International Journal on Advances in Systems and Measurements

2018 vol. 11 nr. 3&4

The *International Journal on Advances in Systems and Measurements* is published by IARIA.
ISSN: 1942-261x
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Systems and Measurements, issn 1942-261x
vol. 11, no. 3 & 4, year 2018, http://www.iariajournals.org/systems_and_measurements/

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Systems and Measurements, issn 1942-261x
vol. 11, no. 3 & 4, year 2018, http://www.iariajournals.org/systems_and_measurements/

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA
www.iaria.org

Copyright © 2018 IARIA

International Journal on Advances in Systems and Measurements
Volume 11, Number 3 & 4, 2018

Editors-in-Chief

Constantin Paleologu, University "Politehnica" of Bucharest, Romania
Sergey Y. Yurish, IFSA, Spain

Editorial Advisory Board

Vladimir Privman, Clarkson University - Potsdam, USA
Winston Seah, Victoria University of Wellington, New Zealand
Mohammed Rajabali Nejad, Universiteit Twente, the Netherlands
Nageswara Rao, Oak Ridge National Laboratory, USA
Roberto Sebastian Legaspi, Transdisciplinary Research Integration Center | Research Organization of Information and System, Japan
Victor Ovchinnikov, Aalto University, Finland
Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany
Teresa Restivo, University of Porto, Portugal
Stefan Rass, Universität Klagenfurt, Austria
Candid Reig, University of Valencia, Spain
Qingsong Xu, University of Macau, Macau, China
Paulo Estevao Cruvinel, Embrapa Instrumentation Centre - São Carlos, Brazil
Javad Foroughi, University of Wollongong, Australia
Andrea Baruzzo, University of Udine / Interaction Design Solution (IDS), Italy
Cristina Seceleanu, Mälardalen University, Sweden
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway

Indexing Liaison Chair

Teresa Restivo, University of Porto, Portugal

Editorial Board

Jemal Abawajy, Deakin University, Australia
Ermeson Andrade, Universidade Federal de Pernambuco (UFPE), Brazil
Francisco Arcega, Universidad Zaragoza, Spain
Tulin Atmaca, Telecom SudParis, France
Lubomír Bakule, Institute of Information Theory and Automation of the ASCR, Czech Republic
Andrea Baruzzo, University of Udine / Interaction Design Solution (IDS), Italy
Nicolas Belanger, Eurocopter Group, France
Lotfi Bendaouia, ETIS-ENSEA, France
Partha Bhattacharyya, Bengal Engineering and Science University, India
Karabi Biswas, Indian Institute of Technology - Kharagpur, India

Jonathan Blackledge, Dublin Institute of Technology, UK
Dario Bottazzi, Laboratori Guglielmo Marconi, Italy
Diletta Romana Cacciagrano, University of Camerino, Italy
Javier Calpe, Analog Devices and University of Valencia, Spain
Jaime Calvo-Gallego, University of Salamanca, Spain
Maria-Dolores Cano Baños, Universidad Politécnica de Cartagena,Spain
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Vítor Carvalho, Minho University & IPCA, Portugal
Irinela Chilibon, National Institute of Research and Development for Optoelectronics, Romania
Soolyeon Cho, North Carolina State University, USA
Hugo Coll Ferri, Polytechnic University of Valencia, Spain
Denis Collange, Orange Labs, France
Noelia Correia, Universidade do Algarve, Portugal
Pierre-Jean Cottinet, INSA de Lyon - LGEF, France
Paulo Estevao Cruvinel, Embrapa Instrumentation Centre - São Carlos, Brazil
Marc Daumas, University of Perpignan, France
Jianguo Ding, University of Luxembourg, Luxembourg
António Dourado, University of Coimbra, Portugal
Daniela Dragomirescu, LAAS-CNRS / University of Toulouse, France
Matthew Dunlop, Virginia Tech, USA
Mohamed Eltoweissy, Pacific Northwest National Laboratory / Virginia Tech, USA
Paulo Felisberto, LARSyS, University of Algarve, Portugal
Javad Foroughi, University of Wollongong, Australia
Miguel Franklin de Castro, Federal University of Ceará, Brazil
Mounir Gaidi, Centre de Recherches et des Technologies de l'Energie (CRTEn), Tunisie
Eva Gescheidtova, Brno University of Technology, Czech Republic
Tejas R. Gandhi, Virtua Health-Marlton, USA
Teodor Ghetiu, University of York, UK
Franca Giannini, IMATI - Consiglio Nazionale delle Ricerche - Genova, Italy
Gonçalo Gomes, Nokia Siemens Networks, Portugal
Luis Gomes, Universidade Nova Lisboa, Portugal
Antonio Luis Gomes Valente, University of Trás-os-Montes and Alto Douro, Portugal
Diego Gonzalez Aguilera, University of Salamanca - Avila, Spain
Genady Grabarnik,CUNY - New York, USA
Craig Grimes, Nanjing University of Technology, PR China
Stefanos Gritzalis, University of the Aegean, Greece
Richard Gunstone, Bournemouth University, UK
Jianlin Guo, Mitsubishi Electric Research Laboratories, USA
Mohammad Hammoudeh, Manchester Metropolitan University, UK
Petr Hanáček, Brno University of Technology, Czech Republic
Go Hasegawa, Osaka University, Japan
Henning Heuer, Fraunhofer Institut Zerstörungsfreie Prüfverfahren (FhG-IZFP-D), Germany
Paloma R. Horche, Universidad Politécnica de Madrid, Spain
Vincent Huang, Ericsson Research, Sweden
Friedrich Hülsmann, Gottfried Wilhelm Leibniz Bibliothek - Hannover, Germany
Travis Humble, Oak Ridge National Laboratory, USA

Florentin Ipate, University of Pitesti, Romania
Imad Jawhar, United Arab Emirates University, UAE
Terje Jensen, Telenor Group Industrial Development, Norway
Liudi Jiang, University of Southampton, UK
Kenneth B. Kent, University of New Brunswick, Canada
Fotis Kerasiotis, University of Patras, Greece
Andrei Khrennikov, Linnaeus University, Sweden
Alexander Klaus, Fraunhofer Institute for Experimental Software Engineering (IESE), Germany
Andrew Kusiak, The University of Iowa, USA
Vladimir Laukhin, Institució Catalana de Recerca i Estudis Avançats (ICREA) / Institut de Ciencia de Materials de Barcelona (ICMAB-CSIC), Spain
Kevin Lee, Murdoch University, Australia
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Andreas Löf, University of Waikato, New Zealand
Jerzy P. Lukaszewicz, Nicholas Copernicus University - Torun, Poland
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Stefano Mariani, Politecnico di Milano, Italy
Paulo Martins Pedro, Chaminade University, USA / Unicamp, Brazil
Don McNickle, University of Canterbury, New Zealand
Mahmoud Meribout, The Petroleum Institute - Abu Dhabi, UAE
Luca Mesin, Politecnico di Torino, Italy
Marco Mevius, HTWG Konstanz, Germany
Marek Miskowicz, AGH University of Science and Technology, Poland
Jean-Henry Morin, University of Geneva, Switzerland
Fabrice Mourlin, Paris 12th University, France
Adrian Muscat, University of Malta, Malta
Mahmuda Naznin, Bangladesh University of Engineering and Technology, Bangladesh
George Oikonomou, University of Bristol, UK
Arnaldo S. R. Oliveira, Universidade de Aveiro-DETI / Instituto de Telecomunicações, Portugal
Aida Omerovic, SINTEF ICT, Norway
Victor Ovchinnikov, Aalto University, Finland
Telhat Özdoğan, Recep Tayyip Erdogan University, Turkey
Gurkan Ozhan, Middle East Technical University, Turkey
Constantin Paleologu, University Politehnica of Bucharest, Romania
Matteo G A Paris, Universita` degli Studi di Milano,Italy
Vittorio M.N. Passaro, Politecnico di Bari, Italy
Giuseppe Patanè, CNR-IMATI, Italy
Marek Penhaker, VSB- Technical University of Ostrava, Czech Republic
Juho Perälä, Bitfactor Oy, Finland
Florian Pinel, T.J.Watson Research Center, IBM, USA
Ana-Catalina Plesa, German Aerospace Center, Germany
Miodrag Potkonjak, University of California - Los Angeles, USA
Alessandro Pozzebon, University of Siena, Italy
Vladimir Privman, Clarkson University, USA
Mohammed Rajabali Nejad, Universiteit Twente, the Netherlands

