, and energy. Tiny interacting embedded systems could also play a valuable role in protecting the environment from perturbations, e.g., sensors so minute, as the size of dust particles, but can detect the dispersion of oil spills or forest fires [6]. Furthermore, signals can come from volunteered data of people's use of mobile devices, social media, web searches, and online transactions, among others, which reflect human cognitive, affective and social behavior patterns. Using distributed multisensing capabilities and information processing, it is possible for the machine intelligence in the SEUS to infer accurate and informative models for situation analysis, situation awareness, decision-making and response, and component feedback in order for the SEUS to sense and shape the contexts that embed it.

Heterogeneous data related to humans, environments, and technologies, and their interactions will often be reported or obtained from a multiplicity of sources, each varying in representation, granularity, objective, and scope. Data pre-processing techniques can be employed to organize, align, and associate input data with context elements. With feature selction, it can also reveal which features can help improve concept recognition, generalization and analysis. Lastly, data fusion can address data and algorithmic complexities and the associated challenges that arise when independent data sources are combined to improve the quality of information.

All the pertinent features, contexts and interactions inferred in the preprocessing stage will be used in our two-part machine intelligent modeling. First, these information will be organized, represented and analyzed as a network. Paperin et al. [41] provide an excellent survey of previous works that demonstrated how complex systems are isomorphic to networks and how many complex properties emerge from network structure rather than from individual constituents. Second, using as inputs the network and resilience properties of the system, machine learning will be used to infer the *relational rules* of system contextual interaction behaviors that define its adaptive and transformative walks and therefore define its resilience. Our modeling will capture how the complex system's ability to vary, adjust or modify the connectivity, dynamism, topology, and linkage of its components (endogenous features), and its capacity to withstand the disturbances (exogenous feature) that perturb it, will dictate its resilience.

A. Simulation of a Complex System and its Properties

Although our aim is to model a socio-ecological urban system and its intricate properties, our major concern at this time, however, is that we have yet to embark on this endeavor. However, as a more than plausible work-around to this lack of complex system to analyze the viability of our framework, we used random Boolean networks (RBNs) to simulate the behavior of a complex system. The question, however, is whether the use of a RBN in lieu of an actual complex system plausible in demonstrating our concepts?

The literature is rich with RBNs being models of large scale complex systems [1][20]. RBNs are idealizations of complex systems where its systemic elements evolve [11]. They are general models that can be used to explore theories of evolution or even alter rugged adaptive landscapes [16]. Furthermore, although RBNs were originally introduced as simplified models of gene regulation networks [22][23][24], they gained multidisciplinary interests since they could contribute to the understanding of underlying mechanisms of complex systems, albeit their dynamic rules are simple [52], and because their generality surpassed the purpose for which they were originally designed [34][52][30][17].

A RBN consists of *N* Boolean nodes, each linked randomly by *K* connections. The state of a node at time t+1 depends on the states of its *K* inputs at time *t* by means of a Boolean function. The randomly generated Boolean functions can be represented as lookup tables that represent all possible 2^K combinations of input states. *N* represents the number of significant components comprising an adapting entity, such as gene, chromosome, trait, species, process, business unit, firm, traders, bankers, or workers – generally the number of agents attempting to achieve higher fitness [34]. The Boolean values may represent, for example, contrasting views, beliefs and opinions, or alternatives in decision-making (e.g., buying or selling a stock [24], cooperating with community or not). We can view *K* conceptually as affecting the mutual influence among nodes in an information network [52] since a directed edge $<x, y>$ means that agent *y* can obtain information from, and can be influenced by, agent *x*. In this way, *K* is proportional to the quantity of information available to the agent [52].

How complex can a RBN be? Given *N* and *K*, there can be $(N!/(N\text{-}K)!)^N$ possible connectivity arrangements, $(2^{2^K})^N$ possible *N* Boolean function combinations, and $((2^{2^K} N!)/(N\text{-}K)!)^N$ RBNs [19]! This is not counting the many possible updating schemes [16], and possibly extending to have nodes with multiple states [45]. With these huge number of possibilities, it is therefore possible to explore with RBNs the various properties of even large-scale complex systems and their many possible contexts [20].

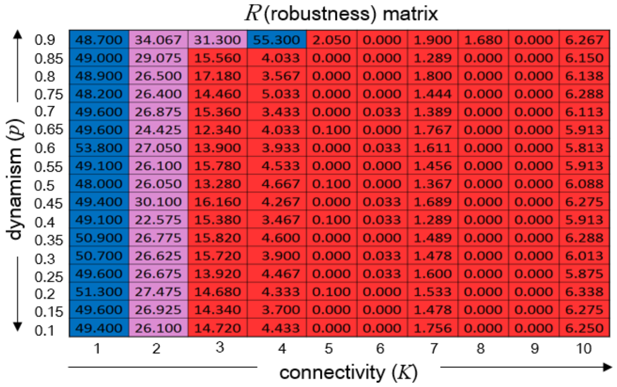
Inherent to RBNs are certain parameters that we found having accounted for by our meta-theory. At the same time, these parameters can be the controlling variables that the system can modify or adjust to demonstrate its resilient capabilities. In a plausible sense, these parameters can be viewed as *simulated* (but possible) outputs of the pre-processing stage of our architecture (in Figure 2) that led to the construct of the network. The parameters are as follows:

- *Connectivity* (*K*). This refers to the maximum or average number of nodes in the input transition function of a network component. As we increase *K*, nodes in the network becomes more connected or tightly coupled, and more inputs affect the transition of a node.
- *Dynamism* (*p*). A Boolean function computes the next state of a node depending on the current state of its *K* inputs subject to a probability *p* of producing 1 in the last column of the lookup table. If p=1 or p=0, then there is no actual dynamics, hence low activity, in the network. However, *p* close to 0.5 gives high adynamical ctivity since there is no bias as to how the outputs should be [17]. High dynamical activity means high variability.
- *Topology* (or *link distribution*). A RBN may have a fixed topology, i.e., all transition functions of the network depend on exactly *K* inputs, or a homogeneous topology, i.e., there is an average *K* inputs per node. Another type of topology is scale-free, where the probability distribution of node degree obeys a power law. In an information network, a scale-free property means that there is a huge heterogeneity of information existing [52], hence, there is more variation in the network. Following [38], the number of inputs for the scale-free topology is drawn from a Zeta distibution where most nodes will have few inputs, while few nodes will have high number of inputs. The shape of the distribution can be adjusted using the parameter γ (set to 2.5) – when γ is small/large, the number of inputs potentially increases/decreases.
- *Linkage* (or *link regularity*). The linkage of a RBN can be uniform or lattice. If the linkage is uniform, then the actual input nodes are drawn uniformly at random from the total input nodes. Following [38], if the linkage is lattice, only input nodes from the neighbourhood $(i\text{-}lattice_i{*}k_i)$:$(i{+}lattice_i{*}k_i)$ are taken, where *i* is the position of the node in the RBN and $lattice_i$ is its lattice dimension whereby nodes are dependent to those in the direct neighborhood. A wider lattice dimension can lead to a RBN with highly interdependent nodes.

B. Simulation Models, Results, and Analyses

We now discuss our various simulation models starting with the base case. Our *base case* is a “conventional” RBN wherein the topology is fixed and the nodes are updated at the same time by the individual transition functions assigned to each, i.e., synchronous update. With several conditions to check, we used for now a single value for *N* (i.e., 20).

The simulations we conducted involved testing the RBN's *robustness* when faced with perturbations. We applied the program of Müssel et al. as outlined in their BoolNet vignette [38] as follows. A perturbation is achieved by randomly permutating the output values of the transition functions, which although preserved the numbers of 0s and 1s, may have completely altered the transition functions. For each simulation a total of 1,000 perturbed copies of the network were created, and the occurences of the original attractors in the perturbed copies were counted. Attactors are the stable states to which transitions from all states in a RBN eventually lead. The robustness, *R*, value is then computed as the percentage of occurrences of the original attractors.



R (robustness) matrix

dynamism (p) \ connectivity (K)	1	2	3	4	5	6	7	8	9	10
0.9	48.700	34.067	31.300	55.300	2.050	0.000	1.900	1.680	0.000	6.267
0.85	49.000	29.075	15.560	4.033	0.000	0.000	1.289	0.000	0.000	6.150
0.8	48.900	26.500	17.180	3.567	0.000	0.000	1.800	0.000	0.000	6.138
0.75	48.200	26.400	14.460	5.033	0.000	0.000	1.444	0.000	0.000	6.288
0.7	49.600	26.875	15.360	3.433	0.000	0.033	1.389	0.000	0.000	6.113
0.65	49.600	24.425	12.340	4.033	0.100	0.000	1.767	0.000	0.000	5.913
0.6	53.800	27.050	13.900	3.933	0.000	0.033	1.611	0.000	0.000	5.813
0.55	49.100	26.100	15.780	4.533	0.000	0.000	1.456	0.000	0.000	5.913
0.5	48.000	26.050	13.280	4.667	0.100	0.000	1.367	0.000	0.000	6.088
0.45	49.400	30.100	16.160	4.267	0.000	0.033	1.689	0.000	0.000	6.275
0.4	49.100	22.575	15.380	3.467	0.100	0.033	1.289	0.000	0.000	5.913
0.35	50.900	26.775	15.820	4.600	0.000	0.000	1.489	0.000	0.000	6.288
0.3	50.700	26.625	15.720	3.900	0.000	0.033	1.478	0.000	0.000	6.013
0.25	49.600	26.675	13.920	4.467	0.000	0.033	1.600	0.000	0.000	5.875
0.2	51.300	27.475	14.680	4.333	0.100	0.000	1.533	0.000	0.000	6.338
0.15	49.600	26.925	14.340	3.700	0.000	0.000	1.478	0.000	0.000	6.275
0.1	49.400	26.100	14.720	4.433	0.000	0.000	1.756	0.000	0.000	6.250

Figure 3. Base case: map of the different regimes based on the sensitivity of conventional RBNs to perturbations.

It is very important to realize that robustness here is *not* resilience per se, since resilience refers to *what* enables a system (such as change in connectivity, dynamism, topology, and linkage, among others) to preserve its core identity when faced with perturbations [53]. We used *R to quantify the amount of RBN core identity that was preserved*. Hence, *R* is an indicator or measure of systems resilience.

Figure 3 shows our base case *R*-matrix in a dynamism-connectivity space, where each component is a *R* value. We can observe that it is at K=1 that the RBNs were most robust. The RBNs losing robustness at K=2 is indicative of critical slowing down and the system may therefore be tipped more easily into an alternative state [44], i.e., from order to chaos, which therefore reflects criticality. Hence, the ordered phase is found when K<2, the chaotic phase occurs for K>2, while the critical regime lies at the phase transition, i.e., at K=2 [16]. We can therefore observe from the base case the regimes present in our meta-theory (blue is order, purple is critical, and red is chaos).

We now move on to the results of the various simulation models we ran, beginning with the one in Figure 4. Each rectangle in the 3×5 *topology-linkage* space is a *R*-matrix with *p*-*K* dimensions. For example, $R_{2\times 3}$ corresponds to the robustness matrix of RBNs with homogeneous topology and lattice linkage of size 2.5. $R_{1\times 1}$ is the same *R*-matrix in Figure 3. We can see from the *R*-matrices the interesting properties that emerged. We can observe the critical regime broadening to K=3 (e.g., $R_{1\times 2}$, $R_{2\times 2}$, $R_{2\times 3}$, etc.), or reoccurring at K>2 (e.g., $R_{1\times 3}$ and $R_{1\times 5}$) in between chaotic regimes, in the fixed and homogeneous RBNs with wider lattice. These extensions and reoccurrences of the critical regime mean alternative opportunities for the system to take advantage of the benefits of the critical regime and the balance of stability and chaos [17]. The wider lattice led to more interdependencies among nearest neighbors, which formed small world networks that brought about such behaviors of the critical regime. This is consistent with the findings of Lizier et al. [28] that a small world topology leads to critical regime dynamics.

Furthermore, the ordered regime expands with homogeneous RBNs. Since the number of input nodes is drawn independently at random, there is more variation in the way components influence each other. This also means that with less tighter connections among components (i.e., as the couplings in the network are loosened), the system becomes less vulnerable to perturbations. $R_{2\times 1}$, for example, shows how the system could transform to the next ordered state from a critical phase instead of deteriorating to a chaotic regime. With the scale-free topology, however, we can see highly robust RBNs. Since few nodes have more connections, and most nodes have few connections, changes can propagate through the RBN only in a constrained fashion. We also have evidence wherein the *over-all mean* robustness began to continuously decrease towards zero for the fixed and homegeneous topology at K=2, which we interpret as a form of diminishing returns before transitioning to the chaotic regime. The over-all mean robustness values for the scale-free RBNs, however, remained satisfactory throughout. A complex system may therefore demonstrate resilience by *broadening* (*extending*) *the critical regim*e, *making the critical regime reoccur*, or *changing to a scale-free topology*.

Lastly, by applying again the methods of Müssel et al. [38], we tested for the sensitivity of the RBNs to greater perturbations. In each network transition, the transition function of one of the components is randomly selected, and then five bits of that function is flipped. Figure 5 shows the results we obtained. The first interesting phenomenon is the multiple occurrences of the ordered (e.g., in $R_{2\times 1}$ and $R_{2\times 4}$) and critical regimes (e.g., in $R_{1\times 4}$, $R_{2\times 3}$, $R_{2\times 4}$, and $R_{3\times 1}$), even after the chaotic regimes, which can point to resilience. The

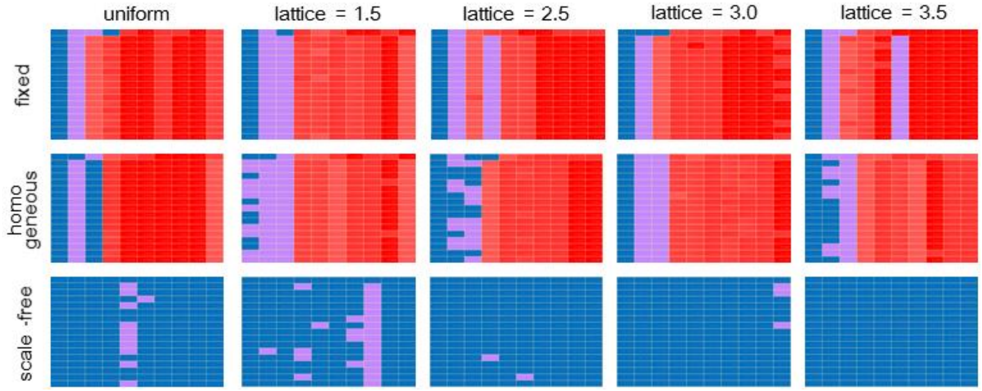


Figure 4. Map of the different regimes based on the sensitivity of RBNs to perturbations when activity, connectivity, topology, and linkage values were varied. The *R*-value ranges for each regime are as follows – order: [43,100] (in blue), critical: [22, 43) (in purple), and chaos: [0, 22) (in red).

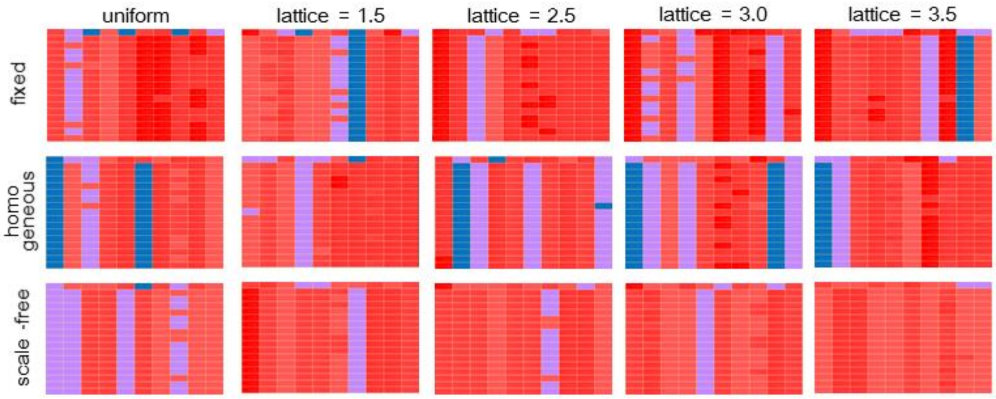


Figure 5. Map of the different regimes based on the sensitivity of RBNs to greater perturbations.

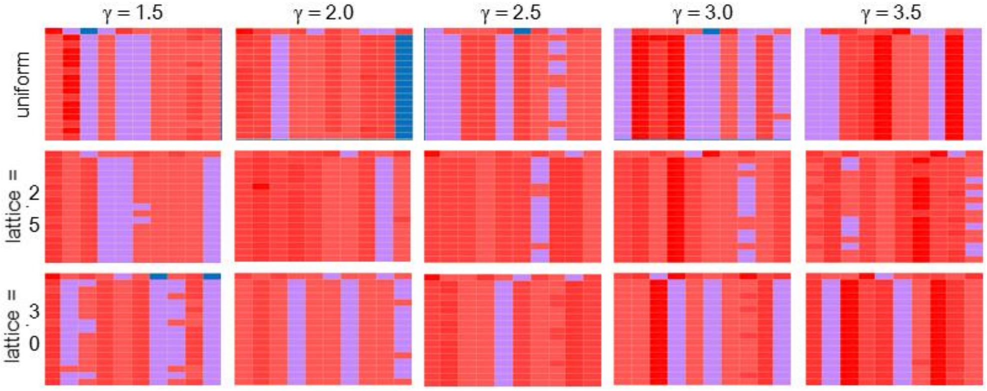


Figure 6. We simulated what will happen with changing γ values. The tables show that with other γ comes more expansions of the critical regime.

second is that we can obviously see how the behavior of the scale-free RBNs changed drastically, i.e., we could not find any ordered regime. This is consistent with the findings of Barabási and Bonabeau [3] that scale-free networks are very robust against random failures but vulnerable to elaborate attacks. In our case, flipping bits in each transition of the network was too much perturbation for the scale-free RBN. But this does not mean, however, that its resilience is entirely lost. When we varied the parameter γ of the Zeta distribution, another interesting phenomenon emerged as shown in Figure 6 – we see more expansions and reoccurrences of the critical regime given other γ values. Again, to be capable of *prolonging or increasing the number of critical regime occurrences is indicative of a system being resilient*.