Konandur Rajanna, Indian Institute of Science, India
Nageswara Rao, Oak Ridge National Laboratory, USA
Stefan Rass, Universität Klagenfurt, Austria
Candid Reig, University of Valencia, Spain
Teresa Restivo, University of Porto, Portugal
Leon Reznik, Rochester Institute of Technology, USA
Gerasimos Rigatos, Harper-Adams University College, UK
Luis Roa Oppliger, Universidad de Concepción, Chile
Ivan Rodero, Rutgers University - Piscataway, USA
Lorenzo Rubio Arjona, Universitat Politècnica de València, Spain
Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance, Germany
Subhash Saini, NASA, USA
Mikko Sallinen, University of Oulu, Finland
Christian Schanes, Vienna University of Technology, Austria
Rainer Schönbein, Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Germany
Cristina Seceleanu, Mälardalen University, Sweden
Guodong Shao, National Institute of Standards and Technology (NIST), USA
Dongwan Shin, New Mexico Tech, USA
Larisa Shwartz, T.J. Watson Research Center, IBM, USA
Simone Silvestri, University of Rome "La Sapienza", Italy
Diglio A. Simoni, RTI International, USA
Radosveta Sokullu, Ege University, Turkey
Junho Song, Sunnybrook Health Science Centre - Toronto, Canada
Leonel Sousa, INESC-ID/IST, TU-Lisbon, Portugal
Arvind K. Srivastav, NanoSonix Inc., USA
Grigore Stamatescu, University Politehnica of Bucharest, Romania
Raluca-Ioana Stefan-van Staden, National Institute of Research for Electrochemistry and Condensed Matter, Romania
Pavel Šteffan, Brno University of Technology, Czech Republic
Chelakara S. Subramanian, Florida Institute of Technology, USA
Sofiene Tahar, Concordia University, Canada
Muhammad Tariq, Waseda University, Japan
Roald Taymanov, D.I.Mendeleyev Institute for Metrology, St.Petersburg, Russia
Francesco Tiezzi, IMT Institute for Advanced Studies Lucca, Italy
Wilfried Uhring, University of Strasbourg // CNRS, France
Guillaume Valadon, French Network and Information and Security Agency, France
Eloisa Vargiu, Barcelona Digital - Barcelona, Spain
Miroslav Velev, Aries Design Automation, USA
Dario Vieira, EFREI, France
Stephen White, University of Huddersfield, UK
Shengnan Wu, American Airlines, USA
Qingsong Xu, University of Macau, Macau, China
Xiaodong Xu, Beijing University of Posts & Telecommunications, China
Ravi M. Yadahalli, PES Institute of Technology and Management, India
Yanyan (Linda) Yang, University of Portsmouth, UK

Shigeru Yamashita, Ritsumeikan University, Japan
Patrick Meumeu Yomsi, INRIA Nancy-Grand Est, France
Alberto Yúfera, Centro Nacional de Microelectronica (CNM-CSIC) - Sevilla, Spain
Sergey Y. Yurish, IFSA, Spain
David Zammit-Mangion, University of Malta, Malta
Guigen Zhang, Clemson University, USA
Weiping Zhang, Shanghai Jiao Tong University, P. R. China

International Journal on Advances in Systems and Measurements
Volume 11, Numbers 3 & 4, 2018

CONTENTS

FPGAs and the Cloud – An Endless Tale of Virtualization, Elasticity and Efficiency

Oliver Knodel*‡, Paul R. Genssler†‡, Fredo Erxleben‡ and Rainer G. Spallek‡

* Department of Information Services and Computing, Helmholtz-Zentrum Dresden-Rossendorf, Dresden, Germany
† Department of Computer Science, Karlsruhe Institute of Technology, Karlsruhe, Germany
‡ Department of Computer Science, Technische Universität Dresden, Dresden, Germany
Email: *o.knodel@hzdr.de, †genssler@kit.edu, ‡{firstname.lastname}@tu-dresden.de

***Abstract*—Field Programmable Gate Arrays (FPGAs) provide a promising opportunity to improve performance, security and energy efficiency of computing architectures, which are essential in modern data centers. Especially the background acceleration of complex and computationally intensive tasks is an important field of application. The flexible use of reconfigurable devices within a cloud context requires abstraction from the actual hardware through virtualization to offer these resources to service providers. In this paper, we present our Reconfigurable Common Computing Frame (RC2F) approach – inspired by system virtual machines – for the profound virtualization of reconfigurable hardware in cloud services. Using partial reconfiguration, our framework abstracts a single physical FPGA into multiple independent virtual FPGAs (vFPGAs). A user can request vFPGAs of different size for optimal resource utilization and energy efficiency of the whole cloud system. To enable such flexibility, we create homogeneous partitions on top of an inhomogeneous FPGA fabric abstracting from physical locations and static areas. The $RC2F_{SEC}$ extension combines this virtualization with a security system to allow for processing of sensitive data. On the host side our Reconfigurable Common Cloud Computing Environment (RC3E) offers different service models and manages the allocation of the dynamic vFPGAs. We demonstrate the possibilities and the resource trade-off of our approach in a basic scenario. Moreover, we present future perspectives for the use of FPGAs in cloud-based environments.**

Keywords–Cloud Computing; Virtualization; Reconfigurable Hardware; Partial Reconfiguration.

I. Motivation

The idea of FPGAs as virtualized resources in Cloud environments in the projects RC3E and RC2F was temporarily completed with introducing homogeneous virtualized FPGAs in 2017 by Knodel et al. in [1]. This article henceforth describes the two parts of out project – RC3E and RC2F – beginning with first considerations related to FPGA-Clusters in [2]. First cloud approaches with service models were introduced in [3] and [4], the overall RC3E-Cloud description in [5], a hardware migration in [6] and additional security considerations by Genssler et al. in [7].

Cloud computing itself is based on the idea of computing as a utility [8]. The user gains access to a shared pool of computing resources or services that can rapidly be allocated and released "with minimal management effort or service provider interaction" [9]. An essential advantage, compared to traditional models in which the user has access to a fixed number of computing resources, is the elasticity within a cloud. Even in peak load situations, a sufficient amount of resources are available [8].

With the theoretically unlimited number of resources, their enormous energy consumption arises as a major problem for data centers housing clouds. One possibility to enhance computation performance by simultaneously lowering energy consumption is the use of heterogeneous systems, offloading computationally intensive applications to special hardware coprocessors or dedicated accelerators. Especially reconfigurable hardware, such as FPGAs, provide an opportunity to improve computing performance [10], security [11] and energy efficiency [12].

A profound and flexible integration of FPGAs into scalable data center infrastructures, which satisfies the cloud characteristics, is a task of growing importance in the field of energy-efficient cloud computing. In order to achieve such an integration, the virtualization of FPGA resources is necessary. Provisioning vFPGAs makes reconfigurable resources available to customers of the data center provider. These customers are usually service providers themselves – nevertheless, they will be called *users* throughout this paper. Those users can accelerate their specific services, reduce energy consumption and thereby service costs.

The virtualization of reconfigurable hardware devices is a recurring challenge. Decades ago, the virtualization of FPGA devices started due to the limitation of logical resources [13]. Nowadays, FPGAs have grown in size and full utilization of the devices cannot always be achieved in practice. One possibility to increase utilization is our virtualization approach, which allows for flexible design sizes and multiple hardware designs on the same physical FPGA. One challenge of this approach are the unsteady load situations of elastic clouds, which process short- and long-running acceleration tasks.

In this paper, we introduce our virtualization concept for FPGAs, which is inspired by traditional virtual machines (VMs). One physical FPGA can consist of multiple vFPGAs belonging to different services with different runtimes. Each vFPGA can be configured using partial reconfiguration [14] and the internal configuration access port (ICAP). The vFPGAs are, therefore, flexible in their physical size and location. This vertical scalability of vFPGAs from a small design up to a full physical FPGA enables an efficient utilization of the reconfigurable resources. Moreover, the vFPGAs are fully homogeneous among each other and thereby become a wholesome virtualized cloud component, which also supports an efficient migration of a whole vFPGA context.

The paper is structured as follows: Section II introduces similar concepts and related research in the field of vir-

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

tualization of reconfigurable hardware, cloud architectures and bitstream relocation. The requirements for a profound provision of FPGAs in a cloud environment are discussed in Section III. Section IV introduces the prototypical cloud management system RC3E followed by definitions necessary for the virtualization of the FPGAs themselves in Section V. In Section VI, we give an overview on our FPGA related virtualization concept RC2F. Our prototype, which implements our concept with homogeneous and in their size flexible vFPGAs, is presented in detail in Section VII. The additional security extension $RC2F_{SEC}$ is introduced in Section VIII, followed by device utilization, vFPGA sizes and performance results of the simulation of our FPGA-Cloud in Section IX. Section X concludes and gives an outlook.

II. Related Work

The provisioning of reconfigurable hardware in data centers and cloud environments has gained more and more importance in the last years as shown by the overview from Kachris et al. [15]. Initially used mainly on the network infrastructure level, FPGAs are now also employed on the application level of data centers [12]. Typical use cases in this field are background accelerations of specific functions with static hardware designs. The FPGAs' special feature to reconfigure hardware at runtime is still used rather rarely. Examples are the anonymization of user requests [16] and increasing security [11] by outsourcing critical parts to attack-safe hardware implementations. In most cases, the FPGAs are not directly usable or configurable by the user, because the devices are, due to a missing provisioning or virtualization, hidden deeply in the data center.

The development of methods for the deployment of FPGA related projects in a cloud infrastructure is performed by Kulanov et al. in [17]. A comparable contribution with stronger focus on the transfer of applications into an FPGA grid for high performance computing is shown in [18]. The application focus on a single cloud service model with background acceleration of services using FPGAs. An approach, which places multiple user designs on a single FPGA, is introduced by Fahmy et al. [19], using tightly attached FPGAs to offload computationally intensive tasks. The FPGAs are partially reconfigurable and can hold up to four individual user designs. The approach was extended by Asiatici et al. in [20] with additional memory virtualization. A cloud integration model with network-attached FPGAs and multiple user designs on one FPGA was introduced by Weerasinghe et al. [21].

The term *virtualization* itself is used for a wide range of concepts as shown by Vaishnav et al. in [22]. An example for abstractions on the hardware description level is VirtualRC [23], which uses a uniform hardware / software interface to realize communication on different FPGA platforms. BORPH [24] provides a similar approach, employing a homogeneous UNIX interface for hardware and software. The FPGA paravirtualization pvFPGA [25], which integrates FPGA device drivers into a paravirtualized Xen virtual machine, presents a more sophisticated concept. A framework for the integration of reconfigurable hardware into cloud architecture is developed by Chen et al. [26] and Byma et al. [27]. The framework of Byma et al. allows user-specific acceleration cores on the reconfigurable hardware devices, which are accessible via an Ethernet connection. In [28] Chen et al. use FPGAs for processing network streams on virtualized FPGA resources similar to our approach. A virtualized execution runtime for FPGA accelerators in the cloud is shown by Asiatici et al. in [29]. They demonstrated a complete methodology and a resource management framework that allows a dynamic mapping of the FPGA resources in a simple cloud environment.

Approaches more closely related to the *context-save-and-restore* mechanism required by our migration concept can be found in the field of bitstream readback, manipulation and hardware preemption. In ReconOS [30], hardware task preemption is used to capture and restore the states of all flip-flops and block RAMs on a Virtex-6 to allow multitasking with hardware threads. In combination with homogeneous bitstreams for different physical vFPGA positions, methods like relocation of designs as shown in [31], provide an opportunity for an efficient context migration of virtualized FPGAs. A preemption of the reconfiguration process itself is shown by Rossi et al. in [32].

The outlined systems virtualize FPGAs and makes them easily available in the cloud. But not every user can utilize such a service, because their sensitive data is at risk in a data center. Security audits are well established in traditional systems, but new cloud environments provide new challenges [33, 34]. In [35] the idea of securing FPGAs in the cloud is outlined, but no prototype realized or protocol described. A secure cloud featuring FPGAs was proposed in [16] relying on a third party, called trusted authority, to establish any trust in the hardware in the cloud. In [36] a simple public key based systems was implemented, however, their protocols fail to protect against, e.g., replay attacks. But none of these proposals virtualizes the FPGA to increase their flexibility and utilization.

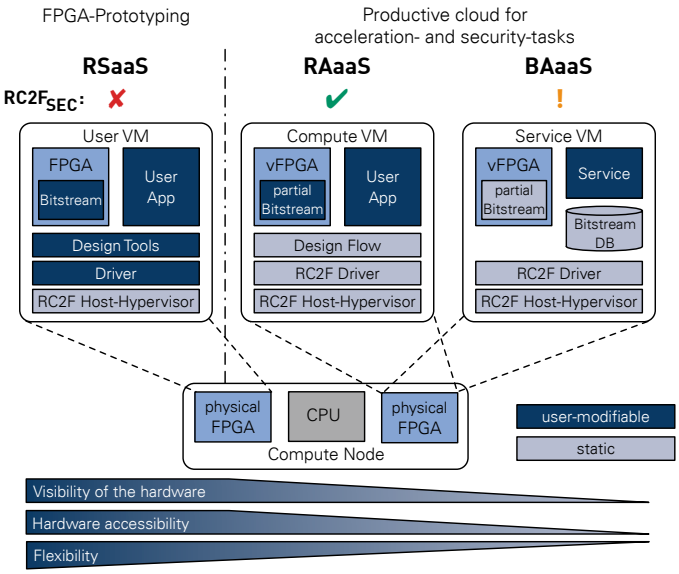


Figure 1. The three service models provided in our cloud environment. In the RSaaS model, users can allocate full physical FPGAs. The RAaaS and BAaaS model allow concurrent user designs on a single physical FPGA.

III. Possibilities and Requirements for FPGAs in the Cloud

The overall motivation is to build a system providing the FPGA for a wide range of service providers with various requirements. The particularity hereby is that we have a data

center provider with physical FPGAs, a cloud provider offering a virtual infrastructure and a service provider who offers only a background acceleration, which requires a virtualized FPGA as shown in Section VII. In the following we introduce three key service perspectives as shown in Figure 1. The figure gives also an overview on modifiable and fixed components for each of the service models and shows also the different levels of visibility, accessibility, flexibility and security.

A. Reconfigurable Silicon as a Service – RSaaS

This model provides full access to the reconfigurable resource and is primarily intended for a *cloud provider* to develop special acceleration cores without the use of a virtualized FPGA or with a dedicated secured access to the cloud. A cloud provider can allocate a full physical FPGA from the data center operator to implement the hardware of their choice. The FPGA is forwarded and passed through to a VM by the management environment. This model allows developers to reconfigure the full physical FPGA, thus, the $RC2F_{SEC}$ extension cannot protect the users' data. It also opens new attack vectors that do not exist in current cloud environments and so this model should be limited to cloud providers. The concept can be compared to bare-metal cloud services and is related to the traditional cloud service models Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

B. Reconfigurable Accelerators as a Service – RAaaS

A model with less freedom for the developer (*service provider*) and typically used by service providers is the Reconfigurable Accelerators as a Service (RAaaS) model. Only vFPGAs of different sizes are visible, allocatable and usable. The model allows the development of hardware designs, which can be used for background acceleration of a specific service and the communication is performed via the framework introduced in Section VII. Such restrictions have the advantage that the $RC2F_{SEC}$ extension can be used, which significantly increases the security of the system compared to the RSaaS model. The RAaaS model can be compared to the PaaS model.

C. Background Acceleration as a Service – BAaaS

The third model is suitable for applications and services using background acceleration running in common data centers. The vFPGA is not visible or accessible by the service *users*. Instead, services are using vFPGAs in the background to accelerate specific tasks. The pre-build configuration files and host applications are used by the cloud service provider. Resource allocation and vFPGAs reconfiguration occurs in the background using the RC3E resource management system. Because this model provides concrete service applications to the user, it is similar to the PaaS model. Especially the BAaaS service model demands resource pooling and a rapid elasticity for typical workloads. FPGAs allow a higher flexibility than virtual machines due to faster booting times. From a security perspective, this model is similar to current cloud environments, because of the limited reconfigurability of the FPGA. The $RC2F_{SEC}$ extension cannot be used in this model. In Section IX we demonstrate the cloud's performance with a workload using our background acceleration service model.

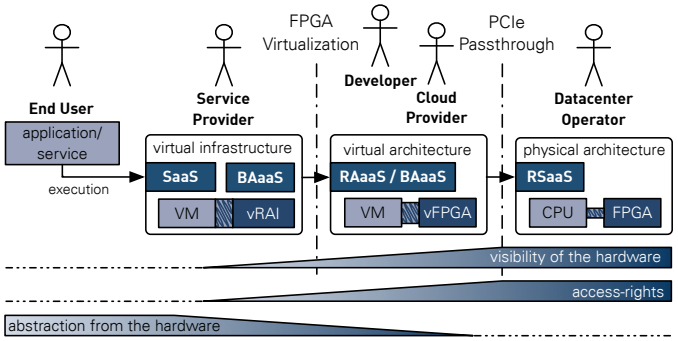


Figure 2. Involved stakeholders and the visibility of resources in a flexible environment. Background acceleration is primarily used in systems were service provider and datacenter operator are the same.