C. Machine-Intelligent Modeling

We can see from our simulation results that the various parameters we used can quantitatively explain our meta-theory. It is clear that the combinations of their specific values can be used to predict system states and changes and steer the system to desirable regimes, i.e., resilient states. The question now is how to infer these parameter relations that can be used as rules of contextual interaction behaviors that define the complex system's adaptive and transformative walks and therefore define its resilience.

Our solution is to use machine learning (ML) to automatically discover the hidden relations from the data we obtained about the complex system. We represent system contextual interaction behaviors as sets of feature vector and label pairs. Each feature vector is represented as a tuple of attribute values, i.e., <*topology*, *linkage*, *lattice*, *gamma*, *connectivity*, *activity*, *perturbation*>, and labeled with the corresponding *R* value that is indicative of system regime.

The ML algorithm should infer a model that is predictive – given the feature vector, what is the system regime (and its robustness)? Furthermore, the predictive model is one that can be used to help steer the system to a desirable regime – from the current feature vector that indicates the contextual situation of the system, which may be undesirable given *R*, which features can or should be modified to achieve a desirable regime. *This capacity to modify the contexts and predict the resulting behavior can make the system resilient*.

Our dataset consisted of 7,120 feature vectors, which corresponds to the various simulation scenarios we ran using our different RBN models. It is important to note that even though our data can still be considered minimal (considering for example that we only used one value for *N*, limited value ranges for the parameters, and only synchronous updates), the advantage of using a data-centric approach is that as the data further increases, ML can be used to automatically handle growing intricacies and complexities, as well as automatically infer the new relations emerging from the data.

To obtain the model with the best predictive capacity, we ran several well-known ML algorithms that are (*a*) function-based: linear regression models (LRM), multi-layer perceptrons (MLP), radial basis function networks (RBFN),

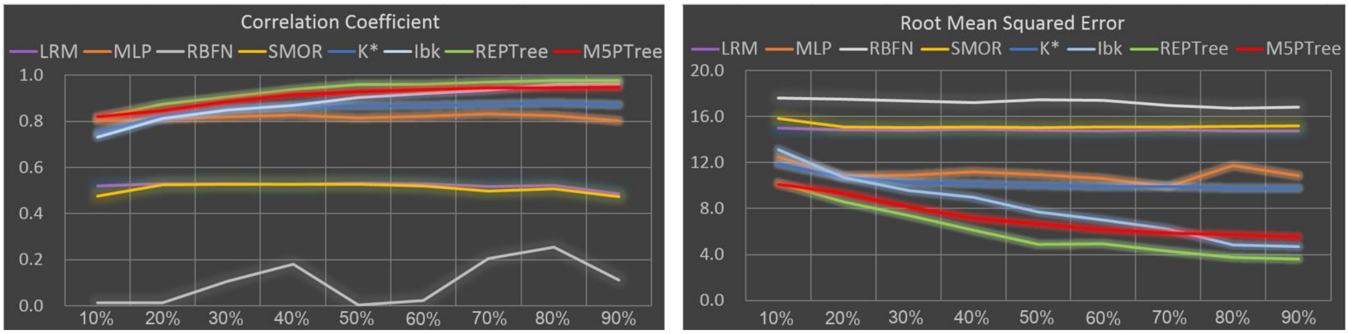


Figure 7. Prediction accuracy of the various models using %-split validation with increasing % values

and support vector machines for regression (SMOR), (*b*) instance-based or lazy: *K** and *k*-nearest neighbor (Ibk), and (*c*) tree-based: fast decision tree (REPTree) and MP5 model tree (MP5Tree), using the WEKA open-source software. Due to space constraints, it is best that we refer the reader to the documentation [51] of these algorithms. We used *%-split* validation where *x*% of the data was used for training and the rest for testing the accuracy of the model. We measured the performance of the regression analysis in terms of correlation coefficient and root mean squared error to show the strength of prediction or forecast of future outcomes through a model or an estimator on the basis of observed related information. The correlation coefficient is also indicative of how good the approximation function might be constructed from the target model. We constructed several models by increasing the size of the training set from 10% to 90% of the total data, with increments of 10% (horizontal axis of the graphs in Figure 7), which allowed us to see the performance of the inferred models with few or even large amount of data, and also gave us the feel of an incremental learning capacity

Figure 7 shows the accuracy of prediction of the models. We can see that the models inferred by the decision tree-based (REPTree and MP5Tree) and instance-based *k*-nearest neighbor (Ibk) algorithms outperformed the others. These models can accurately predict in more than satisfactory levels the contextual interaction behaviors of the system even with only 10% of the data. We note that our goal at this time is not to improve the algorithms or discover a new one, but to prove the viability of our framework. We anticipate, however, that as the complexity of the system and the data grows, our algorithms may also need to improve.

The other advantage of the tree-based models is that the relation rules can be explicitly observed from the tree. Model trees are structured trees that depict graphical if-then-else rules of the hidden or implicit knowledge inferred from the dataset [4][18]. Model trees used for numeric prediction are similar to the conventional decision trees except that at the leaf is a linear regression model that predicts the numeric class value of the instances reaching it [18]. Figure 8 shows the upper portion (we could not show the entire tree of size 807 due to space constraints) of the REPTree we obtained using 10%-split validation with the elliptical nodes representing the features (colored so as to distinguish each feature), the edges specifying the path of the if-then-else rules, and the square leaf nodes specifying the corresponding *R*-values depending on which paths along the tree were selected. We can see how the rules delineated in a fine-grained manner the attribute values that eventually led to satisfactory predictions. We can also see how certain features are more significant to the classification task even early in the tree. The connectivity feature, for example, is prominent in both sides of the tree, and that the dynamism feature is not as significant in the upper levels of the tree as compared to the lattice feature. All these mean that by observing the tree, we can determine which features are significant not only to the classification task, but more importantly to a more relevant sense, which features are actually influential to the resilient (as well as vulnerable) walks of the system.

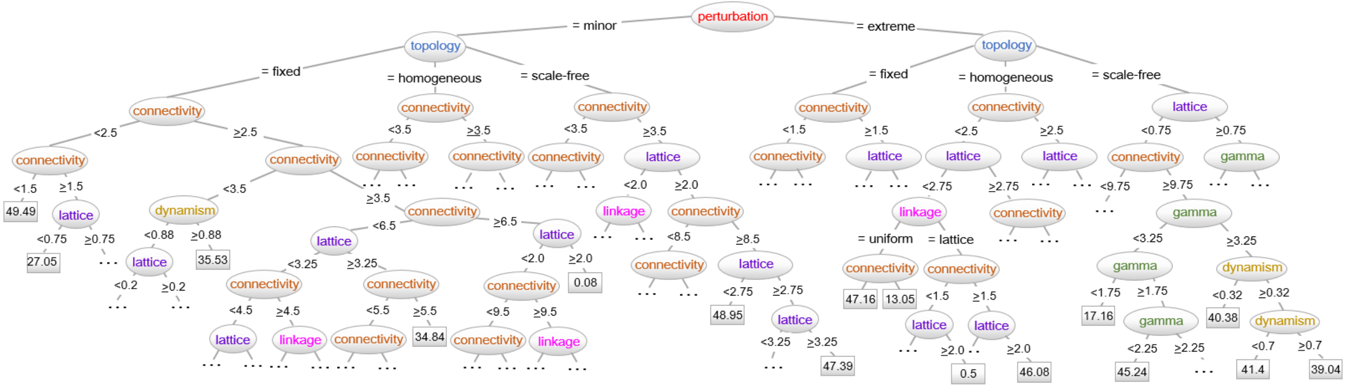


Figure 8. REPTree generated using Weka with a 10%-split validation. The size of the tree is 807, but only parts of it can be shown due to space constraints. The nodes specify the features (colored so as to distinguish each) with the edges as attribute values, and the leaf nodes as *R*-values.

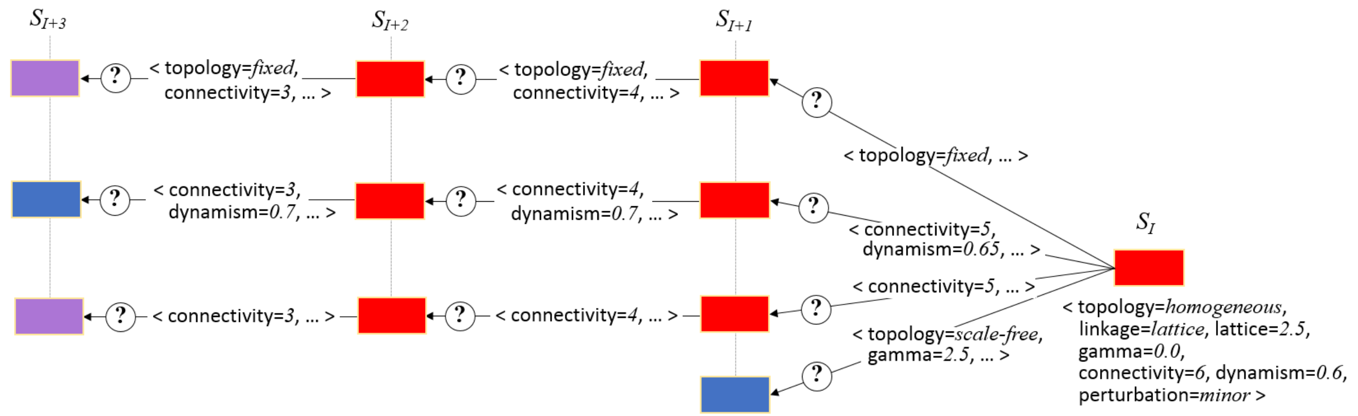

S_{I+3}
S_{I+2}
S_{I+1}
S_I
?
< topology=*fixed*, connectivity=*3*, ... >
< topology=*fixed*, connectivity=*4*, ... >
< topology=*fixed*, ... >
< connectivity=*3*, dynamism=*0.7*, ... >
< connectivity=*4*, dynamism=*0.7*, ... >
< connectivity=*5*, dynamism=*0.65*, ... >
< connectivity=*3*, ... >
< connectivity=*4*, ... >
< connectivity=*5*, ... >
< topology=*scale-free*, gamma=*2.5*, ... >
< topology=*homogeneous*, linkage=*lattice*, lattice=*2.5*, gamma=*0.0*, connectivity=*6*, dynamism=*0.6*, perturbation=*minor* >

Figure 9. Illustration of how the strength of the predicitve models can be used to find the desirable regime states. The regime states (colored blocks) and their contextual features (in angle brackets) were taken from the robustness maps, i.e., $R_{2\times3}$, $R_{1\times3}$, and $R_{3\times3}$, in Figure 4.

Lastly, we refer to Figure 9 to illustrate how our strong predictive model can be used to help steer the system to a desirable regime. The regime states shown in the figure, which were taken from the regime maps in Figure 4 (specifically, from $R_{2\times3}$, $R_{1\times3}$, and $R_{3\times3}$), are obviously only a tiny portion of the possible entire regime search space since each cell in every R-matrix in Figures 4 to 6 is a regime state. Let us say that the system landed in the chaotic regime S_I, hence undesirable, as a result of the situational context (indicated by the feature vector shown below it) it found itself into. The predictive model can be used to predict the resulting regime when one or more of the S_I contextual features are changed. Hence, from the current regime state S_I, depending on which features are changed, the system may enter in one of the many possible S_{I+1} regime states. Although it seems elementary for the system to follow the prediction that suggests changing to scale-free topology with γ=2.5 in order to immediately reach a new ordered state, what should be considered is the high cost of changing to a topology that will necessitate breaking many of the current ties (e.g., geophysical, relational, monetary, etc.). Hence, it may be more advantageous for the long haul for the system to seek alternative paths with longer chaos, but less painful and costly. This capacity to modify contextual features and predict the resulting regime demonstrates systems resilience.

The next challenges we need to consider in our future work towards a truly strong predictive capacity is greater. One formidable challenge in determining the optimal path to the desirable regime is the possibly thousands, millions or even astronomical number of potential paths, each with its own set of multiple candidate divergence. Without special algorithms to find the correct paths efficiently, the required computing time might be prohibitive. Equally challenging is the notion that the shortest path is not necessarily the optimal one. Again, the longer path may in fact possess the more bearable pain, trauma and adjustment compared to the immediate, but extreme and radical, change. Hence, what is significantly missing in our modeling is the quantifiable cost associated to each change. Although we can account for the actual cost accurately only in retrospect, the challenge is for us to find the function that can meaningfully approximate the cost of system adaptation and transformation.

IV. In Defense of Our Framework and Approach

With our world witnessing critical systemic changes [26], we are concerned with how our systems can be resilient, i.e., it is able to persist in, adapt to, or transform from dramatically changing circumstances. We believe that a deeper understanding of what fundamentally constitutes and leads to critical system changes sheds light on our understanding of the resilience of our systems.

Our solution of mutual reinforcing between theoretic and data-centric models allows for less perfect theory and inferred models to begin with, but with both components learning mutually and incrementally towards improved accuracy. Through our meta-theory we are able to have a strong basis of what will constitute our machine intelligent modeling. What the meta-theory can take from the inferred models, however, is to improve its knowledge by incorporating the fine-grained features, e.g., changing lattice and γ values, as well as the magnitude of the perturbations, which can have specific influences towards specific regimes. The meta-theory has to incrementally improve its knowledge based on what is being discovered by the intelligent modeling component. Our theoretic and data-centric models will surely need to co-evolve as we collect more data with increased range of network parameter values, other ways of introducing perturbations, using different transition schemes [16], and with agents having multiple states [45], among others. Furthermore, as nonlinear and unpredictable system intricacies become more detailed and pronounced, our machine intelligent modeling should account for emerging algorithmic and data complexities. This mutual reinforcing of theory and intelligent models is not found even in well-established frameworks, such as the Adaptive Cycle [21], Self-organized Criticality [2], and Dual-Phase Evolution [41].

Our framework is data-centric as opposed to using formal verifications. We can argue that formal or mathematical verification does not always guarantee reality and is not absolutely reliable. It can even fall short given the computational intractability of complex systems. The intractability of a complex system state space leads to issues of big data, which is where machine learning inference becomes viable. Furthermore, formal models tend to abstract much of the realistic nonlinear and stochastic intricacies of the system's internal workings [37].

Due to the absence of our intended real world complex system data, we simulated the viability of our framework using random Boolean networks (RBNs). If RBNs are in fact general models of complex systems, then our simulations would have sound basis – which is actually the case. RBNs are models of self-organization in which both structure and function emerge without explicit instructions [54]. Secondly, it is by the random nature of RBNs, albeit the transition functions are fixed, that systemic behaviors that emerge from known individual component behaviors cannot be determined a priori (e.g., exact number and characteristics of possible basins of attractions). All these and that a RBN's "infusion of historical happenstance is to simulate reality" [11, p.88] may attest to the fact that our meta-theory being demonstrated by RBNs is not at all forced.

Lastly, we also believe that our proposed framework's contribution is to help solve the problem of persistently having linear, fragmented and incomplete knowledge in our theories and models. This insufficiency of knowledge is because our system and the contexts that engulf it are complex, indeed chaotic, and their behaviors are nonlinear, spanning multiple simultaneous temporal and spatial scales, and with large interrelations and interdependencies among variables. And even though we are fully cognizant of their non-computable aspects [7][15], we continue to wrap our minds around what is only computable [7]. All these lead to shallow and disconnected understanding of the evolving nature of our systems and the phenomena that consist and embed them. We address this problem by having knowledge integration and incremental learning in our framework, i.e., (*i*) the integration of transdisciplinary knowledge via our meta-theory, (*ii*) the integration into data-centric models of low-level signals or features of various phenomena that are endogenous (e.g., interdependencies, dynamism, topology, connectivity, etc.) and exogenous (e.g., perturbations) to the complex system, and (*iii*) the mutual reinforcing and incremental knowledge learning of the meta-theory (theoretic) and intelligent models (data-centric) that shall lead to increased predictive isomorphism [33][34].

V. Conclusion

We argued for a framework that characterizes what fundamentally constitutes complex system change by cohesively integrating concepts in complexity, chaos, self-organization and critical transition theories into one meta-theory. The meta-theory states that what comprises system change are the changing contexts, the fitness of the system to continuously evolve, and the capacity of the system to evolve its understanding and manipulation of the context.

We then argued for the use of networks and machine learners to quantitatively explain what leads to system change and how the system can adapt to and transform through change. Our network-centric analyses show that the ability by which the system can vary, adjust or modify its controlling variables, specifically those that pertain to the connectivity, dynamism, topology, and sphere of influence of its components (all endogenous), and its capacity to withstand the disturbances (exogenous) that perturb it, will dictate the rules of its adaptation and transformation. We obtained these rules as relations of system controlling variables by mining the data using ML algorithms instead of the conventional abstract mathematical formulations.

The meta-theory and intelligent modeling link will need to evolve as we collect more data with increased range of system endogenous and exogenous parameter values and more ways of introducing perturbations.

References

[1] R. Albert and A.-L. Barabási, "Dynamics of complex systems: scaling laws for the period of boolean networks," Phys. Rev. Lett., vol. 84, no. 24, June 2000, pp. 5660-5663.

[2] P. Bak, How Nature Works: The Science of Self-Organised Criticality. New York, NY: Copernicus Press, 1996.

[3] A.-L. Barabási and E. Bonabeau, "Scale-free networks," Scientific American, vol. 288, no. 5, May 2003, pp. 60-69.

[4] R.C. Barros, M.P. Basgalupp, D.D. Ruiz, A.C.P.L.F. de Carvalho, and A.A Freitas, "Evolutionary model tree induction," Proc. ACM Symposium on Applied Computing (SAC '10), 2010, pp. 1131-1137.

[5] Ö. Bodin and Maria Tengö, "Disentangling intangible social-ecological systems," Global Environmental Change, vol. 22, no. 2, May 2012, pp. 430-439.

[6] J. Bohn, V. Coroamă, M. Langheinrich, F. Mattern, and M. Rohs, "Social, economic, and ethical implications of ambient intelligence and ubiquitous computing," in Ambient Intelligence, W. Weber, J.M. Rabaey, and E. Aarts, Eds. Springer Berlin Heidelberg, 2005, pp. 5-29.