D. Chaining it all together: RSaaS – RAaaS – BAaaS

Figure 2 illustrates how a physical FPGA is abstracted by multiple layers into a transparent background accelerator. First, the datacenter operator makes the FPGAs available to the cloud providers (RSaaS). Their developers implement applications for the vFPGAs (RAaaS) and package them in Virtual Reconfigurable Acceleration Images (vRAIs), which are described in Section IV-C. Such a vRAI is used by the service provider as a black box in the BAaaS model (see Section IV-C and Section VII-F). At this point a virtual FPGA infrastructure is provided, which can be used to accelerate services executed by end users. Combined with the classic Software as a Service (SaaS) model, this allows for a seamless integration of vFPGAs to accelerate the service, reduce energy consumption and thus, saving operating costs. At this highest level of abstraction, the FPGA is transparent to the end user.

IV. Reconfigurable Common Cloud Computing Environment – RC3E

In this section, we will present the **R**econfigurable **C**ommon **C**loud **C**omputing **E**nvironment – **RC3E** – and explain the components depicted in Figure 3 in detail. In contrast to other cloud architectures with FPGA integration presented in Section II, the RC3E environment is designed especially for an integration of virtualized FPGA resources and the service models described in Section III. The system is a proof-of-concept to study different approaches for the virtualization and the flexible integration of reconfigurable hardware into a cloud management system. The evaluation results will be used for future integrations of specific RC3E components into a cloud management system such as OpenStack [37].

A. Overall System Architecture

The overall system design is a distributed three tier client-server architecture to provide a high degree of scalability and flexibility. RC3E offers three access possibilities to use and administer the RC3E system. The most common way is a *login shell* either on a local computer with our RC3E client or via secure shell login to the remote login server. Additionally, it is possible to connect a web frontend (see Section IV-B) to the core system's API.

The RC3E core system running on the management node, which itself is a three tier architecture, orchestrating the connected clients and all registered compute nodes. It uses

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

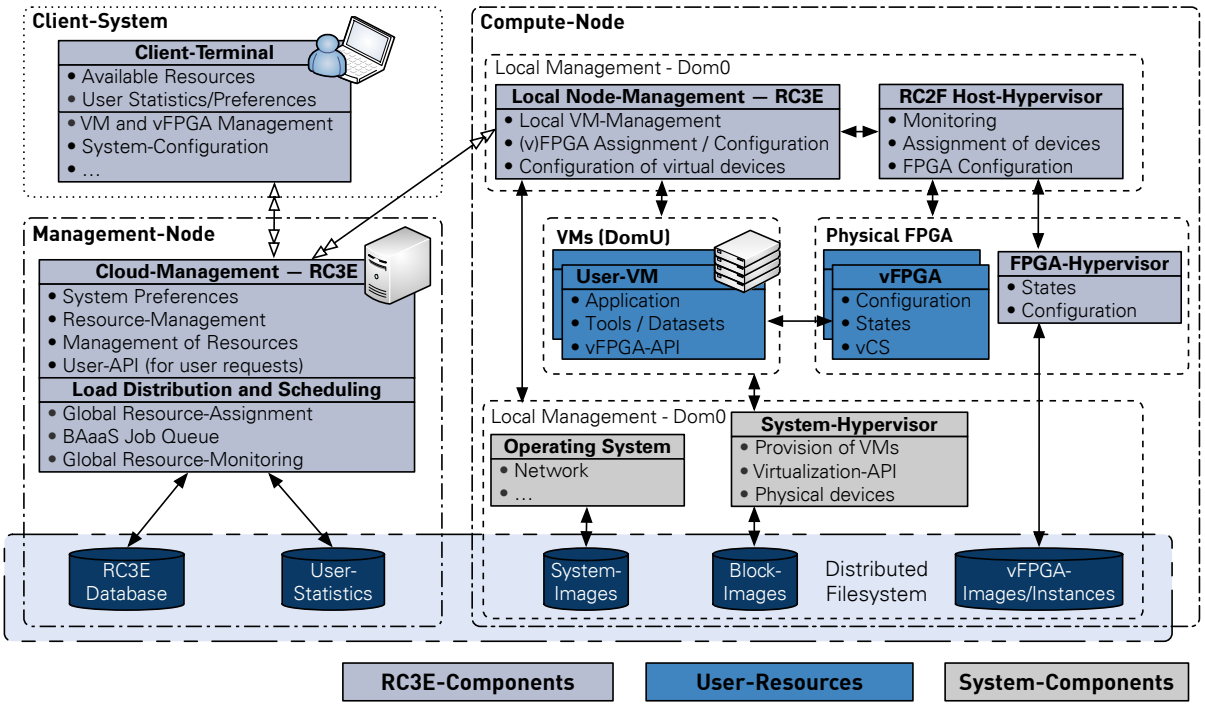


Figure 3. Architecture of the resource management and hypervisor RC3E consisting of core system (management, monitoring and job scheduling) and compute node providing VMs and vFPGAs.

a centralized database to store all required information and manages a distributed file system, which is shared between all compute nodes and the management nodes.

B. Web-based User Interaction and Database Backend

In a cloud environment it is common that the majority of users does not have administrative access to the system. A web-based frontend allows these users a fast and comfortable way to reserve FPGA slices on the server, upload and run their designs. The `Django` framework was used as a foundation of such a web frontend. All data required for the frontend is stored in a `MySQL` database, which interacts directly with `Django`'s web-server. The *RC3E* tools discussed in the other sections have to be present on the same machine as the web-UI in current implementations. Separating the cloud systems control instance from the web-server is desirable and expected to be done with a reasonable effort in the future.

In Figure 5 the modeled entities and their reference relations are shown. Data types and classes provided by `Django` itself are printed in italics for distinction. Attributes that only serve framework-internal purposes and are added by `Django` automatically have been omitted for clarity.

The models focus points and their synergies will be examined in the following. Words in bold typeface thereby correspond with the entities in the model.

To avoid ambiguity, the term *user* refers to any human interacting with the system, while *administrators* refers to users with the privileges to access and modify the system's internal state. Persons without such privileges will be called *consumers*. `Django`'s integrated user group and permission management system is used to reflect user categories and facilitate access control across the web-UI.

It was decided to use a fine-grained modeling approach to retain flexibility and changeability in an attempt to create a future-proof software base for future development iterations. This also includes the avoidance of unmanaged data redundancy by preferred usage of foreign references.

Entities are represented on the database level as separate tables with each of the entity's attributes as a table column and each instance of the entity as a table row, containing the actual attribute values within the respective cells.

The modeling is heavily influenced by the operation principles of foreign key references in SQL-based data storage systems. Thus, in situations where two entities *A* and *B* form a *[A]1:n[B]*-relationship, the foreign key has been placed on the *B* side referencing *A* to avoid creating a separate associative entity each time such a relation shows up. In *[A]m:n[B]* scenarios such a separate entity can not be avoided though. The UML-style *B contains A* symbolization should therefore be read as *B contains a reference to A*. Foreign key references are set up to execute cascading deletions and modifications, since the entity containing the reference would enter an invalid state if not deleted/modified as well.

1) Nodes, FPGAs and Regions: A **node** represents the physical cloud server in which **FPGA**s are installed. Each node is named and identified by an unique IP-address. An optional comment allows the node's administrator to easily convey additional information to the consumer aside from the installed FPGAs.

Installed FPGAs are initialized by the *RC3E* management system, which provides **PCI-addresses** for the node and the device itself. The latter are queried during the registration of the FPGA with the web-frontend and associated database entries are created automatically.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

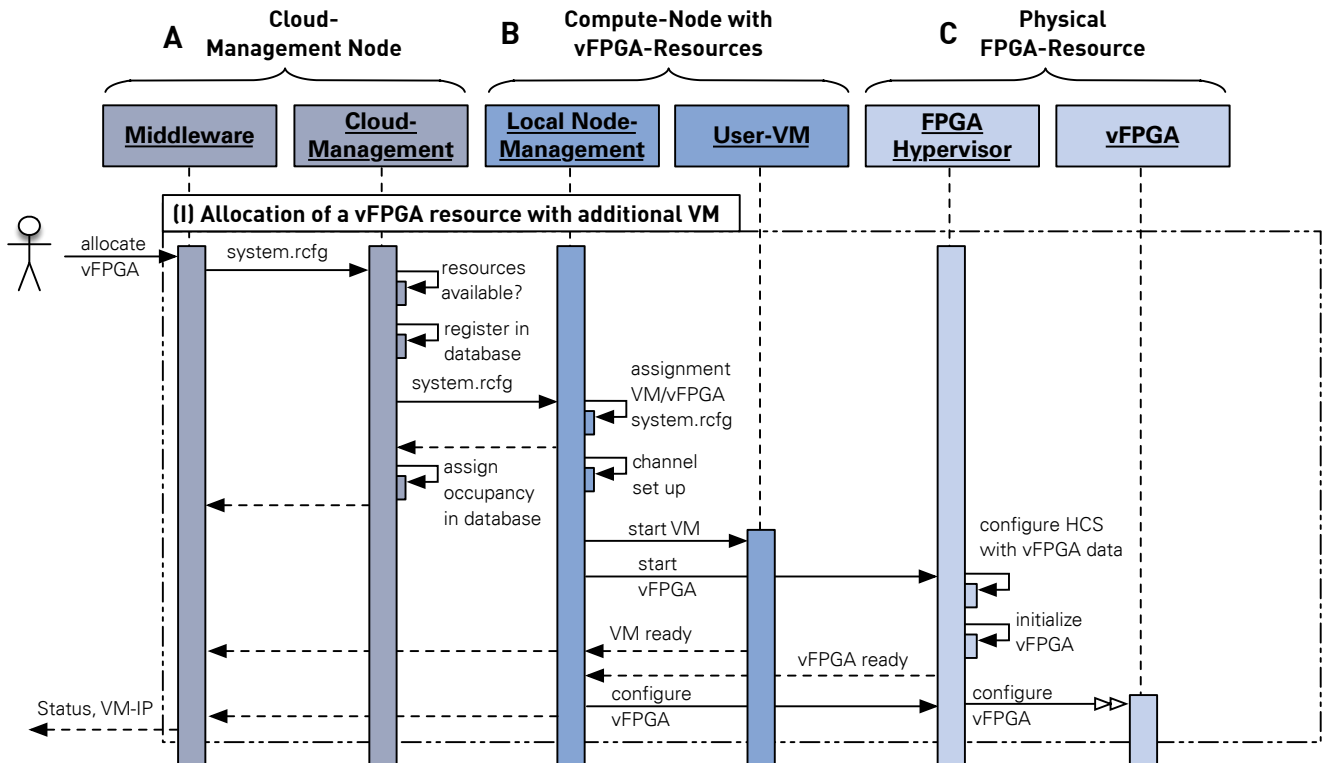


Figure 4. Sequence diagram showing interaction of levels in the RC3E system via cloud management nodes, the compute node with a (free) vFPGA resource up to the physical FPGA for three exemplary scenarios.

To offer a uniform description of an FPGA, the **FPGA model** has been introduced as an abstraction of generic and structural information. The **producer** entity has been externalized with the prospect of providing additional information about it in future implementations. An FPGA model references a **region type** and holds the amount of regions an FPGA of this model has. With this approach only homogeneous FPGA architectures can be modeled. For heterogeneous architectures the region type and -count would have to be externalized and act as associative entity between the FPGA model and the region type.

One or multiple **region** instances are created alongside with an FPGA instance, the amount depending on the FPGAs region count. Its region type is determined by the associated FPGA model and an region index is determined. These indices are unique per FPGA, 0-based and continuous, with the purpose of identifying regions on an FPGA and determine whether they are neighbors and can therefore be reserved together. For programming purposes the *RC3E*-system provides a file path to a memory device, which is also stored.

2) Reservations and Virtual FPGAs: The most common interactions of customers with the system are the reservation of vFPGAs and the programming of such. For the first step, the customer provides points in time for the start and end of the reservation period and selects a region type suitable for his use case along with the required amount of consecutive regions. The database-backend will then be queried for matching FPGA regions. Already existing reservations are taken into account when selecting a sufficient amount of consecutive FPGA regions to reserve. On success, a new vFPGA instance is created, alongside with a **region reservation** for each affected region. The latter is an association entity to facilitate the *[vFPGA]n:m[Region]*-relation. While region reservation database entries are removed after the reservation period has passed, vFPGA entries are retained for bookkeeping purposes.

3) Programming the Virtual FPGA: The administrators provide information about the installed **programmer**s and the programming **script** for the available FPGA models. Both entities are used to determine which programmer-FPGA model-combinations are supported and thus, which programmers are offered to the customer for usage with his reserved vFPGA. Upon programming, the script's template gets parsed and placeholders within it matched against the available **device variables** and **runtime variables**. If a match occurs, it is replaced by the variables' value. Device variables are bound to specific FPGAs and are set by the administrator while runtime variables may be `python` expressions or fixed values and will be evaluated at the point of replacement. Within the reservation period the user may upload a bitfile, which will then be passed on to the programmer alongside with the appropriate script and variables. In case of a reservation spanning multiple regions, the memory device path of the region with the lowest index is the one used by the programmer for the whole reserved section. Uploaded bitfiles are currently not stored in the database backend.

C. Description of vFPGAs (RCFG and vRAI)

All necessary information for the execution of a background accelerator is combined in a so called vRAI. The vRAI can be delivered as fully encapsulated accelerators to

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

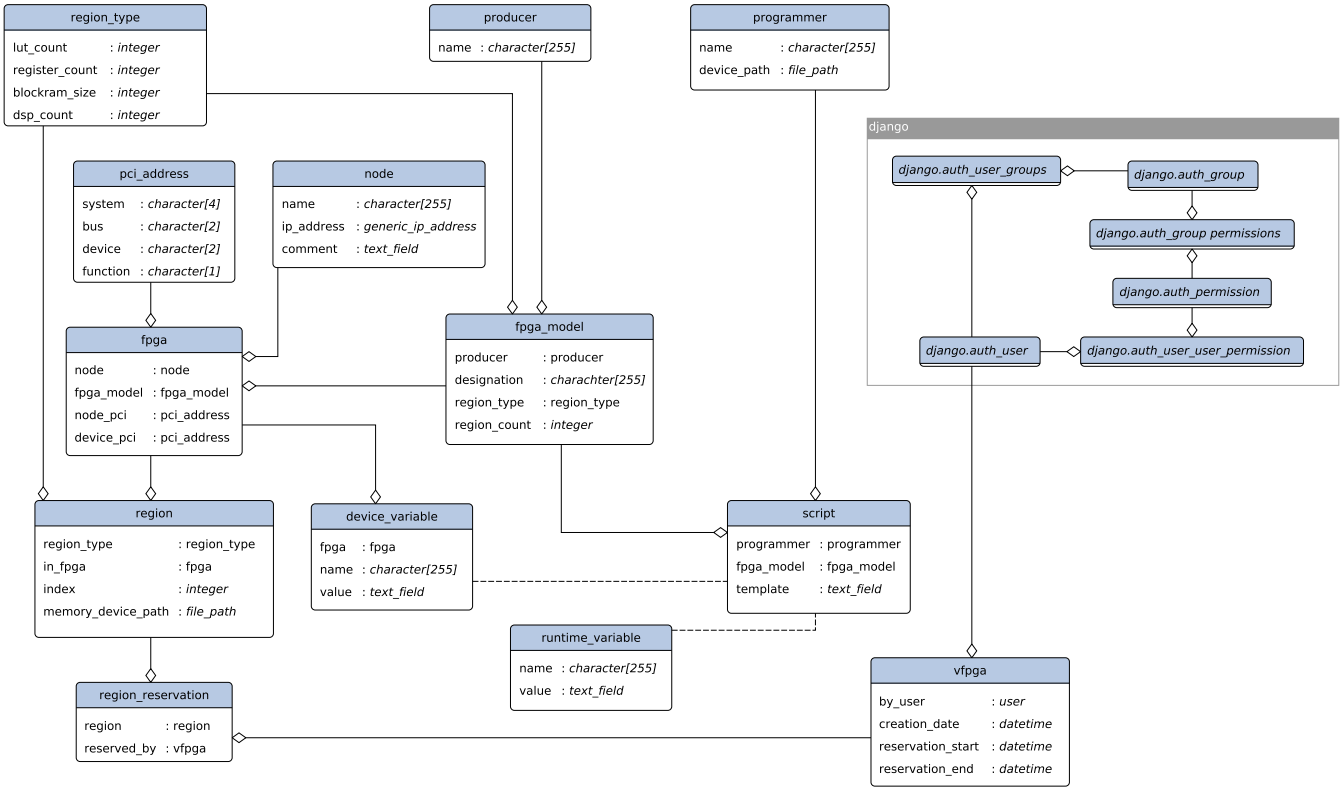


Figure 5. Overview over the entities involved in modeling a database backend. Italic text represents primitives provided by the `Django` framework.