[7] S.R. Carpenter, C. Folke, M. Scheffer, and F. Westley, "Resilience: accounting for the noncomputable," Ecology and Society, vol. 14, no. 1, article 13, 2009. Available online from: http://www.ecologyandsociety.org/vol14/iss1/art13/ 2015.03.05

[8] S.R. Carpenter et al., "General resilience to cope with extreme events," Sustainability, vol. 4, 2012, pp. 3248-3259.

[9] J. Casti, X-Events: The Collapse of Everything. New, York, NY: HarperCollins Publishers, 2012.

[10] L. Chelleri, J.J. Waters, M. Olazabal, and G. Minucci, "Resilience trade-offs: addressing multiple scales and temporal aspects of urban resilience," Environment & Urbanization, 2015, pp. 1-18. Available online from: http://eau.sagepub.com/content/early/2015/01/09/0956247814550780.full.pdf+html 2015.03.05

[11] R.S. Cohen, "How useful is the complexity paradigm without quantifiable data? A test case: the patronage of 5th-6th century Buddhist caves in India," in Chaos and Society (Frontiers in Artificial Intelligence and Applications), A. Albert, Ed. Amsterdam, The Netherlands: IOS Press, 1995, pp. 83-99.

[12] J.P. Crutchfield, "The hidden fragility of complex systems – consequences of change, changing consequences," in Cultures of Change: Social Atoms and Electroniuc Lives, G. Ascione, C. Massip, J. Perello, Eds. Barcelona, Spain: ACTAR D Publishers, 2009, pp. 98-111.

[13] J. Diamond, Collapse: How Societies Choose to Fail or Survive. England: Penguin Publishing Group, 2011.

[14] C. Folke et al., "Resilience thinking: integrating resilience, adaptability, and transformability," Ecology and Society, vol. 15, no. 4, article 20, 2010. Available online from: http://www.ecologyandsociety.org/vol15/iss4/art20/ 2015.03.05

[15] T.B. Fowler and M.J. Fischer, Eds., Rare Events: Can We Model the Unforeseen? Sigma, vol. 10, no. 1. Noblis, September 2010. Available online from:

http://www.noblis.org/noblis-media/20f758e0-b3b9-4b76-8b81-4cde0d7341f9 2015.03.05

[16] C. Gershenson, "Updating schemes in random Boolean networks: do they really matter?" Proc. Ninth International Conference on Simulation and Synthesis of Living Systems (Artificial Life IX). MIT Press, 2004, pp. 238-243.

[17] C. Gershenson, "Guiding the self-organization of random Boolean networks," Theory in Biosciences, vol. 131, no. 3, 30 Nov 2011, pp. 181-191.

[18] M. Göndör and V.P. Bresfelean, “REPTree and M5P for measuring fiscal policy influences on the Romanian capital market during 2003-2010,” International Journal of Mathematics and Computers in Simulation, vol. 6, no. 4, pp. 378-386, 2012.

[19] I. Harvey and T. Bossomaier, "Time out of joint: attractors in asynchronous random Boolean networks," Proc. Fourth European Conference on Artificial Life, 1997, pp. 67-75.

[20] K.A. Hawick, H.A. James, and C.J. Scogings, “Simulating large random Boolean networks,” Res. Lett. Inf. Math. Sci., vol. 11, 2007, pp. 33-43.

[21] C.S. Holling, "Understanding the complexity of economic, ecological, and social systems," Ecosystems, vol. 4, no. 5, 2001, pp. 390-405.

[22] S.A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets,” Journal of Theoretical Biology, vol. 22, no. 3, March 1969, pp. 437-467.

[23] S.A. Kauffman, “Antichaos and adaptation,” Scientific American, vol. 265, no. 2, August 1991, pp. 78-84.

[24] S.A. Kauffman, The Origins of Order: Self-Organization and Selection in Evolution. New York, NY: Oxford University Press, 1993.

[25] C.G. Langton, “Computation at the edge of chaos: Phase transitions and emergent computation,” Physica D, vol. 42, 1990, pp. 12-37.

[26] R. Legaspi, H. Maruyama, R. Nararatwong, and H. Okada, “Perception-based resilience: accounting for the impact of human perception on resilience thinking,” Proc. 2014 IEEE Fourth International Conference on Big Data and Cloud Computing, December 2014, pp. 547-554.

[27] T.G. Lewis, Bak’s Sand Pile: Strategies for a Catastrophic World. Williams, CA: Agile Press, 2011.

[28] J.T. Lizier, S. Pritam, and M. Prokopenko, "Information dynamics in small-world Boolean networks," Artificial Life, vol. 17, no. 4, Fall 2011, pp. 293-314.

[29] P.H. Longstaff, T.G. Koslowski, and W. Geoghegan, “Translating resilience: a framework to enhance communication and implementation,” Proc. 5th International Symposium on Resilience Engineering, June 2015.

[30] A.M. Machado and A.L.C. Bazzan, "Self-adaptation in a network of social drivers: using random Boolean networks," Proc. Workshop on Organic Computing, 2011, pp. 33-40.

[31] P. Martin-Breen and J.M. Anderies, “Resilience: a literature review,” The Rockefeller Foundation, September 18, 2011. Available online from: http://www.rockefellerfoundation.org/blog/resilience-literature-review 2015.03.05

[32] H. Maruyama, R. Legaspi, K. Minami, and Y. Yamagata, “General resilience: taxonomy and strategies,” Proc. IEEE 2014 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE), IEEE Press, March 19-21 2014, pp. 1-8, ISBN: 978-1-4799-2628-2.

[33] B. McKelvey, "Towards a Campbellian realist organization science," in Variations in Organization Science: In Honor of Donald T. Campbell, J.A.C. Baum and B. McKelvey, Eds., Thousand Oaks, Calif: SAGE Publications, 1999, pp. 383-411.

[34] B. McKelvey, "Self-Organization, complexity catastrophe, and microstate models at the edge of chaos," in Variations in Organization Science: In Honor of Donald T. Campbell, J.A.C. Baum and B. McKelvey, Eds., Thousand Oaks, Calif: SAGE Publications, 1999, pp. 279-307.

[35] D. Meadows, J. Randers, and D. Meadows, Limits to Growth: The 30-Year Update. White River Junction, VT: Chelsea Green Publishing Company, 2004.

[36] M. Mitchell, P.T. Hraber, P.T., and J.P. Crutchfield, “Revisiting the edge of chaos: evolving cellular automata to perform computations,” Complex Systems, vol. 7, no. 2, 1993, pp. 89-130.

[37] F. Morrison, The Art of Modeling Dynamic Systems: Forecasting for Chaos, Randomness and Determinism. Mineola, NY: Dover Publications, 2008.

[38] C. Müssel, M. Hopfensitz, and H.A. Kestler, BoolNet package vignette, 2014. Available online from: http://cran.r-project.org/web/packages/BoolNet/vignettes/BoolNet_package_vignette.pdf 2015.03.05

[39] N.H. Packard, “Adaptation toward the edge of chaos,” in Dynamic Patterns in Complex Systems, J.A.S. Kelso, A.J. Mandell, M.F. Shlesinger, Eds. Singapore: World Scientific, 1988, pp. 293-301.

[40] S.E. Page, Diversity and Complexity. Princeton, NJ: Princeton University Press, 2011.

[41] G. Paperin, D.G. Green, and S. Sadedin, "Dual-phase evolution in complex adaptive systems," Journal of the The Royal Society Interface, vol. 8, no. 58, 2011, pp. 609-629.

[42] C. Perrow, Normal Accidents: Living with High Risk Technologies. Princeton, NJ: Princeton Univ Press, 1999.

[43] S. Poslad, Ubiquitous Computing: Smart Devices, Environments and Interactions. Wiley, 2009.

[44] M. Scheffer et al., "Anticipating critical transitions," Science, vol. 338, no. 6105, 19 October 2012, pp. 334-348.

[45] R.V. Solé, B. Luque, and S. Kauffman, “Phase transitions in random networks with multiple states,” Technical Report 00-02-011, Santa Fe Institute, 2000.

[46] J.A. Tainter, The Collapse of Complex Societies. Cambridge, UK: Cambridge University Press, 1988.

[47] N.N. Taleb, The Black Swan – The Impact of the Highly Improbable. New York: Random House, 2007.

[48] T. Taylor, "Exploring the concept of open-ended evolution," Proc.Thirteenth International Conference on the Simulation and Synthesis of Living Systems (Artificial Life 13), C. Adami, D.M. Bryson, C. Ofria, and R.T. Pennock, Eds. MIT Press, 2012, pp. 540-541.

[49] E. Thelen, "Dynamic systems theory and the complexity of change," Psychoanalytic Dialogues vol. 15, no. 2, 2005, pp. 225-283.

[50] R. Valerdi et al., “A research agenda for systems of systems architecting,” Int. J. of System of Systems Engineering, vol. 1, no. 1/2, 2008, pp. 171-188.

[51] Weka 3: Data Mining Software in Java. Available online from: http://www.cs.waikato.ac.nz/ml/weka/index.html 2015.03.05

[52] T. Zhou, B.-H. Wang, P.-L. Zhou, C.-X. Yang, J. Liu, "Self-organized Boolean game on networks," Physical Review E, vol. 72, no. 046139, 28 October 2005, pp. 1–6.

[53] A. Zolli and A.M. Healy, Resilience: Why Things Bounce Back. New York, NY: Free Press, July 2012.

[54] H. Atlan, F. Fogelman-Soulie, J. Salomon, and G. Weisbuch, “Random Boolean Networks,” Cybernetics and Systems: An International Journal, vol. 12, nos. 1-2, 1981, pp. 103-121.

Autonomic Duty Cycling for Target Tracking in a Bio-Inspired Wireless Sensor Network

Camila H. S. Oliveira and Miguel F. de Castro
Computer Science Dept. / GREat Research Lab / Federal University of Ceará
Email: {camila, miguel}@great.ufc.br

***Abstract*—The evolution of new miniatured devices, increasingly cheaper and more efficient, is enabling the use of Wireless Sensor Networks (WSN) for different application scenarios, such as the Internet of Things. However, even when nodes are deployed with a high degree of redundancy, common in target tracking applications, energy management is still a challenge for research. In order to optimize energy consumption in this dense network scenario, duty cycling mechanisms arise as an efficient solution to improve and maximize the WSN lifetime. In this paper, a new autonomic duty cycling mechanism called BioSched is proposed, aiming at simplicity, robustness and scalability. The mechanism is implemented over Biologically-inspired Optimization for Sensor network Lifetime (BiO4SeL), an autonomic routing protocol based on Ant Colony heuristics designed to optimize WSN lifetime. Based on the routes found by the ants in BiO4SeL, BioSched puts a subset of nodes with less residual energy to sleep while ensuring good delivery rate. Results show that the proposed duty scheduling enhances BiO4SeL in network energy saving and lifetime, and also outruns a related duty cycling algorithm called EC-CKN (Energy Consumed uniformly-Connected K-Neighborhood).**

***Keywords*—Duty Cycling, WSNs, Tracking.**

I. Introduction

WSNs are networks typically made up of thousands of sensor devices able to monitor several types of phenomena and transmit relevant harvested information to one or more base stations. There are numerous application fields to WSN, such as healthcare, security, agriculture, military, smart homes, etc. One of the classic applications of WSN is target tracking, in which the main objective is to search and track a given target within a given area of interest, possibly remote or hostile [1]. Wild life monitoring is an example of its possible scenarios. In general, this application requires sensors closest to the target to keep sending data to the sink, while other nodes sense for a possible target movement, not necessarily sending data to the sink.

For this specific application, the first challenge is to have an efficient routing algorithm in order to enable data harvesting at the sink node. This algorithm must be autonomic and robust, since nodes can be deployed in a field where human intervention is unviable. For the same reason, the routing algorithm must be energy efficient, as nodes' batteries replacement can also be unviable. Intending to overcome these challenges, Castro *et al.* [2] proposed BiO4SeL. BiO4SeL is an autonomic protocol that performs route discovery and maintenance using ants in order to optimize network lifetime. In BiO4SeL, the ants work in a distributed and autonomic way in order to find paths between source and destination. The best paths – not necessarily the shorter ones – are traced through by the ants depositing pheromone at each node in the path, just like biological ants. The pheromone mechanism used by the ants makes it easy to adapt to topology changes and eliminates the need of localization and global information. BiO4SeL uses the residual energy data on a given node in order to deposit and to evaporate the pheromone. The less energy a hop has, the faster its pheromone will dissipate. This will lead to a lower probability of this node to be chosen as a packet relay. Using this parameter along with the probabilistic factor involved on the route decision, the protocol distributes energy consumption among all nodes in the network while optimizing the network lifetime through the decrease of energy variance between the nodes.

BiO4SeL was designed considering a traffic behavior compatible with target tracking application, *i.e.*, one node transmits information to the sink at a given moment (typically the one closest to the target). As shown in [2], BiO4SeL presents good performance when compared to other related protocols: Ad-Hoc On-Demand Distance Vector (AODV) [3], Ant-based Routing Algorithm for Manets (ARAMA) [4], and Energy Aware Routing for Low Energy Ad Hoc Sensor Networks (EAR) [5], and meets its main objective of optimizing the network's lifetime, among others.

Another possible approach that can be associated with BiO4SeL in order to improve network lifetime is to deploy more nodes than necessary to cover the surveilled area (deploy dense network), relying on a duty cycle for these nodes. The work described in this paper takes advantage of the pheromone-based routing scheme provided by BiO4SeL in order to propose a new duty cycling scheme called BioSched. It was designed with two main objectives: (1) to overcome the energy consumption distribution problem of BiO4SeL, and (2) to adapt BiO4SeL to optimize network lifetime when operating in dense networks.

The rest of this paper is organized as follows. In Section II, some related works are discussed. Section III presents an overview of the BiO4SeL protocol. Section IV provides a detailed description of the proposed scheduling mechanism. Section V shows the experiments based on simulations and their numerical results, and Section VI concludes this paper.

II. Related Works

There are different approaches in literature aiming at optimizing network lifetime, such as topology control [6] [7], data aggregation/fusion [8] [9] and routing [10] [11]. In dense networks, node duty cycling is a solution that shows good results. The main idea is to take advantage of the large (more than necessary for full coverage) number of nodes deployed in the environment and coordinate the node operation mode, making them relay their duty cycle in order to save energy. However, the decision about which nodes should change their operation mode needs to take into account network connectivity and coverage requirements [12]. Recently, some techniques have been used to perform node activity scheduling. Some of them deal with the network connectivity problem, others aim at the coverage problem and only very few of them deal with both requirements. Most works that try to ensure 100% coverage and/or connectivity, required for target tracking applications, employ complex methods or have strong assumptions such as network global knowledge or localization awareness.

In [13], a solution based on a graph theory technique called minimum dominating sets was employed to eliminate nodes redundancy and leave in active operation mode a minimum number of nodes needed to ensure network coverage. In theory, these sophisticated methods achieves good results. However, in real WSN, the cost to compute those solutions can be unaffordable. Besides the complexity, the work that uses that kind of technique is in most cases centralized and based on location information. Others examples using complex solutions can be seen in [14] [15].

In [16] [17] [18], some solutions based on routing are presented. In general, the scheduling mechanism integrated in the routing procedure are computationally simpler and do not overload the network. However, it is hard to find works that operate in a simple, autonomous and distributed way. For example, in [17], the nodes have no autonomy to decide when to change their operation mode, and the solution is centralized in a coordinator node.

In [19], authors propose a distributed scheduling algorithm in which nodes are supposed to collaborate to keep a minimum amount of nodes active in order to maintain k-coverage, where k is a parameter. However, the algorithm is based on relatively heavy processing, and it could become an issue when scenario is dynamic.

Another approach by Yuan *et al.* [20], named EC-CKN, selects nodes to sleep based on remaining energy and connectivity, which is a similar approach to ours. The duty cycling is performed in fixed-sized rounds. In each round, the nodes communicate with each others in order to decide which ones will switch to sleep mode. Hence, the decision is not taken by each node autonomously, which can compromise the adaptability in dynamic scenarios.

In order to try to solve the aforementioned deficiencies, this paper proposes a duty cycling mechanism called BioSched based on the routing process provided by the BiO4SeL protocol. The mechanism proposed works in an autonomous and distributed way, such as BiO4SeL, and gives autonomy to nodes to decide when to change its operation mode using local information only (already acquired by BiO4SeL from neighbor nodes), thus providing a simpler solution. In order to evaluate the relative performance of our approach, we compare it with regular BiO4SeL and also with EC-CKN [20].

III. BiO4SeL Protocol

BiO4SeL is a protocol that incorporates the autonomous and distributed behavior of ants and employs a heuristic solution based on the residual energy of nodes to control the pheromone deposition along the paths. The protocol assumes that the nodes do not have prior network knowledge, not even the base station location. It uses ants to set up multiple paths between source and base station, and implements an inverse probabilistic routing table in each node in order to calculate the next hop in the path on the basis of just the node residual energy and the amount of pheromone cumulated in the hop. More details on BiO4SeL can be found in [2].

BiO4SeL operation comprises three phases: Bootstrap, Initial Route Discovery and Data Forwarding. At the Bootstrap phase, each node sends an *iHello* message by broadcast to its neighbors, which is used to initialize the node route tables with neighbors information, including energy.

With all nodes aware of their neighborhood, the protocol initiate the Initial Route Discovery phase. At this phase, the path discovery is done. At the end of this phase, there are paths established between all nodes and the base station.

After that, the Data Forwarding phase takes place. In this phase, data is forwarded from the source node to the base station and announcement ants (*hello*) are broadcasted at regular time interval in order to update the neighborhood with the node status. When a node receives a *hello* ant from a neighbor, the node updates the neighbor residual energy and the expiry time to the neighbor is restarted. At the beginning of the Data Forwarding phase, the best path is the shortest one. However, over time, the procedure of pheromone deposition and evaporation changes the priority to paths where the nodes have more residual energy. In order to encourage exploration of new paths, a probability (*pE*) is used and has its value calculated on the basis of nodes's residual energy.

However, this pheromone update process can concentrate pheromones in a few nodes preventing the choice of different routes. In order to avoid that and to encourage better distribution of energy consumption, BiO4SeL also implements an evaporation procedure. Evaporation takes place when the same hop is chosen several times. As the mentioned solution to decrease the concentration of energy waste takes some time to be effective, the BiO4SeL shows that energy waste still concentrates in the nodes surrounding the shortest paths. Aiming to solve this problem and to increase the network lifetime, the BioSched v1 and v2 presented in the next section show a method to save energy in the nodes most frequently used and in the nodes that waste all their batteries by sending signaling messages.

IV. THE BIOSCHED ALGORITHM

Scheduling node activity results in unquestionable power saving. However, choosing which nodes must be switched to sleep mode may be a critical issue, since it can impact essential requirements for network operation, such as connectivity and coverage, which are crucial for target tracking applications. Aiming to give strict coverage and connectivity guarantees, some works in the literature present their solutions based on restrictive assumptions, what makes their activity impracticable in real life WSN. According to [21], all scheduling mechanisms must provide three basic features: simplicity, scalability and robustness. With these features, a solution can save energy without overloading the network.

In order to fulfill the requirements mentioned above, this paper presents BioSched. The main difference between BioSched and the mechanisms found in the literature is main objectives. BioSched is not interested in switching the maximum possible number of nodes to sleep mode. Instead, it is designed to save energy from the most requested nodes and switching them to sleep mode. Thus, the neighboring nodes can be included in routing and the energy consumption is balanced.

The BioSched proposal includes two versions: the BioSched v1 and BioSched v2. The first one implements a heuristic method that takes into account energy and the workload of each node. In this case, the nodes with a lower battery and higher workload tend to be put to sleep. The second one, in addition to the nodes with high workload, also put to sleep the nodes that have never been required to forward packets. The main feature of BioSched is that the node itself decides whether to be switched to sleep node or not, ensuring that the mechanism is autonomous and simple.

A. BioSched v1

The BioSched v1 was fully integrated into BiO4SeL in order to take advantage of the protocol and to avoid network overload. In order to perform the scheduling activity, BioSched defines three possible states of a node: *sleeping*, *active* and *thinking*. When the nodes are in the *active* and *thinking* states, they are able to perform sensoring and transmission functions. In the *sleeping* state, nodes do not perform any networking function. The state changes are controlled by their own nodes based on their workload. Besides, all information used to make decisions is based on the node local information gathered from BiO4SeL messages.

At the beginning of network operation, all nodes are in the *active* state. When BiO4SeL comes into its second phase, where the routing process starts, the nodes begin to observe their own workload and, based on that, they decide whether to change their state or not. When in the *active* state, if a node decides to change to another state, it only switches to *thinking* state. Once the node is in the *thinking* state, it has two options: go back to *active* state or go to *sleeping* state. This decision is based on the node's workload and on the estimation of how this change can affect network connectivity. The state diagram with the change conditions is shown in Figure 1.

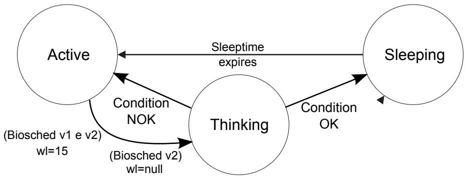


Fig. 1. BioSched state diagram.

The node workload (*wl*) is defined as frequency in that the node is chosen to forward a data packet. Until the source starts sending data to the destination, all nodes have their *wl* set to 0. When the nodes are chosen to transmit data packets, each time it is chosen its *wl* is increased in one unit. The *wl* is the variable observed by the node to trigger the node change of state to *thinking* mode. The state change is triggered off when the *wl* reaches the threshold defined as 15. The *wl* threshold and all the other values set to the variables used in this work are empirically defined by simulation experiments.

Once the node is in the *thinking* state, it has to decide whether it goes to *sleeping* state or to *active* state. The decision is made using only information received from neighbors through the BiO4SeL messages. The node is able to go to sleep when it obeys the set of conditions defined to estimate if the node's state change results in connectivity and coverage problems or not. The following conditions have to be fulfilled to allow nodes to switch to sleep mode: (1) At least ten neighbors in *active* state (*MinNb*); (2) At least one among these ten neighbor nodes fulfills the following conditions: (a) Not being source, destination or previous node in the path; (b) Being closer to base station than the previous node in the path; (c) Being the neighbor of the previous node in the path.

The previous node in this context is the node that sent the last data packet to the node that is in the *thinking* state ($node_{thinking}$). When the $node_{thinking}$ receives the data packet directly from the source, the source is also the previous node. After verifying the conditions, if the $node_{thinking}$ decides to go to *sleeping* state, the node has to calculate how much time it will stay there. After calculating the sleep time, the node's workload (*wl*) is reset and a new count will be started when the node comes back to active state. When all these steps are followed, the node broadcasts an update message to its neighbors and goes to *sleeping* state. At the moment the node wakes up, it goes to the *active* state and does not send any update message. The state of the nodes is updated through the *Hello* message used by BiO4SeL to keep neighbor tables updated. The node calculates the sleep time according to the following equation:

$$T = (\frac{avg\_en}{energy}) \times sp \tag{1}$$

where *T* is the sleep time, *avg_en* is the energy average of all node neighbors, *energy* is the current node energy and *sp* is a constant used as a factor to control the sleep time assigned to the nodes. As the objective of BioSched is to increase the network lifetime, the lower the node's energy compared to its

TABLE I
COVERAGE ENSURED WITH DIFFERENT NEIGHBOR QUANTITY [22].

# of neighbors	% of coverage	# of neighbors	% of coverage
5	91.62%	9	98.85%
7	96.89%	11	99.57%

neighbors' energy, the greater its sleep time. Therefore, the sleep time is directly proportional to the ratio between the the energy average of all node neighbors and the current node's energy.

The constant *sp* was set to 20s after several simulations varying this value between 1s and 30s. The results showed that when simulation set *sp* to 1s, the network get overhead due to the large amount of update message. Thus, a too short *sp* has as a main consequence packets loss and energy waste. On the other hand, when the *sp* is set to 30s, the distribution of energy waste between the nodes is compromised, which may result in energy depletion of just one node, decreasing significantly the network lifetime. This happens when a node is in the *thinking* state, sometimes all node neighbors that fulfill the requirements described above are sleeping. As its neighbors may sleep for a long time, the node energy can run out before it has the opportunity to change its state. The best results were obtained setting *sp* to 20s. This simulation shows that an average time is better because it is enough to save node energy without concentrating energy consumption in the active nodes.

In the case where the node is in the *thinking* state is not able to go to the *sleeping* state, it just comes back to *active* state while keeping its workload. From the first time on, every time the node's workload increases, it goes to *thinking* state to try the chance of going to *sleeping* state.

BioSched v1 ensures network coverage according to data found in [22], where an analytical model was defined and employed to determine the redundant node amount according to a percentage required by the application. Table I presented in [22] defines that if a node has more than 4 neighbors, then it has more than 90% of its range of transmission covered by its neighbors. As the conditions in BioSched is limited to 10 (the value of *MinNb*) it guarantees 98.86% of coverage. However, connectivity is not 100% guaranteed because there is the possibility of losing the state update message. However, results show that BioSched v1 can ensure a high delivery rate.

B. BioSched v2

In BioSched v2, as well as in BioSched v1, the nodes also have the autonomy to decide when to change their states. In this extension, the nodes that have a nill workload also have the opportunity of changing their state. As in BioSched v1 only the overused nodes can go to a sleep state, most nodes in the network remain active while wasting energy to send and receive *hello* messages. BioSched v2 was proposed in order to save energy, by also preserving the less used nodes in routing process.

All states, constant and messages defined in BiO4SeL v2 are also valid in BiO4SeL v3. Indeed, BiO4SeL v3 uses the message *Hello* in order to implement its extension. In this case, each time before sending a *Hello*, a node changes to the *thinking* state and checks if it has its *contPktDados* variable equal zero and at least ten active neighbors. Then, if the two conditions are met, instead of sending a *Hello*, the node sends a *UpdateStatus* message and goes to *sleeping* state. Otherwise, it sends a *Hello* and comes back to the *active* state. The procedure followed to calculate the sleeptime and wake up the nodes are the same used in BiO4SeL v2.

V. SIMULATIONS AND RESULTS

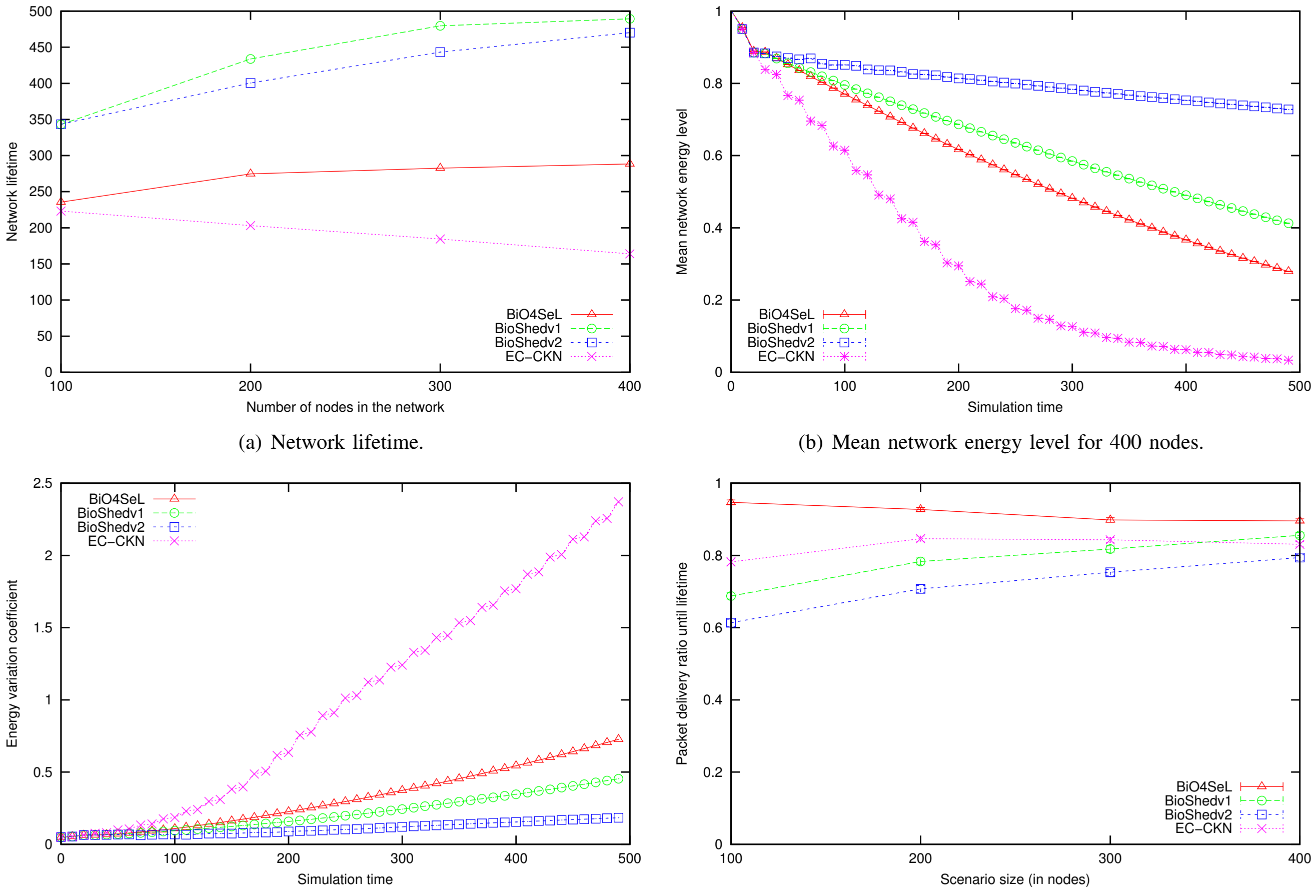
This section describes the experiments carried out using Network Simulator NS-2. As the objective of this work is to perform activity scheduling in dense WSN, the scenarios simulated vary in the number of nodes and in network density. The definition of density employed in this work was found in [23] and claims that a network is dense when a node has more than 10 neighbors. The density level varies from little dense to very dense. As simulations are interested in analyzing the mechanism according to network density, the scenario size was fixed in 50mx50m. Table II shows the simulation parameters and their values, where the density is given by the mean amount of neighbors per node.

TABLE II
SIMULATED SCENARIOS

# of Nodes	Density	# of Nodes	Density
100	20	300	60
200	40	400	80

Our solution was implemented using C++ and performed using IEEE 802.15.4 as underlying PHY and MAC layers, with most of the parameters set to default values. We used IEEE 802.15.4 because it is the standard protocol for simulation of WSN in NS-2. The simulation considers a homogeneous and static network with one source and one destination. The positions of the source and destination have been previously assigned. Assuming the scenario size of $50m^2$ fixed to simulation, the source was placed at position (5,25) and the destination at (45,25). Related to the tests, for each number of node shown in Table II, 30 different scenarios were generated. For each scenario, the simulation was repeated 30 times, changing the randomization seeds.

In [2], the total simulation time was set to 200 time units. However, in order to observe energy consumption behavior and the increase in the network lifetime, in this work, the simulation time was increased to 500 time units. The rate of data generation was kept unchanged from [2], sending one 76 KB packet each 0.4 time units. The results presented in this section include a confidence interval considering α = 0.95 and they aim to show the result of an autonomic energy management performed by BioSched v1 and v2, comparing results with original BiO4SeL and EC-CKN. These experiments are shown in the next subsections.



(a) Network lifetime.

(b) Mean network energy level for 400 nodes.

(c) Energy coefficient of variation for 400 nodes

(d) Packet delivery ratio before lifetime

Fig. 2. Simulation results.

A. *Network lifetime*

This experiment evaluated network lifetime, defined as the time in which the first node runs out its battery. Figure 2(a) shows that BioSched v1 and v2 provide an improvement in the network lifetime when compared to basic BiO4SeL and EC-CKN. Furthermore, the experiment suggests that the improvement in lifetime increases as the network density increases.

B. *Network mean battery level*

This experiment evaluated energy consumption throughout the simulation time with a scenario size of 400 nodes. As shown in Figure 2(b), at the beginning, until 30s, the mean energy consumption is the same for all protocols. That is explained by the fact that BioSched v1 and v2 are implemented over BiO4SeL, and EC-CKN also requires an initial stabilizing period. Thus, as the main modifications in BiO4SeL to integrate BioSched v1 and v2 were made in the Data Forwarding phase, the protocols have had the same initial phase behavior. As BioSched v2 puts a larger number of nodes to sleep, it substantially increases its mean battery level when compared to BioSched v1. EC-CKN produced the worst results, as it requires regular negotiation based on signaling messages, which consumes considerable energy. As in BioSched the node is autonomous to take its own decision, the energy expended in signaling is saved. Similar results ware observed with scenario sizes 100, 200 and 300 nodes.

C. *Network energy coefficient of variation*

In this experiment, the Coefficient of Variation (CoV) was used to evaluate how much, on average, node energy differs from the mean network energy level. The aim of this experiment was to show how energy load balancing is performed by BioSched. A lower value for the CoV means that the energy in the network is consumed more evenly among nodes. Figure 2(c) shows the CoV for scenario size of 400 nodes.

This figure shows that BioSched v1 and v2 improved battery consumption by balancing it among nodes, reducing the CoV. The same behavior occurred with other scenario sizes. This happened because even if BiO4SeL did not use all nodes in routing, it still used up the battery sending and receiving *Hello* messages, as the nodes do not sleep. Therefore, as the battery average was smaller and the energy consumption was more concentrated, the energy coefficient of variation tends to increase. As EC-CKN does not have the same routing balancing concern, the shortest path is used to route data packets and the energy consumption by this route is bigger, what increases CoV.

An important fact is that even if BioSched v2 did not perform the best battery consumption distribution, it presented the smaller CoV. This was a consequence of putting most nodes to sleep, thus, saving energy and keeping a higher battery average.

D. Packet delivery ratio until lifetime

The objective of this experiment was to evaluate BioSched v1 and v2 impact in the packet delivery ratio, *i.e.*, the ratio between the amount of packets received by the destination and sent by the source, in the interval $[0;1]$, when compared to basic BiO4SeL and EC-CKN.

Figure 2(d) shows that, in a very dense scenario, BiO4SeL decreases a little of its delivery ratio. BioSched v1 and v2, in their turn, increase their delivery ratio as the network density rises. That happens because as network density increases, the possibility of BioSched having connectivity problem decreases. As mentioned above, BioSched does not ensure 100% network connectivity, but the results show that, in the very dense network scenario, BioSched tends to work better than BiO4SeL to guarantee a higher delivery ratio. EC-CKN, in its turn, presents a very good delivery ratio before lifetime. However, as its lifetime is the smallest among all protocols, it evinces that the heavy use of the shortest path enhances delivery rate, but this is not a longstanding situation.

VI. Conclusions and Future Works

This paper proposed an autonomic mechanism for BiO4SeL protocol in order to enhance power optimization in dense WSN by means of activity cycling. Aiming to achieve the objective of energy saving, this work developed a simple, autonomous and distributed proposal called BioSched. In this algorithm, the decisions are based only on the node workload and on locally stored neighbor information. The algorithm performs node activity scheduling without generating overheads (other than the overheads already produced by the bio-inspired routing algorithm) in order to improve network lifetime. The tests were based on a traffic scenario compatible with target tracking application, where only one node produces data to be delivered to the sink at a given time.

The first contribution of BioSched v1 and v2 is their ability to perform node activity scheduling in a distributed and autonomous way. This feature, combined with the routing optimization provided by BiO4SeL, outperformed the basic BiO4SeL and also a related work called EC-CKN. Other characteristics of BioSched v1 and v2 are their simplicity, robustness and scalability. As the proposed mechanisms were fully integrated into BiO4SeL, they inherit its characteristics of robustness and scalability. In addition, the simplicity of the mechanisms consists in the way the nodes are chosen to switch to sleep mode. The fact that the nodes do not require information from all network nodes, and each node bases its decision only on its own workload, characterizes the simplicity of the proposed protocols. As a result of these two contributions, results obtained by simulations show that BioSched v1 and v2 can significantly increase the lifetime of the network and improve the distribution of energy consumption when compared to BiO4SeL and EC-CKN.

As future works, we intend to test these algorithms in more heterogeneous scenarios, where power storage, processing and transmission range are differently distributed among nodes in the network. We will also consider evaluating the impact of the introduction of some power harvesting nodes in the network.