the higher-level cloud service developers. From the point of view of the provider of a service, there is the requirement to process a request in the form of a function call as compact, safe or energy-efficient as possible, without having any knowledge of the physical hardware in the background. The execution of a vRAI requires allocation of a vFPGA, which fulfills all requirements described in the **R**econfigurable (Device) **Con**fig**u**ration (RCFG). In order to allocate and execute an accelerator from a VM, several components are required within the vRAI package (see Figure 6):

- The required vFPGA-Images for all possible vFPGA-Slots (necessary for a migration of a vFPGA-Instance).
- The RCFG file describing the required vFPGA suitable for the vFPGA image.
- The host application for initialization and interaction with the vFPGA-Instance, which is embedded directly into the user's offloading service (BAaaS).
- Virtual Context Bit Mask (VCBM) to read the relevant bits within the vFPGA instances that identify the current state (see Section VII-E).

Since different RCFGs are required depending on the different service models, these are outlined below and explained accordingly. Figure 7 shows an exemplary RCFG, which describes a complete physical FPGA in the model RSaaS `service='rs'` with the name `name='fpga0'` gets and in the VM instance `vm='vm1-hvm'` via hardware virtualization and PCI passthrough at a certain address in the PCI tree `pci ='01:00.0'` is displayed. The virtual network address is sent to the system via `vif='10.0.0.43'`. The VM must have its own configuration file depending on the virtualization, and the embedding of the VM configuration in the RCFG for the FPGA is also possible. Since different FPGAs are to be provided in this model, there is a corresponding entry with the name of the FPGA board `board='vc707'`. The configuration of the FPGA is done using the JTAG interface `config= 'jtag'`, where an initial design is additionally specified: `design ='led.bit'`. Using a RCFG file, the RSaaS model can allocate only one FPGA and its associated VM, otherwise the requested system may become too complex and it may not necessarily be mapped to the physical hardware resources.

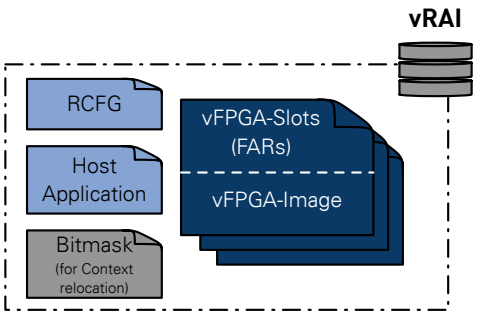


Figure 6. Virtual Reconfigurable Acceleration Image (vRAI) package with all the files required to run a vFPGA-Instance, such as vFPGA-Image (partial bitstreams), RCFG, host program and optional bitstream masks for the bitstream relocation.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
service = 'rs'                  #Service Model RSaaS
name = 'fpga0'                  #FPGA-Instance Name
vm = 'vm1-hvm'                  #VM-Instance Name

board = 'vc707'                 #FPGA-Board
vif = 'ip=10.0.0.43'            #FPGA-IP
vpci = '01:00.0'                #PCI Node in VM-Instance
design = 'led.bit'              #Initial Design
config = 'jtag'                 #Configuration Method
```

Figure 7. Configuration file for the allocation of a physical FPGA in the Service Model RSaaS.

More complex is the description of the vFPGAs in the model RAaaS `service=ra` as shown in Figure 8. In addition to the already known parameters, in this example two vFPGAs with different number of vFPGA slots `size=[2, 1]` are allocated via `vfpga=[2]`, where both are passed to the same VM `vm=['vm1-pvm']`. The number of frontend interfaces is determined by `frontends=[2, 1]` for each vFPGA, where the number must be less than or equal to the number of vFPGA slots. At this point, the cloud management must try to map the desired virtual system to a physical system, where in the model RAaaS additionally the position of the vFPGA on the physical FPGA can be specified by the field `loc=[0,2]`. Via `debug=['csp']` in the resource management model RAaaS it is communicated which debug/tracing interface is to be additionally instantiated. The capacity of the external DDR memory can also be specified (`memory=[2000,1000]`), as well as the desired state of vFPGAs `boot=['paused']`. The location of the values within the lists, such as `size`, `loc`, `vif`, or `design`, decides how to map the entries. Furthermore, if there is only one entry in a list, such as `key`, it will be applied equally to all vFPGAs. If no clear assignment is possible or if this is not permitted, an error message is output.

```
service = 'ra'                  #Service Model RAaaS
name = ['vfpga-bsmc']           #vFPGA/User Design Name
vm = ['vm1-pvm']                #VM-Instance Name

vfpga = [2]                     #Number of vFPGAs
size = [2, 1]                   #vFPGA-Slots
frontends = [2, 1]              #Frontend-Interfaces
loc = [0,2]                     #vFPGA-Slot on device
memory = [2000,1000]            #DDR-Memory Size in MByte
vif = ['ip=10.0.0.42', ...]     #vFPGA-IPs

boot= ['paused']                #Initial vFPGA-State
design = ['bsmc-2.bit', ...]    #Initial Designs
```

Figure 8. Configuration file for the allocation of a vFPGA-Cluster in the Service Model RAaaS.

In the model BAaaS, the actual user has no knowledge of the vFPGA resources. The RCFG file, which is stored together with all the required vFPGA images in the vRAI, is reduced. For example, as shown in Figure 9, there are no locations of the vFPGA slots (`loc`) or information about the debug/tracing interface (`debug`) required. The RCFGs must be checked by the resource manager for the rights of the users within the service model before the global allocation of the appropriate resource is first performed and assigned to the user. The concrete processing of the content then happens within the Dom0 of the assigned node as introduced in Section IV-B1.

```
service = 'ba'                  #Service Model BAaaS
name = ['vfpga-kmeans']         #vFPGA/User Design Name
vm = ['vm1-pvm']                #VM-Instance Name

vfpga = [1]                     #Number of vFPGAs
size = [4]                      #vFPGA-Slots
frontends = [2]                 #Frontend-Interfaces
memory = [4000]                 #DDR-Memory Size
vif = ['ip=10.0.0.151']         #vFPGA-IP