References

[1] K. Ramya, K. P. Kumar, and V. S. Rao, “A survey on target tracking techniques in wireless sensor networks,” vol. 3, no. 4, Aug 2012, pp. 93–108.

[2] M. F. De Castro, L. B. Ribeiro, and C. H. S. Oliveira, “An autonomic bio-inspired algorithm for wireless sensor network self-organization and efficient routing,” *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 2003–15, Nov. 2012.

[3] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” Jul 2003, IETF RFC 3561.

[4] O. Hussein, T. Saadawi, and M. J. Lee, “Probability routing algorithm for mobile ad hoc networks’ resources management,” vol. 23, no. 12. Washington, DC, USA: IEEE Computer Society, 2005, pp. 2248–2259.

[5] R. Shah and J. Rabaey, “Energy aware routing for low energy ad hoc sensor networks,” in *IEEE Wireless Communications and Networking Conference (WCNC2002)*, vol. 1, Mar 2002, pp. 350 – 355 vol.1.

[6] J. Yu, E. Noel, and K. Tang, “Degree constrained topology control for very dense wireless sensor networks,” in *GLOBECOM 2010*, Dec 2010, pp. 1 –6.

[7] Z. Luqiao and Z. Qinxin, “Interference and energy aware topology control,” in *Int. Conf. on Consumer Electronics, Communications and Networks (CECNet 2011)*, Apr 2011, pp. 1357 –1359.

[8] A. Avokh and G. Mirjalily, “Dynamic balanced spanning tree (dbst) for data aggregation in wireless sensor networks,” in *Telecommunications (IST), 2010 5th International Symposium on*, Dec 2010, pp. 391 –396.

[9] L.-Y. Sun, W. Cai, and X.-X. Huang, “Data aggregation scheme using neural networks in wireless sensor networks,” in *Intl. Conf. on Future Computer and Communication (ICFCC 2010)*, vol. 1, May 2010, pp. V1–725 –V1–729.

[10] W. Guo, W. Zhang, and G. Lu, “A comprehensive routing protocol in wireless sensor network based on ant colony algorithm,” in *2nd. Intl. Conf. on Networks Security, Wireless Communications and Trusted Computing - Vol. 01*, ser. NSWCTC ’10, 2010, pp. 41–44.

[11] M. Paone, A. Cucinotta, A. L. Minnolo, L. Paladina, A. Puliafito, and A. Zaia, “A bio-inspired distributed routing protocol for wireless sensor networks: Performance evaluation,” in *IEEE 30th Intl. Conf. on Distributed Computing Systems Workshops*, ser. ICDCSW ’10, 2010, pp. 247–255.

[12] S. Mahfoudh and P. Minet, “Survey of energy efficient strategies in wireless ad hoc and sensor networks,” in *Networking, 2008. ICN 2008. Seventh Intl. Conf. on*, Apr 2008, pp. 1 –7.

[13] B. Pazand and A. Datta, “An energy-efficient node-scheduling scheme for wireless sensor networks based on minimum dominating sets,” vol. 19. New York, NY, USA: John Wiley & Sons, Inc., Feb 2009, pp. 75–99.

[14] J. Chen, J. Jia, Y. Wen, D. Zhao, and J. Liu, “Modeling and extending lifetime of wireless sensor networks using genetic algorithm,” in *1st. ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, ser. GEC ’09, 2009, pp. 47–54.

[15] Y. Lin, X.-m. Hu, and J. Zhang, “An ant-colony-system-based activity scheduling method for the lifetime maximization of heterogeneous wireless sensor networks,” in *12th Annual Conf. on Genetic and Evolutionary Computation*, ser. GECCO ’10, 2010, pp. 23–30.

[16] V. Vaidehi, U. Selvan, J. Jayendran, and K. Praveen, “Redundant node deactivation by scheduling in wireless sensor networks,” in *Recent Trends in Information Technology (ICRTIT), 2011 Intl. Conf. on*, Jun 2011, pp. 613 –617.

[17] R. Saravanakumar, S. Susila, and J. Raja, “An energy efficient cluster based node scheduling protocol for wireless sensor networks,” in *Solid-State and Integrated Circuit Technology (ICSICT), 2010 10th IEEE Intl. Conf. on*, Nov 2010, pp. 2053 –2057.

[18] A. Swain, R. Hansdah, and V. Chouhan, “An energy aware routing protocol with sleep scheduling for wireless sensor networks,” in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE Intl. Conf. on*, Apr 2010, pp. 933 –940.

[19] E. Bulut and I. Korpeoglu, “Sleep scheduling with expected common coverage in wireless sensor networks,” vol. 17, no. 1. Secaucus, NJ, USA: Springer-Verlag New York, Inc., Jan. 2011, pp. 19–40.

[20] Z. Yuan, L. Wang, L. Shu, T. Hara, and Z. Qin, “A balanced energy consumption sleep scheduling algorithm in wireless sensor networks,” in *Proc. of 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, 2011, pp. 831–835.

[21] I. Dietrich and F. Dressler, “On the lifetime of wireless sensor networks,” vol. 5. New York, NY, USA: ACM, Feb 2009, pp. 5:1–5:39.

[22] K. Wu, Y. Gao, F. Li, and Y. Xiao, “Lightweight deployment-aware scheduling for wireless sensor networks,” vol. 10. Hingham, MA, USA: Kluwer Academic Publishers, Dec 2005, pp. 837–852.

[23] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *Proc. of the 1st Intl. Conf. on Embedded networked sensor systems*, ser. SenSys ’03. New York, NY, USA: ACM, 2003, pp. 1–13.

Tubes and Metrics for Solving the Dilemma-Zone Problem

Leonard Petnga
Department of Civil and Environmental Engineering
University of Maryland, College Park, MD 20742, USA
Email: lpetnga@umd.edu

Mark A. Austin
Department of Civil and Environmental Engineering
and Institute for Systems Research
University of Maryland, College Park, MD 20742, USA
Email: austin@isr.umd.edu

***Abstract*—Our research is concerned with the modeling and design of semantically-enabled, efficient, safe and performant cyber-physical transportation systems (CPTS). As a class of cyber-physical systems (CPS), CPTS are characterized by a tight integration of software and physical processes for smartness, increased performance, safety and management of system functionality. We adopt this perspective in our investigation of solutions to the dilemma zone (DZ) problem, which currently claims thousands of lives every year at traffic intersections. In this paper, we define and introduce new "dilemma metrics" to solve this problem. Coupled with an innovative tubular (3D) characterization of the decision problem that arises at the onset of the yellow light, these metrics enable simple and actionable decision capabilities to deal with unsafe configurations of the system. We also set a pathway toward integrating dilemma metrics and dilemma tubes with an ontological framework – the latter encodes the reasoning platform supporting the broad implementation of the algorithmic solutions resolving unsafe configurations of CPTS, such as the ones created by the DZ problem.**

Keywords-Dilemma Zone; Metrics; Cyber-Physical Transportation Systems; Artificial Intelligence; Safety.

I. Introduction

During the past twenty years, transportation systems have been transformed by remarkable advances in sensing, computing, communications, and material technologies. The depth and breadth of these advances can be found in superior levels of automobile performance and new approaches to automobile design that are becoming increasing reliant on sensing, electronics, and computing. The trend toward "transportation smartness" is so pervasive that by next year, as much of 40% of an automobile's value will be embedded software and control related components [1][2]. And yet, despite an exponential increase in the number of software lines of code (SLOC) to achieve these benefits, accidents at traffic intersections claim around 2,000 lives annually within the US alone [3]. A key component of this safety problem is the dilemma zone (DZ), which is an area at a traffic intersection where drivers are indecisive on whether to stop or cross at the onset of a yellow light.

II. Project Scope and Objectives

Our research addresses challenges that are hindering the system-level development of cyber-physical transportation systems (CPTS). Challenges that remain to be overcome include: (1) the integration of cyber-physical systems (CPS) technologies into existing infrastructure, (2) the realization of "zero fatality" transportation systems, and (3) the development of formal models and credible, actionable performance and safety metrics [5]. To this end, metrics for system safety are needed to: (1) evaluate the operation and control of transportation systems in a consistently and systematic way (including situations such as the dilemma zone), (2) identify, measure and predict the effects of interconnectivity between systems components as well as system performance, and (3) set standards and serve as measure of effectiveness (MoEs) guiding model-based systems engineering (MBSE) efforts.

We consider in this project the interplay among the key players of transportation systems at traffic intersections, and the consequences of their interactions on overall traffic system level safety. This work-in-progress paper focuses on one aspect of the problem – development of metrics to capture the essence of these interactions, and support the characterization of the problem and its representation using three-dimensional dilemma tubes. Section III is a review of existing approaches to the dilemma zone problem and their limitations with regard to the current trend toward CPTS. Section IV introduces the new dilemma zone metrics and their tubular representation. Section V describes our plans for ongoing research.

III. Dilemma Zone Problem and Cyber-Physicality of Traffic Systems

Dilemma Zone: Definition and Existing Solution Approaches. Also called the twilight zone, Amber signal or decision zone, the dilemma zone is the area at a traffic intersection where drivers are indecisive on whether to stop or cross at the onset of a yellow light. The behavior of users in "twilight zones" is responsible for hundreds of lives lost and billions worth in damages at stop light intersections in the United States [3]. Scholars distinguish two types of dilemma zone that differ by the perspective adopted on the problem. Type I DZ is viewed from the "physics of the vehicle" as in [6] and [7] while Type II adopts the driver's perspective as reported in [8]. Both perspectives use the stop line as a reference for their measurement as shown on Figure 1. However, the boundaries of DZ of type II are sometimes measured with a temporal tag (i.e., representing the duration to the stop line) added to a probabilistic estimate [9]. In this work, the dilemma zone will be considered in the sense defined by Type I.

Past research has focused on finding ways to protect

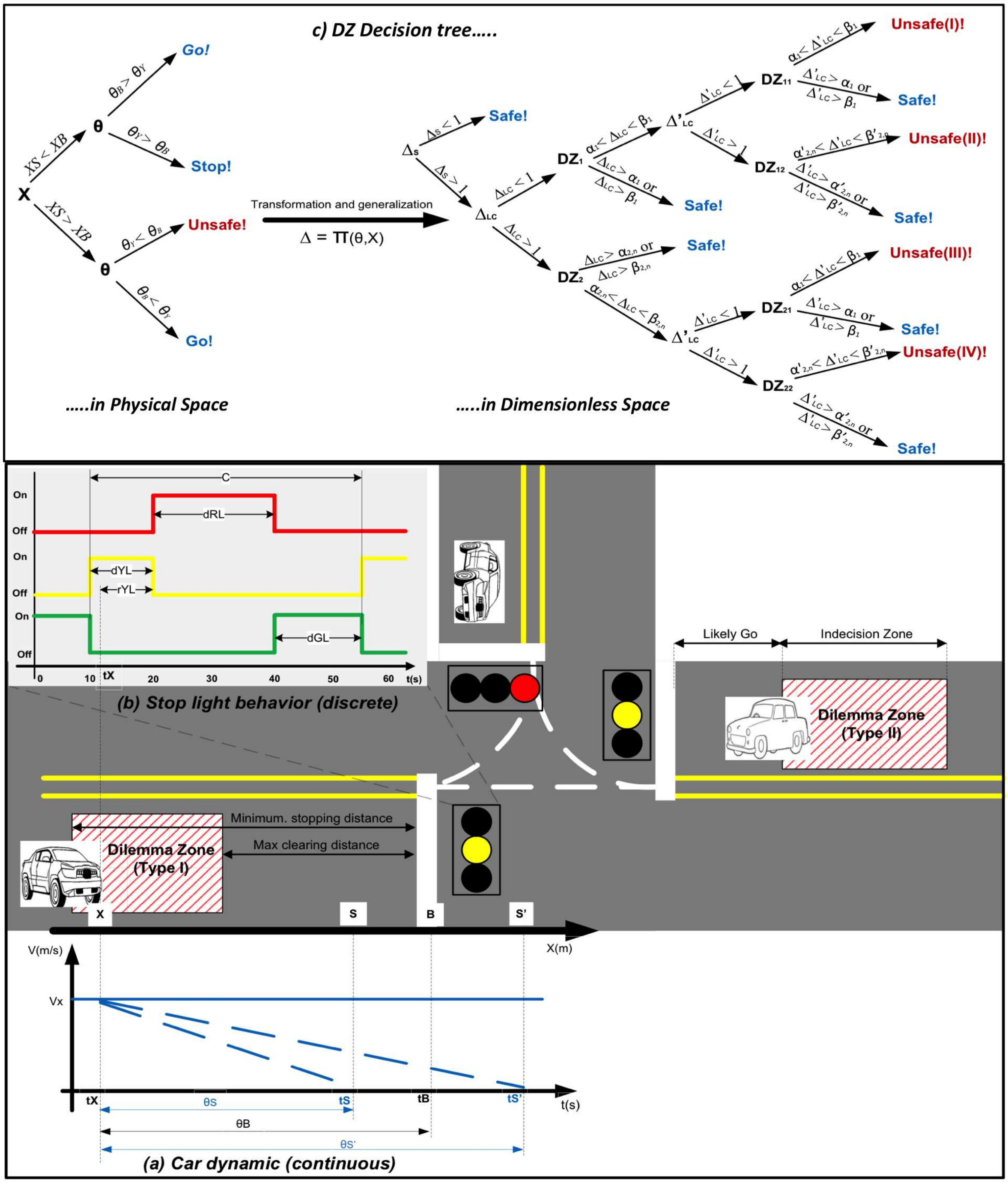

c) DZ Decision tree…..
X
$XS < XB$
$XS > XB$
θ
θ
$\theta_B > \theta_Y$
Go!
$\theta_Y > \theta_B$
Stop!
$\theta_Y < \theta_B$
Unsafe!
$\theta_B < \theta_Y$
Go!
Transformation and generalization
$\Delta = \Pi(\theta, X)$
Δ_S
$\Delta_S < 1$
Safe!
$\Delta_S > 1$
Δ_{LC}
$\Delta_{LC} < 1$
$\Delta_{LC} > 1$
DZ_1
$\alpha_1 < \Delta_{LC} < \beta_1$
$\Delta_{LC} > \alpha_1$ or $\Delta_{LC} > \beta_1$
Safe!
Δ'_{LC}
$\Delta'_{LC} < 1$
$\Delta'_{LC} > 1$
DZ_{11}
$\alpha_1 < \Delta'_{LC} < \beta_1$
Unsafe(I)!
$\Delta'_{LC} > \alpha_1$ or $\Delta'_{LC} > \beta_1$
Safe!
DZ_{12}
$\alpha'_{2,n} < \Delta'_{LC} < \beta'_{2,n}$
Unsafe(II)!
$\Delta'_{LC} > \alpha'_{2,n}$ or $\Delta'_{LC} > \beta'_{2,n}$
Safe!
DZ_2
$\Delta_{LC} > \alpha_{2,n}$ or $\Delta_{LC} > \beta_{2,n}$
Safe!
$\alpha_{2,n} < \Delta_{LC} < \beta_{2,n}$
Δ'_{LC}
$\Delta'_{LC} < 1$
$\Delta'_{LC} > 1$
DZ_{21}
$\alpha_1 < \Delta'_{LC} < \beta_1$
Unsafe(III)!
$\Delta'_{LC} > \alpha_1$ or $\Delta'_{LC} > \beta_1$
Safe!
DZ_{22}
$\alpha'_{2,n} < \Delta'_{LC} < \beta'_{2,n}$
Unsafe(IV)!
$\Delta'_{LC} > \alpha'_{2,n}$ or $\Delta'_{LC} > \beta'_{2,n}$
Safe!
…..in Physical Space
…..in Dimensionless Space
C
On
Off
dRL
On
Off
dYL
rYL
On
Off
dGL
0
10
tX
20
30
40
50
60
t(s)
(b) Stop light behavior (discrete)
Likely Go
Indecision Zone
Dilemma Zone (Type II)
Minimum. stopping distance
Max clearing distance
Dilemma Zone (Type I)
X
S
B
S'
X(m)
V(m/s)
Vx
tX
θS
tS
tB
tS'
t(s)
θB
θS'
(a) Car dynamic (continuous)

Figure 1. Type I & Type II Dilemma Zones and Decision Trees.

from, or eliminate, DZs using mostly a pure traffic control engineering view of the problem. These efforts have resulted in signal timing adjustment solutions that ignore or can't properly account for the physics of vehicles or driver's behaviors [10][11][12]. In order to deal with uncertainties, other scholars have used stochastic approaches such as fuzzy set [6] and Markov chains [7]. For all of these traditional techniques, the baseline of the solution can be either reduced (explicitly or not) to a space or temporal-based dilemma zone, but not both.

Autonomous Cars and Intelligent Traffic Control Systems. Recent work, such as that found in [13] and [14] illustrates the switch of researcher's interest toward investigating solutions to the DZ problem that incorporate both the car physics and light timing, while also providing a pathway forward for vehicle-to-infrastructure (V2I) interactions and integration. These solutions will soon become a reality, in part, because of an increased use of artificial intelligence in automating the command and operation of both cars and traffic signals. For automobiles, many aspects of autonomy – from braking to cruise control and driving functions – are in advanced stages of experimentation. Finding ways to put smartness into vehicles has contributed to reduced fatalities on highways mostly in the developed world. Looking ahead, even more automation is coming with self-driving cars [15][16].

The addition of artificial intelligence to traffic signal controls now makes sense due to an ability to determine the position, speed and direction of vehicles, and adjust light cycling times in a coordinated way to make the intersection crossing more efficient. Researchers have been developing and testing various technologies with mixed results [17][18][19]. As a case in point, a pilot study conducted by Carnegie Mellon University, reports a 40% reduction of intersection waiting times, an estimated 26% decrease in travel time, and a projected 21% decrease of CO_2 emissions [19]. Tapping into the full potential of these intelligence capabilities is hard as: (1) most vehicles can't currently communicate with traffic light controllers, and (2) autonomous vehicles still struggle in operating safely in adversary weather conditions (heavy rain, snow covered roads, etc.) and changing environment (temporary traffic signals, potholes, human behaviors, etc.). We assume in this paper that these problems will be resolved by ongoing research activites.