key = ['AAAABC1yc2 ... BuHNE']  #User AES-Key
boot= ['booting']               #Initial vFPGA-State
design = ['kmeans-quad.vrai']   #Initial Design
```

Figure 9. Configuration file for the allocation of a single vFPGA in the service model BAaaS

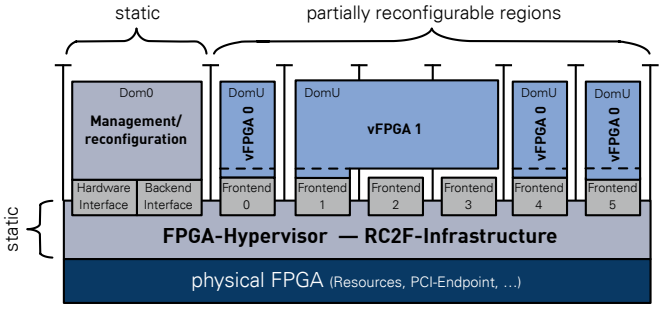


Figure 10. Paravirtualization concept used in RC2F to provide virtual FPGAs (vFPGAs) using partial reconfiguration. vFPGAs can be combined to group larger regions and thereby provide more resources.

V. Definitions for FPGA virtualization in the cloud-context

In order to establish a common name for the following chapters with regard to the virtualized FPGAs, the necessary terms are defined in order to better distinguish the vFPGAs (see Definition 1) according to their life cycle based on the requirements analysis in this chapter to be able to. The terms are based on those of system virtualization after [38].

Definition 1: vFPGA *A virtual FPGA (vFPGA) is located within a physical FPGA on one or more vFPGA slots (see Definition 2). A vFPGA is perceived by the user as a stand-alone resource with a dynamic number of hardware resources (slices, LUTs, registers, etc.).*

Definition 2: vFPGA-Slots *A vFPGA is mapped to individual physical regions with a fixed number of hardware resources, and thus fixed size within the physical FPGA, called vFPGA-Slots.*

Definition 3: vFPGA-Design *The vFPGA-Design is the hardware design/the user's hardware design, which is placed and wired from a netlist (RTL level) within a vFPGA with its frontend interfaces.*

Definition 4: vFPGA-Instance *A vFPGA-Image (see Definition 5) within a vFPGAs that is directly associated with a user and can contain user-specific data (context) is called a vFPGA-Instance and can be detached from vFPGA-Slots (see Definition 2).*

Definition 5: vFPGA-Image *A partial bitstream, which forms the basis for a vFPGA-Instance, is called vFPGA-Image, the specific vFPGA slots is assigned.*

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

In addition to the definitions just made, which relate to the specific vFPGAs and their life cycle, furthermore, the different hypervisors in the overall system are to be differentiated and defined. The term *hypervisor*, is defined as a system for managing and allocating guest-to-host resources, forms the basis for the following definitions.

Definition 6: System-Hypervisor *The System-Hypervisor corresponds to the classic hypervisor (VMM), which provides the VMs within the system virtualization on the host system.*

Definition 7: RC2F Host-Hypervisor *The management structure for the vFPGAs on their host system is called RC2F Host-Hypervisor, or just Host-Hypervisor.*

Definition 8: FPGA-Hypervisor *The FPGA-Hypervisor is the management structure on the FPGA that monitors the accesses of the vFPGAs within the physical FPGA.*

VI. FPGA Virtualization

As the cloud itself is based on virtualization, the integration of FPGAs requires a profound virtualization of the reconfigurable devices in order to provide the vFPGAs as good as other resources in the cloud. Furthermore, it is necessary to abstract from the underlying physical hardware.

A. Requirements for Virtual FPGAs in a Cloud Environment

As discussed in Section II, the term *virtualization* is used for a wide range of concepts. The application areas of FPGAs in clouds require a direct use of the FPGA resources to be efficient. Thus, an abstraction from the physical FPGA infrastructure is only possible in size and location. Our approach is related to traditional system virtualization with VMs that corresponds to a Type-1 bare-metal virtualization with use of a hypervisor [39]. This kind of virtualization is designed for the efficient utilization of the physical hardware with multiple users. Therefore, it is necessary to adapt the required FPGA resources closely to the requirements of the users' hardware design capsuled by vFPGAs. By this, an efficient utilization of the physical hardware with multiple concurrent vFPGAs on the same hardware can be achieved.

Furthermore, the vFPGA has to appear as a fully usable physical FPGA with separated interfaces and its own infrastructure management like clocking and resetting. For an efficient cloud architecture, which requires elasticity [9], it is necessary to migrate vFPGAs with their complete context (registers and BlockRAM), which requires to enclose a complete state management of the vFPGA as described in [6] and [1]. An extraction of internal DSP registers is not supported in recent Xilinx FPGAs and must be considered in the design.

One of the first virtualized systems was the IBM Virtual Machine Facility/370 (VM/370) [40] in 1960 with a first abstraction and partitioning in host and guest. Nowadays a common definition is that

"*Virtualization provides a way of relaxing the forgoing constraints and increasing flexibility. When a system device (...), is virtualized, its interface and all resources visible through the interface are mapped onto the interface and resources of a real system actually implementing it.*" [38, p. 3]

The two classic approaches are either the use of a VMM, a small operating system controlling the guest system's access to the hardware, or multiple guest systems embedded into a standard host operating system [41]:

- Type 1: Bare metal (VMM or Hypervisor)
- Type 2: Host operating system

Another distinction can be made on the level of code execution and driver access, where the relevant approaches are [38]:

- Hardware virtualization (full virtualization)
- Paravirtualization
- Hardware-assisted virtualization

An interesting starting point for FPGA virtualization is especially the VMM concept with paravirtualization in which the interfaces to the VMs are similar to those of the underlying hardware. The VM interfaces are modified to reduce the time spent on performing operations, which are substantially more difficult to run in a virtualized than in a non-virtualized environment. This kind of paravirtualized system is introduced in Section VII-C. The unprivileged guests (DomU) run on a hypervisor, which forwards calls from frontend driver to the backend driver of the management VM (Dom0).

B. FPGA Virtualization Approach

We decided to virtualize the FPGA similar to a paravirtualized system VM executed by a hypervisor to provide access to the interfaces. Figure 10 shows an FPGA virtualization inspired by the paravirtualization introduced before. The virtualization is limited to the interfaces and the designs inside the reconfigurable regions, which constitute the actual vFPGAs as unprivileged Domain (DomU). Each vFPGA design is generated using the traditional design flow with predefined regions for dynamic partial reconfiguration [14] and static interfaces. The vFPGAs can have different sizes (Figure 10) and operate completely independent from each other. The infrastructure encapsulating the vFPGAs has to be located in the static region corresponding to a privileged domain (Dom0) or hypervisor.

The interface providing access to the vFPGAs is a so-called *frontend interface*, which is connected inside the hypervisor to the *backend interface* in the static FPGA region. There, all frontends are mapped to the static PCIe-Endpoint and the on-board memory controller inside the Dom0, which also manages the states of the vFPGAs.

VII. FPGA Prototype RC2F

Our prototype RC2F introduced in [4] provides multiple concurrent vFPGAs allocated by different users on a single physical FPGA. The main part of the FPGA frame(work) consists of a hypervisor managing configuration and user cores, as well as monitoring of status information. The controller's memory space is accessible from the host through an API. Input- and output-FIFOs are providing high throughput for streaming applications. The vFPGAs appear to the user as individual devices inside the System VM on the host.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

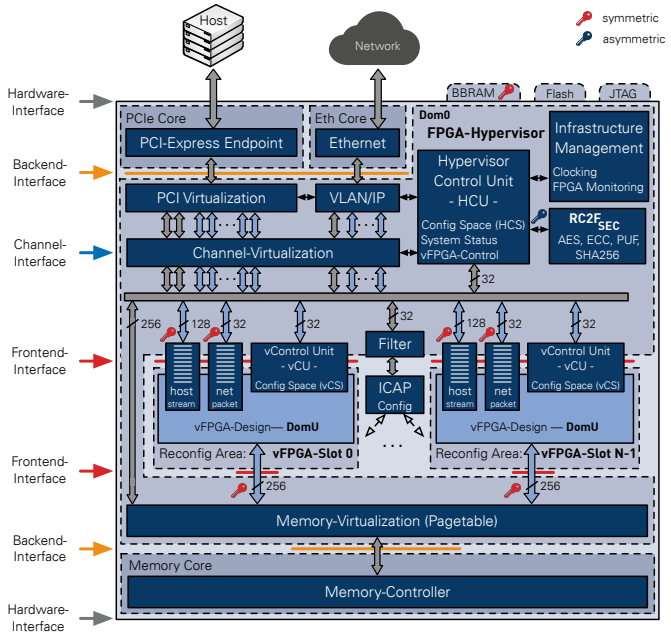


Figure 11. Virtualization frame RC2F with hypervisor, I/O components and partial reconfigurable areas housing the vFPGAs. The vFPGAs have access to the host using PCIe (FIFO interface and config space), to the Cloud network using Ethernet and the virtualized DDR3 memory.