Toward Cyber-Physical Traffic Management Systems. Real-time situational awareness (e.g., traffic, location, speed) and decision, combined with vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications and control are valid and effective pathways for a solution to both congestion and safety at intersections. As such, we fully adopt a CPS view of the traffic system with regard to the DZ problem.

The value of this perspective has already been demonstrated by Petnga and Austin [20]. Autonomous vehicles (i.e., the physical system) interact with the light (i.e., the cyber system) with the objective of maximizing traffic throughput, while ensuring vehicle crossings are safe at the intersection. Enhanced performance and safety at the intersection have been proven possible, thanks to the critical role of temporal semantics in improving system level decision making. Also, when bi-directional connections between the vehicle and light are possible, new relationships can be established to characterize their tight coupling – this, in turn, enables the various computers in the CPTS to exchange information, reason, and make informed decisions. These capabilities are critical for those cases where the vehicle physics is such that they can neither stop nor proceed without entering and occupying the intersection while the traffic light is red. Therefore, the development of metrics for the DZ problem will greatly benefit from (and enrich) this CPTS perspective.

IV. Metrics for Characterizing the Dilemma Zone Problem

Safety Requirements to Decision Trees and Dilemma Metrics. The core safety requirement of the system car-light that should prevail all the time at intersection can be expressed as follows. "No vehicle is allowed to cross the intersection when the light is red". This is a non-functional requirement, a hard constraint whose violation is the driving force behind accidents at intersections. As shown on Figure 1 a) and b), the continuous dynamic of the vehicle and discrete behavior of the light illustrate the very different nature of both entities. This complicates the ability of the system to satisfy the safety requirement at the onset or in the presence of the yellow light.

Understanding the mechanisms by which system-level safety is achieved or violated is critical in addressing the DZ challenge. Decision trees appear to be the most suitable analysis tool to explore the different possible paths the system could follow and identify safe and unsafe ones. The tree shown on the left-hand side of Figure 1 c) shows the decision tree of the autonomous car - in the physical space - when it knows the traffic lights critical parameters at the time the decision is made. Petnga and Austin [20][21] have shown that the probability of the car making the right decision is higher when it knows before hands the following: (1) Duration Θ_Y of the yellow light before it turns red; (2) Vehicle stopping distance XS, and (3) Travel duration Θ_B or distance to light XB. However, moving forward requires a deep understanding of the interrelationships between cross-cutting system parameters from the various domains (car, light, time, space) involved at meta level. Also, the ability of the system to efficiently reason about unsafe situations and propose a satisfactory way out is critical.

We argue that this complexity can be kept in check by casting the problem in dimensionless terms and setting up a transformation $\Delta = \Pi(\Theta, X)$ of the initial decision tree from the physical space to a dimensionless space. Expressing the system decision tree in dimensionless space as a result of the transformation Π necessitates the definition of intermediary variables and parameters. We begin by noting that the car will not always catch the onset of the yellow light; thus, what is really relevant for efficient decision making here is the time left before the stop light turns red. Using the remaining duration of the yellow light r_{YL}, its full duration d_{YL} and the ones of the green and red lights ie d_{GL} and d_{RL}, we define the duration of a stop light cycle C, reduced cycle C_{YL} and cycle index k. The short (α_1) and full (α_2) yellow light duration as well as the short (β_1) and full (β_2) stop light indexes are also defined. The details are as follows.

$$\begin{aligned}
C &= d_{YL} + d_{RL} + d_{GL} && (1)\\
C_{YL} &= r_{YL} + d_{RL} + d_{GL} && (2)\\
k &= \frac{C}{C_{YL}} && (3)\\
\alpha_1 &= \frac{r_{YL}}{C_{YL}} && (4)\\
\alpha_2 &= \frac{d_{YL}}{C_{YL}} && (5)\\
\beta_1 &= \frac{r_{YL} + d_{RL}}{C_{YL}} && (6)\\
\beta_2 &= \frac{d_{YL} + d_{RL}}{C_{YL}} && (7)
\end{aligned}$$

We add to the aforementioned physical variables the stopping duration $\Theta_B^{'}$ of the car – should it decide to stop – and define the **car stopping distance metric** Δ_S, the **light-car crossing time metric** Δ_{LC} and the **light-car stopping time**

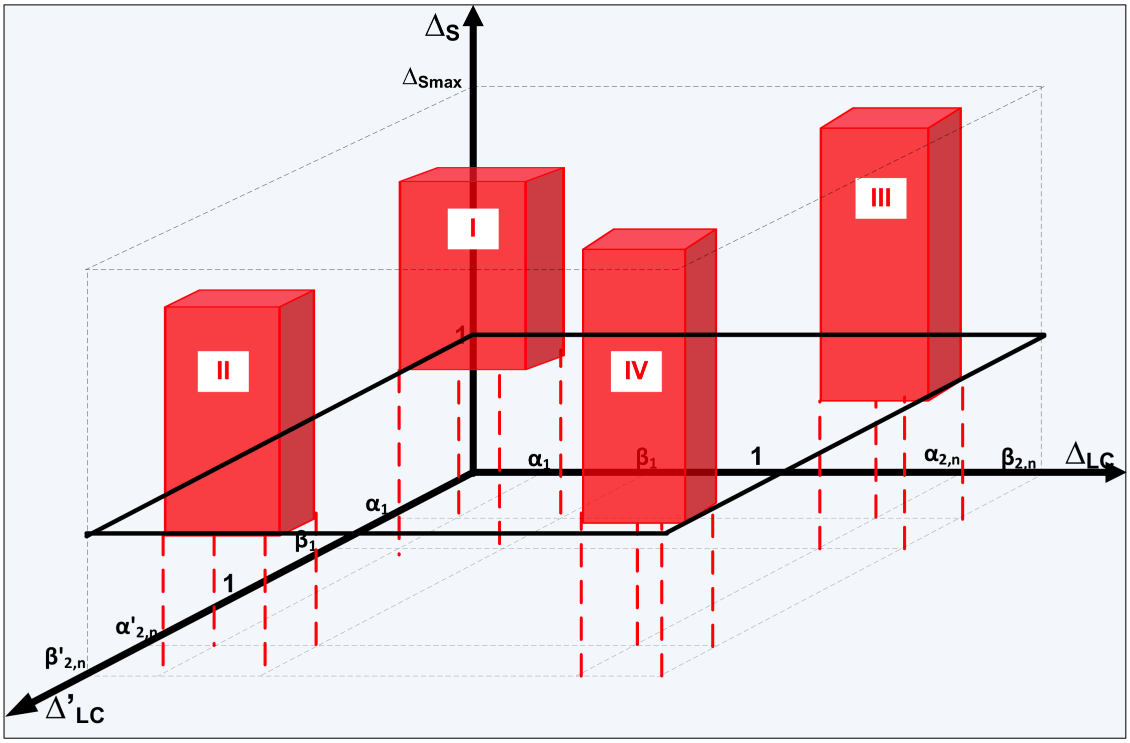


Figure 2. Dilemma Tubes in the Dimensionless (Δ) space.

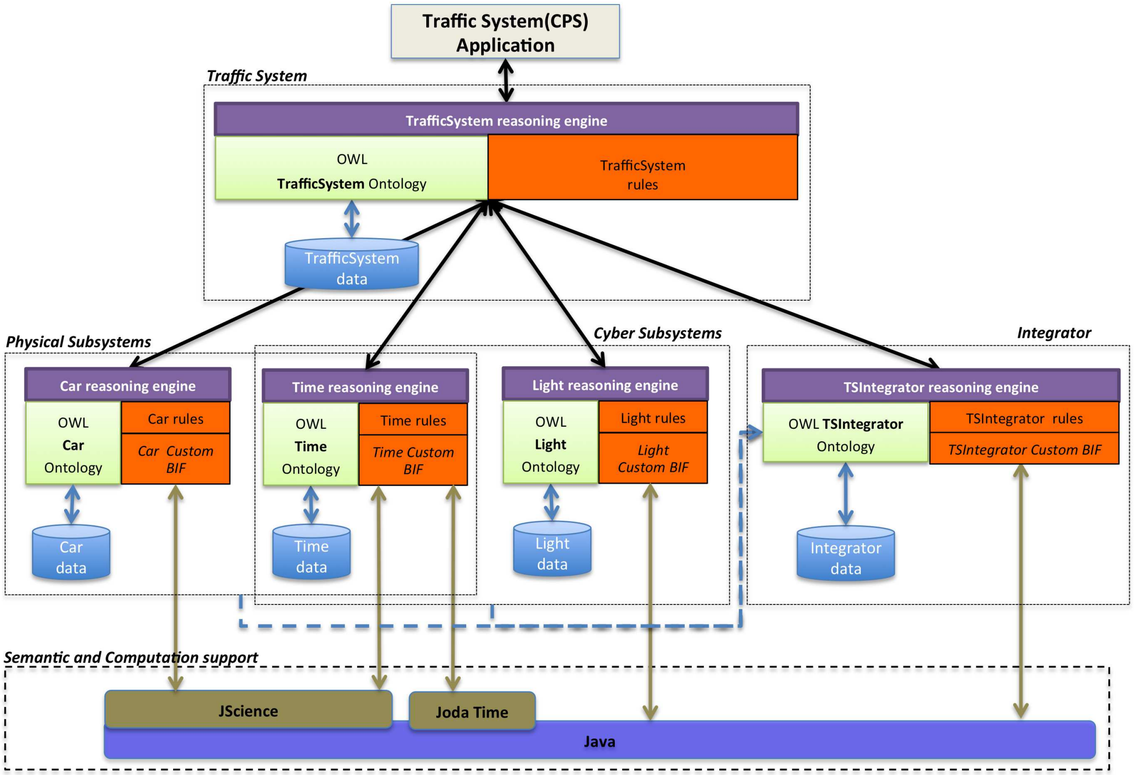


Figure 3. Architecture of the traffic system as a CPS.

metric $\Delta^{'}_{LC}$ as follows.

$$\Delta_S = \frac{XS}{XB} \quad (8)$$

$$\Delta_{LC} = \frac{\Theta_B}{C_{YL}} \quad (9)$$

$$\Delta^{'}_{LC} = \frac{\Theta^{'}_B}{C_{YL}} \quad (10)$$

All these metrics are dimensionless and serve as the key decision points of the dimensionless decision tree shown on the right-hand side of Figure 1 c). However, in order for us to be able to navigate the decision tree, we need additional information. We use the integer part function E to define n and $n^{'}$ indexes in (11) and (12) as follows.

$$n = E\left(\frac{\Delta_{LC} - 1}{k}\right) \quad (11)$$

$$n^{'} = E\left(\frac{\Delta^{'}_{LC} - 1}{k}\right) \quad (12)$$

They help specify the counterparts of α and β indexes when $\Delta_{LC} > 1$ or $\Delta^{'}_{LC} > 1$ as follows.

$$\alpha_{2,n} = k * \alpha_2 + k * n + 1 \quad (13)$$

$$\beta_{2,n} = k * \beta_2 + k * n + 1 \quad (14)$$

$$\alpha^{'}_{2,n} = k * \alpha_2 + k * n^{'} + 1 \quad (15)$$

$$\beta^{'}_{2,n} = k * \beta_2 + k * n^{'} + 1 \quad (16)$$

Along with (4) through (7), the values of α and β in (13) through (16) are necessary and sufficient to constraint the dimensionless metrics Δ_S, Δ_{LC} and $\Delta^{'}_{LC}$ and render a complete view of all possible outcomes of the decision tree in a dimensionless space Δ. Now, we can see that there are four possible configurations of the system for which it's unsafe as shown by the right-hand side of Figure 1 c).

From Dilemma Metrics to Dilemma Tubes. Each system unsafe configuration identified above corresponds to a "dilemma tube" in the Δ space as shown in Figure 2. For instance, (4), (6) and (8) through (10) provide the foundational elements for defining Tube I. However, in order to fully define the boundaries of each of the four tubes (i.e., I, II, III and IV), we add to the parameters introduced above, the maximum value of Δ_S ie Δ_{Smax} which is the maximum value of all the Δ_S for the system. Physically, it is determined by the physics of the family of vehicles crossing the intersection and the configuration of the traffic intersection as captured by (8). If at any instant the system is projected to enter an unsafe state, this situation will be materialized as a point coordinate $P_{\Delta}(\Delta_S,\ \Delta_{LC},\ \Delta^{'}_{LC})$ inside a particular tube. The physical interpretation of this phenomenon is that the autonomous car does not have a good decision option, and will need external help to safely cross the intersection. Scenarios that lead to unsafe system configurations will follow Unsafe branches of the decision tree on the right-and of Figure 1 c). While they won't necessary unfold in the order presented in the tree, the result will invariably be the same, i.e., the system will be projected to enter an unsafe state. In practice, the calculations can be done concurrently and the location of the resulting point coordinate relative to any of the four dilemma tubes easily determined. However, a vehicle can only be in one of the four dilemma tubes at a time - as they are mutually exclusive - or in any location in the remaining part of the Δ space i.e., a safe region.

Knowing in which tube the unsafe state has been materialized is critical in determining the appropriate course of action to prevent the occurrence of an accident.

V. FUTURE WORK

The key driver of our research is the modeling and design of semantically-enabled, efficient, safe and performant cyber-physical transportation systems. We are systematically working toward the platform infrastructure in Figure 3 (customized for the traffic system). The main aspects of this effort are as follows.

Topic 1. Architectural, ontological and reasoning infrastructures. The CPS perspective introduced above is translated into an ontological architecture where the subdomains involved in the transportation system are formally described at the appropriate level of detail. Thus, cyber, physical and meta domains (such as time and space) will be captured by description logic-enabled domain specific ontologies (DSO), each with its own rules engine. Spatio-temporal reasoning supported by appropriate implemented semantic extensions (such as Jscience or Joda time) will enhance traffic agents decision making capabilities. For the traffic system, the architectural framework will support reasoning in the dimensionless space and enable light reconfiguration, should a car be heading into a dilemma tube. The dilemma metrics introduced in this paper will be implemented in the Integrator rules engine. This entity (physically a smart traffic controller) will be the ultimate responsible of system level decisions. More details on the underlying semantic platform infrastructure supporting this architecture along with illustrative examples (ontologies, rules, extensions. etc.) can be found in [22].

Topic 2. Scripting language support for systems integration. Bringing together the various pieces of the above-mentioned architecture requires their bottom up integration in an organized but systematic way. Beside the necessary ontological integration of DSOs, we need a way to assemble system models. Our plans are to solve this problem with Whistle [23][24], a tiny scripting language where physical units are deeply embedded within the basic data types, matrices, branching and looping constructs, and method interfaces to external object-oriented software packages. Whistle is designed for rapid, high-level solutions to software problems, ease of use, and flexibility in gluing application components together. During the next iteration of development, Whistle will be extended to support co-simulation, graph databases, reasoning with ontologies and rules, and connections to external software packages through JFMI, the Java functional-mockup interface. Computational support will be added for input and output of model data from/to files in various formats (XML, Open Street Map (OSM), Java, etc.).

Topic 3. System modeling, simulation and performance evaluation. CPTS ontological modeling with the platform architecture of Figure 3 will provide insight into the reasoning structure needed to improve decision making at traffic intersections. It is especially important that computation and implementation of Allen's temporal logic and reconfiguration of the light behaviors are handled properly. We also need a component-based framework that is hooked to the ontological platform. The platform will be used for time-history simulations of traffic and light behaviors, and evaluation and visualization of the dilemma metrics and 3D dilemma tubes. Extensions of the platform to support the development and experimentation of V2V and V2I systems will be investigated.

VI. Conclusion

The purpose of this paper has been to describe a new and innovative tubular (3D) characterization of the dilemma zone problem, which enables quick and simple visual representation of the state of the traffic system. In traditional approaches to the DZ problem, cars and stoplights are treated separately. Our dilemma zone tubes result from a systems perspective where the cars and stoplights are treated as a whole. The second purpose of this paper has been to lay down the foundation for integrating these metrics and the tubular representation with an ontological framework for reasoning and decision making support to resolve unsafe configurations of the system. The next iteration of our work will include implementation and scripting of CPTS simulation scenarios with Whistle.

References

[1] J. Sztipanovits, J. A. Stankovic, and D. E. Cornan, Industry-Academy Collaboration in Cyber-Physical Systems(CPS) Research, *White Paper August 31*, White Paper, August 31, 2009.

[2] D. Winter, Cyber-Physical Systems in Aerospace - Challenges and Opportunities, *The Boeing Company*, Safe & Secure Systems & Software Symposium (S5), Beavercreek, Ohio USA, June 14-16, 2011.

[3] D. Hurwitz, The ”Twilight Zone” of Traffic costs lives at Stoplight Intersections, Oregon State University, Corvallis, Oregon, USA, March 03, 2012.

[4] D. Shrank, B. Eisele, and T. Lomax, TTIs 2012 Urban Mobility Report, *Texas A&M Transportation Institute, The Texas A&M University System*, Texas A&M Transportation Institute, The Texas A&M University System, December, 2012.

[5] Energetics Incorporated, Cyber-physical Systems Situation Analysis of Current Trends, Technologies, and Challenges, Energetics Incorporated for the National Institute of Standards and Technology (NIST), 2012.

[6] D. S. Hurwitz, B. H. Wang, M. A. Knodler, D. Ni, and D. Moore, Fuzzy Sets to Describe Driver Behavior in the Dilemma Zone of High-Speed Signalized Intersections, School of Civil and Construction Engineering, Oregon State University, USA and Department of Civil and Environmental Engineering, University of Massachusetts Amherst, USA, March, 2012.

[7] P. Li, Stochastic Methods for Dilemma Zone Protection at Signalized Intersections, Doctor of Philosophy Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University, VA, USA, August, 2009.

[8] C. V. Zeeger and R. C. Deen, Green-Extension Systems at High-Speed Intersections, Division of Research, Bureau of Highways, Department of Transportation, Commonwealth of Kentucky, April, 1978.

[9] M.S. Chang, C.J. Messer, and A. J. Santiago Timing Traffic Signal Change Intervals based on Driver Behavior, Transportation Research Record, 1027, pp. 20-30, 1985.

[10] P.D. Pant and Y. Cheng, Dilemma Zone Protection and Signal Coordination at Closely-Spaced High-Speed Intersections, Report FHWA/OH-2001/12, Ohio Department of Transportation, Columbus, OH, USA, November, 2001.

[11] D. Maas, Dilemma Zone Elimination, Sacramento Department of Transportation, Sacramento, CA, USA, 2008.

[12] K. Zimmerman and J.A. Bonneson, Number of vehicles in the dilemma zone as a potential measure of intersection safety at high-speed signalized intersections 83rd Annual Meeting of the Transportation Research Board Washington, D.C., USA, January, 2004.

[13] N.G. Wassim, J. Koopmann, J.D. Smith, and J. Brewer, Frequency of Target Crashes for IntelliDrive Safety Systems, US Department of Transportation – National Highway Transportation Safety Administration, DOT HS 811 381, October, 2010.

[14] J. Chu, At a Crossroads: New Research Predicts which cars are likeliest to Run Lights at Intersections, Accessible at : http://newsoffice.mit.edu/2011/driving-algorithm-1130; Retrieved: March 02, 2015.

[15] Google inc., The Latest Chapter for the Self-Driving Car: Mastering City Street Driving, Official Google Blog. N.p., n.d. Web. December, 20 2014, Accessible at: http://googleblog.blogspot.com/2014/04/the-latest-chapter-for-self-driving-car.html; Retrieved : March 02, 2015.

[16] General Motors Corporation (GMC), GM Unveils EN-V Concept: A Vision for Future Urban Mobility, Accessible at : http://media.gm.com; Retrieved : March 02, 2015.

[17] Siemens Corporation, Intelligent traffic solutions Accessible at : http://www.siemens.com/; Retrieved : March 02, 2015.

[18] International Business Machine (IBM) Corporation, Intelligent traffic solutions, Accessible at : http://www.ibm.com/smarterplanet/us/en/traffic_congestion/nextsteps/; Retrieved : March 02, 2015.

[19] Carnegie Mellon University (CMU), Smart Traffic Signals, Accessible at: http://www.cmu.edu/homepage/computing/2012/fall/smart-traffic-signals.shtml; Retrieved : March 02, 2015.

[20] L. Petnga and M. A. Austin, Ontologies of Time and Time-based Reasoning for MBSE of Cyber-Physical Systems, 11th Annual Conference on Systems Engineering Research (CSER 2013), Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

[21] L. Petnga and M. A. Austin, Cyber-Physical Architecture for Modeling and Enhanced Operations of Connected-Vehicle Systems, 2nd International Conference on Connected Vehicles (ICCVE 2013). Las Vegas, NV, December 2-6, 2013.

[22] L. Petnga and M. A. Austin, Semantic Platforms for Cyber-Physical Systems, 24th Annual International Council on Systems Engineering International Symposium (INCOSE 2014). Las Vegas, USA, June 30 July 3, 2014.

[23] P. Delgoshaei, M.A. Austin and D. A. Veronica, A Semantic Platform Infrastructure for Requirements Traceability and System Assessment The Ninth International Conference on Systems (ICONS2014), Nice, France, February 23 - 27, 2014.

[24] P. Delgoshaei, M.A. Austin, and A. Pertzborn, A Semantic Framework for Modeling and Simulation of Cyber-Physical Systems, International Journal On Advances in Systems and Measurements, Vol. 7, No. 3-4, December, 2014, pp. 223–238.

Indoor Smartphone Localization with Auto-Adaptive Dead Reckoning

Nils Becker, Michael Jäger, Sebastian Süß
Institute of Software Architecture
Technische Hochschule Mittelhessen - University of Applied Sciences
Gießen, Germany
Email: {nils.becker, michael.jaeger, sebastian.suess}@mni.thm.de

***Abstract*—A common localization method for mobile devices is the fusion of absolute position measurements with relative motion information from sensor units. For each location measurement technique, specific context conditions determine the accuracy of the obtained location estimates. This paper presents a hybrid smartphone localization system fusing an absolute localization method, e.g., Wi-Fi-based signal strength fingerprinting, in an adaptive way with inertial pedestrian navigation, taking into account that each of the involved methods might deliver good results at one location but might also fail at another. Based on an accuracy factor reflecting the current context conditions of a location measurement the influence of each of the involved positioning estimates is weighted accordingly. In a case study using Wi-Fi fingerprinting, accuracy has been improved by 43% in an indoor environment.**

Keywords–Smartphone Positioning; Indoor Positioning; Dead Reckoning; Wi-Fi Fingerprinting; Step Detection;

I. Introduction

Location awareness has become a key feature of many mobile applications. A common problem in the context of navigation and tracking applications is the accurate localization of a mobile device within a well-known area comprising several buildings and also open space, e.g., a company premises, an airport, or a university campus. Such sites are often heterogeneous in the sense that a single localization method delivers good results in one sub-area but fails in another. Solutions typically require hybrid methods comprising a suitable combination of an absolute positioning method with sensor-based relative positioning.

With respect to mobile devices like smartphones an absolute positioning method estimates the device location in terms of latitude and longitude. Relative positioning determines the distance and heading of the movement, when a device is moved to a new position. Elevation might also be of interest. As far as outdoor environments are concerned absolute positioning is commonly based on global navigation satellite systems (GNSS) [1], like the well-known Global Positioning System (GPS) [2], the Russian GLObal NAvigation Satellite System (GLONASS), the Chinese BeiDou, or the european Galileo system. While deviation of second generation GNNS will be in a magnitude of some centimeters in outdoor use [3], satellite systems are not expected to provide sufficient accuracy inside of buildings without being supported by expensive complementary ground component (aka ”pseudolite”) technology [4].

Thus, the quest for accurate and inexpensive indoor localization techniques has fostered intensive research over the last decade and resulted in a number of different promising approaches. While solutions based on cellular signals have not successfully solved the problem of insufficient accuracy, the use of IEEE 802.11 wireless networks, e.g., Wi-Fi, has been widely adopted for real-time indoor localization purposes [5–9]. The rapidly growing usage of Wi-Fi access points as navigation beacons is, among other reasons, due to the ubiquitous availability of Wi-Fi networks and to the fact that a smartphone can easily measure Wi-Fi signal strength values. ”Received Signal Strength Indication” (RSSI) values of several Wi-Fi access points are used to determine the current position of a Wi-Fi receiver. The advent of cheap bluetooth low energy (BLE) beacons [10], e.g., iBeacons [11], might foster their use for the same purpose within the next few years.

Regardless of the beacon types and localization algorithms, absolute indoor localization methods rely on a dense beacon mesh to allow for accurate localization. In a heterogeneous area, thus, a practically important issue is the device localization at spots that lack a sufficiently good beacon signal coverage.

A substantially different approach to localization is dead reckoning, a well-established relative positioning method. Starting from a known position, inertial and other sensors, e.g., accelerometers, gyroscopes, gravity sensors, or barometers, are used to track relative position changes. For example, distance estimation in pedestrian dead reckoning (PDR) systems [12] is typically based on step detection with motion sensors and step length estimation. This is combined with direction information from an electronic compass. Moreover, a barometer could help in determining the current floor in a building. Modern smartphones are crammed with all kinds of sensors and, thus, are well-suited for inertial navigation. Sensor-based localization is, however, subject to unbound accumulating errors, and therefore needs frequent recalibration.

A hybrid method integrates an absolute positioning method with sensor-based navigation. For example, in a GPS-based automotive navigation system sensor-based speed and direction measurements are used to track the current position whenever GPS signals are degraded or unavailable, e.g., in a tunnel. Similarly, a PDR system can be combined with GPS into a hybrid solution for outdoor areas or, together with any absolute indoor position method, e.g., Wi-Fi-based, for use within a building.

An interesting aspect of hybrid systems is the distribution of roles. The absolute positioning could be seen as a minor subsystem of the sensor-based system supplying the start position and, occasionally, intermediate positions for recalibration. However, existing systems typically use the absolute positioning method as a primary method, whereas sensor-based location measurements are only used in case of degraded beacon signals. The absolute base-method is used to compute position estimates (”fixes”) at regular intervals. Each fix is considered a new known start position for inertial navigation. Whenever a fix is not available due to poor signal coverage, the relative movement from the last fix location is used to determine the current device location. A car navigation system, e.g., will use inertial navigation in a tunnel. However, after leaving the tunnel, it will return to the primary method GPS. This commonly used combination pattern does not take into account that, depending on the current beacon reception conditions and despite the accumulating sensor measurement errors, the dead-reckoned position will often be more accurate than the base method fix.

This paper proposes a hybrid localization solution, called ”auto-adaptive dead reckoning”, incorporating a more sophisticated way of combining absolute and relative positioning. Considering that the accuracy of each of the involved methods might fluctuate extremely between measurement locations, the fusing algorithm evaluates context conditions, that are critical for the accuracy, with every measurement. A measurement value which is considered accurate has a stronger impact on the result. The term ”adaptive” is used for a fusion algorithm which associates a weighting factor with each fused method in order to adapt the algorithm to site-specific measurement conditions, e.g., Wi-Fi signal coverage within a building. Static adaptation refers to a configuration time weighting, whereas auto-adaptive (or dynamic) fusion refers to a dynamic weighting for each individual measurement. This advanced fusing technique has been implemented as a component of a mobile application for the Android platform, called SmartLocator [13].

This paper focuses on indoor localization by combining Wi-Fi-based fingerprinting (see II-A) with PDR. Nevertheless, the concept is also applicable to other absolute indoor and outdoor localization techniques, e.g., iBeacons or GPS. Actually, the SmartLocator implementation also comprises localization based on GPS and Near Field Communication (NFC) [14].

After presenting related work in the section II, the concepts of auto-adaptive dead reckoning are described in section III. Section IV discusses experimental results showing the achieved accuracy improvements over non-hybrid as well as hybrid methods with non-dynamic method fusion. Section V reviews some benefits and shortcomings of the presented approach and future research plans.

II. Related Work

A large number of solutions to the problem of real-time indoor localization have been proposed and several efficient algorithms for absolute and relative positioning have been published. Auto-adaptive dead reckoning, as presented in this paper, is based upon Wi-Fi fingerprinting, NFC, and PDR.

A. Wi-Fi-based Fingerprinting

Using an existing Wi-Fi infrastructure for indoor localization is an obvious and well-investigated approach. While RSSI-based distance calculations have proven to be too inaccurate to be used for trilateration-based indoor localization, RSSI-fingerprinting methods are particularly useful in the context of real-time smartphone positioning [5–9].

Fingerprinting is based on a probability distribution of signal strengths at a given location. A map of these distributions is used to predict a location from RSSI samples. From each visible access point the mobile device receives beacon signals. The set of all pairs consisting of access point ID and RSSI value can be seen as a fingerprint for the device’s current location. In order to determine the device position, a database is searched for similar fingerprints. The database itself is created in an offline learning phase, which links fingerprints to a number of known locations called calibration points.

A major advantage of Wi-Fi fingerprinting is that it does not require specialized hardware [6][15][16]. Nevertheless, a non-dynamical Wi-Fi infrastructure with good coverage is needed to achieve reasonable positioning results.

However, the most important disadvantage is the elaborate fingerprint database creation and maintenance. Since the accuracy of estimated positions highly depends on the density of the radio map [6], the construction of a high-density map is inevitable for Wi-Fi-only positioning solutions. The auto-adaptive algorithm, in contrast, allows for a significant reduction of the number of calibration points without loosing too much overall accuracy.

In order to avoid the map creation overhead completely, zero-effort solutions based on crowdsourcing have been proposed [17][18]. Although efficient map creation is outside the scope of this paper, it should be noted that map creation and map usage algorithms are typically loosely coupled. Thus, any successful approach to automate map creation could possibly be generalized for usage with existing fingerprinting systems.

B. Sensor-based Positioning

According to [19], PDR systems can be classified as Inertial Navigation Systems (INSs) or Step-and-Heading Systems (SHSs). While the INSs typically require specialized hardware, the SHSs are well-suited for PDR with smartphones.

The SmartLocator solution presented in this paper implements an SHS, which builds upon efficient algorithms for step detection and heading estimation. The heading is determined by a sensor fusion method described in [20]. Step detection exploits the smartphone’s accelerometer signals. Whenever a peak with a certain amplitude at the z-axis is noticed, a step can be assumed [21]. A modified Pan-Tompkins algorithm is used for signal preparation. Pan-Tompkins, in the context of step detection, has been used by Ying [22] before.

C. Method Fusion

An interesting approach combining Wi-Fi-based fingerprinting with PDR was proposed in [23]. Their fusing algorithm uses a limited history of location measurements for both methods to achieve accurate position estimations. Another promising solution is described in [24] . The algorithm builds on a statistical model for Wi-Fi-localization avoiding the effort of fingerprinting map creation, deliberately taking into account the resulting poor accuracy of the obtained position information. Both fusing methods comprise the use of floor plans and particle filters in order to obtain more accurate position information [25].

Particle filtering, however, comes with some drawbacks, particularly the algorithmic complexity which results in a high processor load and impacts power consumption. Moreover, suitable floor maps have to be supplied and maintained.

III. Proposed Positioning System

This section describes auto-adaptive dead reckoning and its implementation in the SmartLocator positioning system. SmartLocator actually implements a multi-method approach comprising indoor as well as outdoor positioning with seamless transitions. In order to exploit the capabilities of a modern smartphone, the system supports various absolute positioning techniques such as GPS, NFC, and Wi-Fi. Additional support for Bluetooth Low Energy beacons is in preparation. The absolute methods are used opportunistically, depending on their availability.

Although a general discussion of the fusion of several absolute methods is out of scope of this paper, it is worth noting that positions determined with GPS, Wi-Fi, or BLE are considered inaccurate, whereas NFC-based positioning is treated as accurate. The smartphone nearly has to get in touch with an NFC tag in order to read it. Hence, reading a tag with a precisely known position also reveals the exact position of the reading device. Whenever a precise location can be obtained, it overrides all other measurements.

In addition to the absolute positioning capabilities, SmartLocator incorporates a PDR subsystem with step detection and heading estimation. The stride size is simply set to a user-specific fixed value. However, using the absolute localization methods, it could straightforwardly be augmented with automatic stride size recalibration.

The emphasis of this paper is to present the way of fusing PDR with an absolute positioning method. The term ”auto-adaptive dead-reckoning” refers to this fusing approach. From the perspective of PDR, absolute localization is needed to obtain an initial position and for recalibration. In contrast to a full recalibration, we propose a partial recalibration determined by a dynamic weight, which reflects the accuracy of the absolute location estimation. Although this section concentrates on the fusion of PDR and Wi-Fi fingerprinting, the approach is not confined to a specific absolute localization method. It is rather a particular strength of the approach to be method-independent.

Figure 1 illustrates how absolute location sources are combined with relative positioning information.

The following subsections describe the Wi-Fi fingerprinting approach (III-A), the step detection algorithm (III-B) and the auto-adaptive fusion (III-C).

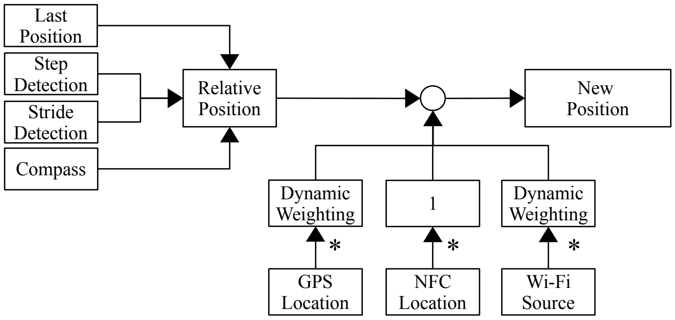


Figure 1. SmartLocator Positioning Concept

A. Fingerprinting

The fingerprint-based position is computed with help of the naïve Bayes classifier [6][15][16], which is more accurate than algorithms comparing distances between RSSIs [26–29]. This advantage has been confirmed during the evaluation of this positioning system.

The naïve Bayes classifier is based on the Bayes theorem, which defines the probability P of the class C under the assumption that x is given as follows:

$$P(C|x) = \frac{P(C)P(x|C)}{P(x)} \tag{1}$$

In case of fingerprinting, $P(C|x)$ describes the probability that fingerprint x belongs to the class C, which represents a position. x is a vector of RSSI values.

It is assumed that all values of the input vector x are independent of each other. For this reason, the conditional probability $P(x|C)$ is the product of the probability of each element in x given class C, $P(x_i|C)$.

$$P(x|C) = \prod_i P(x_i|C) \tag{2}$$

A common approach to compute the likelihood $P(x|C)$, which depends on the training data, is the following [26][15, p. 36]:

$$P(x_i|C) = \frac{1}{n}\sum_{j=1}^{n} K_{Gauss}(x_i, y_j) \tag{3}$$

$$K_{Gauss} = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) \tag{4}$$

K denotes the kernel function. x is the observed fingerprint and y are all fingerprints, recorded for location C. n is the number of recorded fingerprints for location C.

The actual position is interpolated from the three best fitting fingerprints.

$$C_{NN} = \frac{\prod_{i=1}^{3} C_i * P(C_i|x)}{\sum_{i=1}^{3} P(C_i|x)} \tag{5}$$

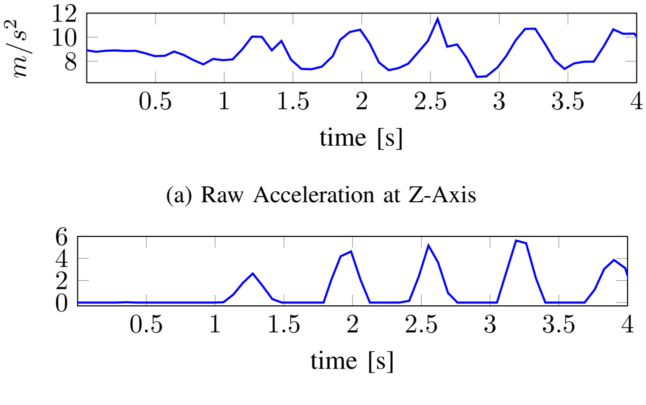
B. Step Detection

The step detection algorithm recognizes pedestrian movements based on a simple peak detection algorithm described by Link et al. [21]. To improve the amount of detected steps and decrease the appearance of false positive detections, the signal is prepared by applying a slightly modified version of the *Pan-Tompkins* method.

$$y(n) = \begin{cases} \frac{1}{4}[2x(n) + x(n-1) - x(n-3) - 2x(n-4)] & \text{if } y(n) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$y(n) = (1 + y(n))^2 - 1 \quad (7)$$

A derivative operator uses low-pass filtered accelerometer values in order to suppress low-frequency components and enlarge the high frequency components from the high slopes (6). Negative values are discarded, as they are not needed for the peak detection. Figure 2 shows the incoming acceleration signal before (a) and after (b) this preparation.



(a) Raw Acceleration at Z-Axis

(b) Squared Derivative Signal

Figure 2. Acceleration Measurements Before and After Preparation

The step detection algorithm examines the signal for peaks by comparing the last three values, represented by the red squares in figure 3. A step is assumed whenever the signal changes by a certain threshold. After a step has been detected, the algorithm pauses for 300ms to prevent a step from being detected twice.

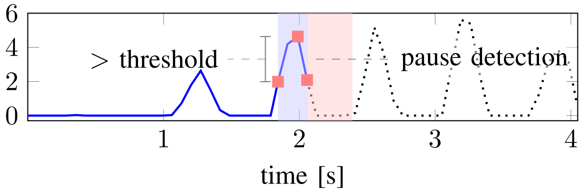


Figure 3. Step Detection Example. Red Squares Represent Analyzed Values

C. Auto-Adaptive Dead Reckoning

The major innovation of SmartLocator's hybrid localization is the accuracy-dependent fusion of absolute and relative positions. Traditional dead reckoning systems overwrite past position determinations whenever a new absolute position is available. This is not reasonable whenever absolute positions' accuracy is bad or varying. Therefore, every absolute position is reckoned with past position estimations. The weighting of the new absolute position depends on an estimation of its accuracy. As a consequence, accurate absolute positions have a greater influence on the final position than less reliable position estimates.

E.g., Wi-Fi positions determined in an area with poor Wi-Fi coverage just have little influence on the final position estimation and the position determined by detecting the pedestrian's steps and heading is weighted strongly. On the other hand, Wi-Fi positions which are determined in an area with lots of access points and good signal quality are used to correct the drift which may occur due to inaccuracies in step detection and heading estimation.

Let $Loc(i,t)$ be a measurement obtained by localization method $M(i)$ at time t, e.g., an absolute Wi-Fi or GPS position. The contribution of $Loc(i,t)$ to the resulting location information depends on the method-specific accuracy factor. The accuracy factor $Q(Loc(i,t))$ is obtained by context evaluation and reflects the measurement's context-dependent reliability.

In addition, a time-dependent factor α_t is added to the accuracy factor. In this way, positions have a stronger influence if the last position determination was long ago. The linear α_t used in SmartLocator is represented by figure 4.

$$\alpha = max(Q(Loc(i,t)) + \alpha_t, 1) \quad (8)$$

$$Loc(t) = Loc(i,t) * \alpha + Loc(t-1) * (1-\alpha) \quad (9)$$

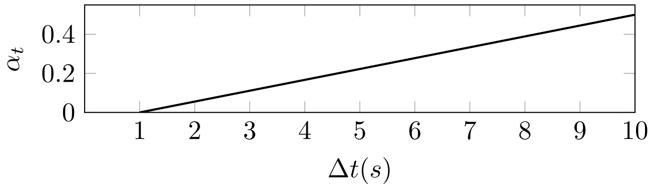


Figure 4. Time-Dependent Factor

1) Accuracy Factors: The accuracy factor $Q(Loc(i,t))$ depends on the method i used for positioning. This section describes various methods to compute the accuracy factor for Wi-Fi fingerprinting, GPS and NFC.

Wi-Fi: Evaluations revealed an average error of 2.94 meters for pure Wi-Fi positioning. However, the error varied from 0.07 to 7.99 meters. Figure 5 shows the analysis of the gathered test data, revealing a relation between the average error and the amount of access points, which have been available for position determination. Even in case of good Wi-Fi coverage, error varies from 0.3 to 7.3 meters. The accuracy factor $Q(Loc(\text{wifi},t))$, illustrated in figure 6, takes this relation into account to reduce the influence of unreliable position measurements.

Copyright (c) IARIA, 2015.   ISBN: 978-1-61208-399-5

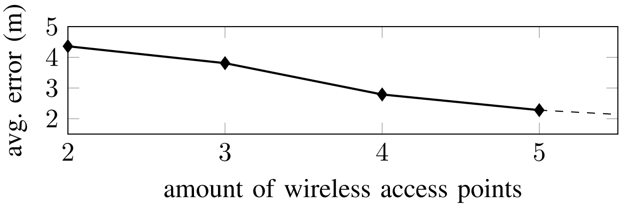


Figure 5. Accuracy Factor for Wi-Fi positioning

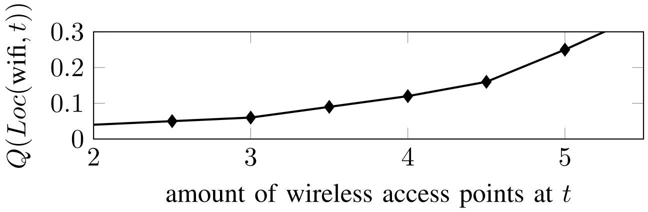


Figure 6. Wi-Fi Accuracy Factor Depending on Amount of Access Points

GPS: The GPS position is determined via the smart phone's operating systems' API. Each GPS position includes an accuracy property, which represents an estimated average error in meters. The accuracy-factor, shown in figure 7, is based on this accuracy property.

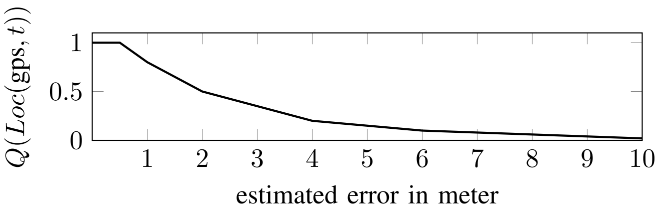


Figure 7. Accuracy Factor for GPS positioning

NFC: Near Field Communication (NFC) is used for positioning by placing passive NFC tags at points of interest. In order to scan an NFC tag, the smart phone needs to get in touch with it. Therefore, the location of the smart phone can be expected to be the location of the NFC tag. As a consequence, the accuracy factor $Q(Loc(\text{nfc}, t))$ always returns the maximum value of 1, which means that an NFC position overwrites prior location determinations completely.

IV. Evaluation

SmartLocator has been tested under realistic circumstances in a university campus. Using eight Wi-Fi access points for positioning, fingerprints at 67 different locations have been recorded. The fingerprint locations are distributed equally with a distance of two meters. Hence, an area of about 280 m^2 is covered. Four orientations have been measured for any location. Three fingerprints for each orientation, resulting in an overall amount of 804 fingerprints.

A track of 70 meters has been walked in various speeds, with different devices and in different directions to get a representative evaluation. 14 reference positions have been marked at the track. Those known reference positions are compared to the estimated positions, to determine the accuracy of the different approaches. Figure 8 shows the test environment, including the test track, which is illustrated by a grey line.

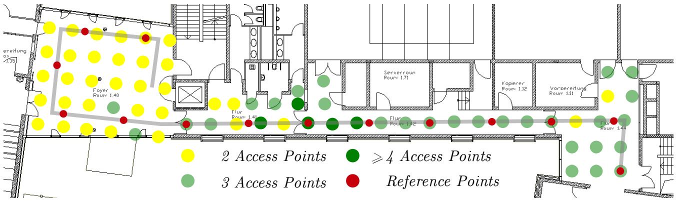
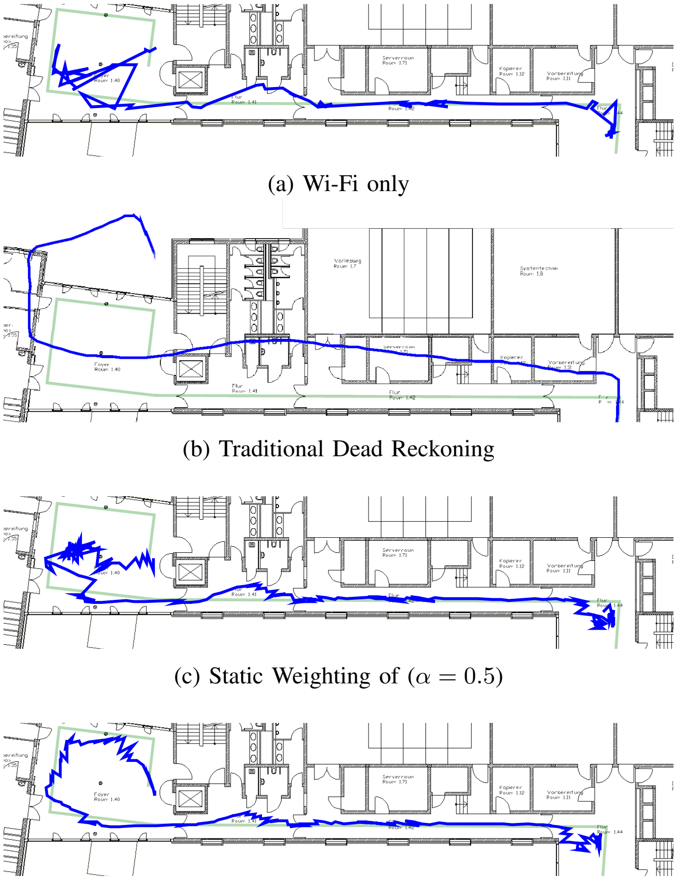


Figure 8. Wi-Fi Positioning Test Area with Fingerprints

Figure 9 shows a visualization of one test run. The test started in the bottom right corner and followed the light green path. The blue line represents the actual positioning result. Figure 9b shows the results gathered with traditional dead reckoning, which means that absolute positioning results overwrite prior positioning estimations. Figure 9c presents a static weighting of 0.5, i.e., new absolute positions are just reckoned up by half. Figure 9d visualizes the positioning results achieved with a dynamic, auto-adaptive combination.



(a) Wi-Fi only

(b) Traditional Dead Reckoning

(c) Static Weighting of ($\alpha = 0.5$)

(d) Auto-Adaptive Weighting

Figure 9. Comparison of Different Weightings

Remarkably, all figures reveal a clearly visible deviation from the real path at the same location (in front of the restrooms, left of the middle). This results from a coincidence of two local environment conditions. The first factor is the poor Wi-Fi-coverage in this area. Furthermore, a heavy metal fire door impacts the magnetometer of the electronic compass. Obviously, if neither of the involved measurement methods obtains an accurate location, the method fusion cannot compensate the resulting drift completely.

The evaluation revealed that the traditional dead reckoning (Trad. D.R.) approach performed even a little bit worse than the pure Wi-Fi positioning. A static combination of relative and absolute positions was able to slightly improve the positioning accuracy, especially in the foyer at the left side of the floor plan. Auto-adaptive combination of Wi-Fi and relative positioning is able to reduce the average positioning error significantly. The average error has been improved from 2.94m (Wi-Fi only) to 1.67 meters, the upper quartile from 3.54m to 2.29m.

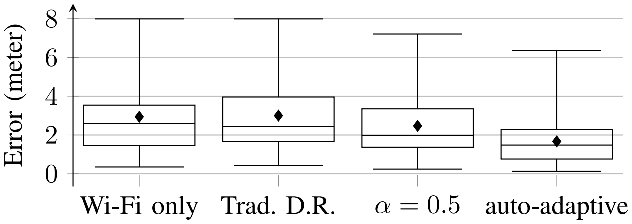


Figure 10. Comparison of Wi-Fi-only Positioning, Classic Dead Reckoning, Static and Dynamic Weighting

V. Conclusion

The positioning system described in this paper reveals a significant increase of localization accuracy through auto-adaptive combination of absolute and relative positions. Even though the accuracy estimations for Wi-Fi positioning are rather rudimentary, the average error has been reduced by 1.27 meters to 1.67 meters. Comparing the average absolute deviation with the results of other solutions, e.g. [23], the auto-adaptive dead reckoning approach seems to be quite promising, although additional evaluations with different environment conditions are necessary to gain more confidence in the statistical evaluation. More sophisticated accuracy estimation methods [30] and the additional use of floor map information [24] could probably improve this result further.

The evaluation shows that areas with bad Wi-Fi coverage and large rooms benefit the most. As a result, this positioning system can be used in areas which do not meet the requirements for Wi-Fi-only positioning approaches.

An unsolved problem is the determination of an initial position at starting locations with poor Wi-Fi coverage. Considering the enormous effort needed to construct a fingerprinting database, it obviously makes sense to also consider the selective deployment of NFC tags in such areas. These tags are cheap, permit exact localization, and will be supported by the vast majority of future smartphones. Moreover, the implementation of NFC-based localization has shown to be rather uncomplicated.

It is an important characteristic of the auto-adaptive fusion method that it is independent from the evaluated positioning methods. It could be applied to any other technique as long as a weighting factor can be determined. It can be assumed that GPS-based outdoor positioning and BLE-based indoor techniques benefit similarly. However, the quantitative evaluation is still in progress.

We consider some performance aspects at last. The low-complexity fusion method and the avoidance of elaborate probabilistic algorithms for particle filtering result in a good real-time behavior. Several test runs with different smartphones have shown that even on low-end hardware the SmartLocator runs without any visible performance problems. However, a more detailed analysis of algorithmic performance factors would be interesting, since time-consuming computations have negative effects on response times and power consumption.

References

[1] C. J. Hegarty and E. Chatre, “Evolution of the global navigation satellitesystem (gnss),” Proceedings of the IEEE, vol. 96, no. 12, 2008, pp. 1902–1917.

[2] US Government, “Official U.S. government information about the global positioning system (GPS) and related topics,” http://www.gps.gov, 2015, accessed: March, 3rd 2015.

[3] P. Misra and P. Enge, Global Positioning System: Signals, Measurements and Performance Second Edition. Lincoln, MA: Ganga-Jamuna Press, 2006.

[4] J. Wang et al., “Pseudolite applications in positioning and navigation: Progress and problems,” Positioning, vol. 1, no. 03, 2002.

[5] M. Youssef and A. Agrawala, “The Horus WLAN location determination system,” in Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, ser. MobiSys ’05, 2005, pp. 205–218.

[6] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 2. Ieee, 2000, pp. 775–784.

[7] M. Weber, U. Birkel, R. Collmann, and J. Engelbrecht, “Wireless indoor positioning: Localization improvements with a leaky coaxial cable prototype,” in 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Guimaraes, Portugal, pp. 21–23.

[8] T.-N. Lin, S.-H. Fang, W.-H. Tseng, C.-W. Lee, and J.-W. Hsieh, “A group-discrimination-based access point selection for WLAN fingerprinting localization,” Vehicular Technology, IEEE Transactions on, vol. 63, no. 8, 2014, pp. 3967–3976.

[9] M. Brunato and R. Battiti, “Statistical learning theory for location fingerprinting in wireless LANs,” Computer Networks, vol. 47, no. 6, 2005, pp. 825–845.

[10] C. Gomez, J. Oller, and J. Paradells, “Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology,” Sensors, vol. 12, no. 9, 2012, pp. 11 734–11 753.

[11] N. Newman, “Apple iBeacon technology briefing,” Journal of Direct, Data and Digital Marketing Practice, vol. 15, no. 3, 2014, pp. 222–225.

[12] S. Beauregard and H. Haas, “Pedestrian dead reckoning: A basis for personal positioning,” in Proceedings of the 3rd Workshop on Positioning, Navigation and Communication, 2006, pp. 27–35.

[13] N. Becker, “Development of a location-based information and navigation system for indoor and outdoor areas,” Master’s

thesis, Technische Hochschule Mittelhessen, Giessen, Germany, 2014.

[14] R. Want, “Near field communication,” IEEE Pervasive Computing, vol. 10, no. 3, 2011, pp. 4–7.

[15] M. A. Rehim, “Horus: A WLAN-based indoor location determination system,” Ph.D. dissertation, University of Maryland, 2004.

[16] Y. Chen and H. Kobayashi, “Signal strength based indoor geolocation,” Princeton University, Tech. Rep., 2002.

[17] P. Bolliger, “Redpin-adaptive, zero-configuration indoor localization through user collaboration,” in Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments. ACM, 2008, pp. 55–60.

[18] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, “Zee: zero-effort crowdsourcing for indoor localization,” in Proceedings of the 18th annual international conference on Mobile computing and networking. ACM, 2012, pp. 293–304.

[19] R. Harle, “A survey of indoor inertial positioning systems for pedestrians,” IEEE Communications Surveys & Tutorials, no. 15, 2013, pp. 1281–1293.

[20] S. Ayub, A. Bahraminisaab, and B. Honary, “A sensor fusion method for smart phone orientation estimation,” in 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting., 2012.

[21] J. B. Link, P. Smith, N. Viol, and K. Wehrle, “Footpath: Accurate map-based indoor navigation using smartphones,” in Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on. IEEE, 2011, pp. 1–8.

[22] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek, “Automatic step detection in the accelerometer signal,” in 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007). Springer, 2007, pp. 80–85.

[23] L.-H. Chen, E.-K. Wu, M.-H. Jin, and G.-H. Chen, “Intelligent fusion of wi-fi and inertial sensor-based positioning systems for indoor pedestrian navigation,” 2014.

[24] F. Ebner, F. Deinzer, L. Köping, and M. Grzegorzek, “Robust self-localization using Wi-Fi, step/turn-detection and recursive density estimation,” in International Conference on Indoor Positioning and Indoor Navigation, vol. 27, 2014, p. 30th.

[25] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics. MIT press, 2005.

[26] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piché, “A comparative survey of WLAN location fingerprinting methods,” in Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on. IEEE, 2009, pp. 243–251.

[27] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg, “Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses,” in Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization. ACM, 2006, pp. 34–40.

[28] J. Letchner, D. Fox, and A. LaMarca, “Large-scale localization from wireless signal strength,” in Proceedings of the national conference on artificial intelligence, vol. 20, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 15.

[29] A. Bekkelien, M. Deriaz, and S. Marchand-Maillet, “Bluetooth indoor positioning,” Master’s thesis, University of Geneva, 2012.

[30] H. Lemelson, M. B. Kjærgaard, R. Hansen, and T. King, “Error estimation for indoor 802.11 location fingerprinting,” in Location and Context Awareness. Springer, 2009, pp. 138–155.