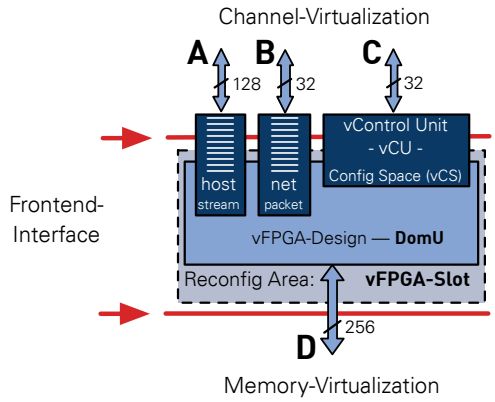


Figure 12. Architecture of a RC2F-vFPGA with (C) the local vFPGA Control Unit (vCU) containing the Virtual Control Space (vCS). The data lines for the memory interface (D) and the two FIFOs are each available for input (A) and output (B) streams.

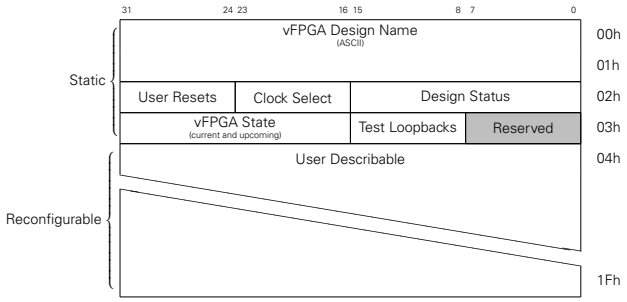


Figure 13. Register and memory interface for the management of vFPGAs accessible by the user VM (rc2f_cs).

A. *System Architecture*

The physical FPGAs are located inside a host system and are accessible via PCIe. On both hardware components (host and FPGA), there are hypervisors managing access, assignment and configuration of the (v)FPGAs. Based on our concept, we transform the FPGAs into vFPGAs with an additional state management and a static frontend interface as shown in Figure 10. Our architecture, designed to provide the vFPGAs, is shown in Figure 11. The hypervisors manage the on-chip communication between backend and frontend interfaces for PCIe (Our prototype uses a PCIe-Core from Xillybus for DMA access [42]), Ethernet and a DDR3 RAM. The RAM is virtualized using page tables, managed by the host hypervisor, which also manages the vFPGA states we introduced in [6]. The number of frontends and their locations are defined by the physical FPGA architecture as shown in Figure 16. The *Hypervisor Control Unit* manages the ICAP controller and the *vControl* units, which maintain and monitor the vFPGAs.

1) vFPGAs: To exchange large amounts of data between the host (VM) and the vFPGAs a FIFO interface is used. To exchange state and control information the vFPGAs can be controlled by the user via a memory interface as shown in Figure 13. The memory is mainly intended for simple transfers and configuration tasks like resets, state management (pause, run, readback, migrate) and the selection of a vFPGA system clock. In addition to these static fields, there is also a user-describable memory region, which can be used as virtual I/O. The communication using Ethernet is also provided but out of the scope of this paper.

2) Components of the RC2F infrastructure: The RC2F infrastructure is exemplarily implemented within the static area with the components as shown in Figure 11. For the infrastructure, as shown in Figure 16, both the right side of the physical FPGA and the lower clock region are provided. The constant components of the static infrastructure within the RC2F infrastructure are:

FPGA-Hypervisor: At the heart of the implementation is the FPGA hypervisor, which provides the frontends to the vFPGAs as shown in Figure 11. Essential components are the configuration memory of the FPGA hypervisor explained in Figure 14, which transmits all control commands and signals to the FPGA, and the ICAP controller for reconfiguring the vFPGA slots and to read out a partially reconfigurable vFPGA instance for migration. The memories are built from components of the Pile of Cores (PoC) library [43] and constructed as shown in Figure 14. The internal clock rate (system clock) of the FPGA hypervisor and device virtualization is 250 MHz. To decouple the FPGA hypervisor from the internal logic of the vFPGAs as well as the I/O components, there are cross-clocking FIFOs at the interfaces between the clock domains.

PCIe-Controller: The PCIe controller is Xilinx's provided Intellectual Property Core (IP-Core) *7 Series FPGAs Integrated Block for PCI Express v3.3* [44] with a Xillybus controller [42], which provides both FIFO and memory interfaces on the FPGA, as well as a driver within the host hypervisor.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

DDR3-Controller and Memory-Virtualization: The used DDR3 controller is the IP Core *Xilinx MIG V1.4* [45], which is the hardware endpoint to the backend interface as introduced in Section VI-B and illustrated in Figure 11. The resulting storage virtualization managed by the host hypervisor organizes the specific translation from the virtual to the physical addresses, thus, separating the user areas in the memory from each other.

Ethernet-Controller: The Ethernet controller used is based on the IP Core *LogiCORE IP Tri-Mode Ethernet MAC v5.2* [46], which is an interface on the Media-Access-Control (MAC) layer of Open Systems Interconnection (OSI) Reference Model [47] offers. Based on this, parts of the PoC library [43] are used to implement the interfaces to the vFPGAs.

In addition to the previously discussed components of the RC2F infrastructure, additional components are required whose hardware resources depend on the number of physical vFPGA-Slots. These components are also in the static region:

Device-Virtualization: Device-Virtualization provides the concurrent communication channels for the vFPGAs. The realization of the PCIe-Virtualization is done by means of the Xillybus-Controller [42] provided components. The provided FIFOs are passed on to the vFPGAs and decoupled (cross-clocking) to allow different clock domains for the FPGA hypervisor and the vFPGA design. Memory virtualization requires one page table per user. The prototypical implementation uses page sizes of 8 MByte.

vFPGA-Frontends: The frontends are implemented as outlined in Figure 12 and Figure 15. The configuration memories are constructed according to Figure 13 and consist on the one hand of a part located in the static area of the FPGA and on the other hand of a user area, which can be used freely. In addition to the stores, the states of each vFPGAs are managed as outlined in Section VII-E.

B. Configuration of the FPGA Hypervisor

The tasks of the FPGA hypervisor are the management of its local vFPGAs and their encapsulation, the state management, as well as the reconfiguration using the ICAP. The interaction between host and FPGA hypervisor is based on the configuration memory shown in Figure 14, which includes configuration of the FPGA hypervisor (system status, reconfiguration data and status) and the administration of the vFPGAs. Other important vFPGA-related entries are an AES-key for encryption of the vFPGA-bitsteam and the allocated vFPGA region(s) for additional validation during reconfiguration.

C. The Role of the Host-Hypervisor

Our virtualization concept on the host-system includes passing through the vFPGAs' FIFO channels and the configuration memories from the host-hypervisor to the user VMs (DomU) and the FPGA hypervisor memory to the management VM (Dom0). The overall system architecture is shown in Figure 15. The frontend FIFOs and the FPGA memories are mapped to device files inside the host hypervisor. There, the system forwards the user devices to the assigned VM using inter-domain communication based on vChan from Zhang et al. [48] in our Xen virtualized environment, similar to the FPGA device virtualization pvFPGA [25].

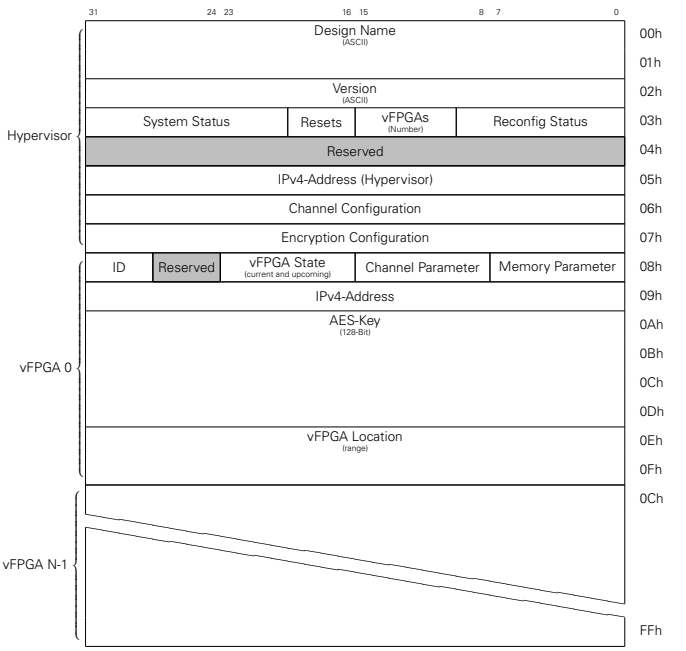


Figure 14. Register and memory interface for the management of the FPGA hypervisor accessible by the host hypervisor (rc2f_gcs).

The management VM thereby accesses the FPGA hypervisor's configuration memory and the ICAP on the FPGA via a dedicated FIFO interface for the configuration stream (read and write). Thus, only the hypervisors can configure the vFPGA regions on the physical FPGA whereby a sufficient level of security can be guaranteed.

D. Mapping vFPGAs onto physical FPGAs

In our example we use six frontends on a Xilinx Virtex-7. Depending on the resources required, the utilization of up to six different-sized vFPGAs is possible with the same static without reprogramming. If one of the vFPGAs covers more than one region, only one frontend connection is used as shown in Figure 10. Among the vFPGAs, the partition pins (PP) between the static and the reconfigurable regions are placed with identical column offset as shown in Figure 16. The regions forming the vFPGAs are not free from static routes as for example the region vFPGA 5 shows.

To reduce migration times, all components, which hold the context of the current vFPGA design as registers, FIFOs or BlockRAM, are placed at the same positions inside each vFPGA. Therefore, it is necessary that all of these positions exist in each region. Hardmacros like PCIe-Endpoints or parts of the FPGA infrastructure interrupt the homogeneous structures. Thus, we establish homogeneous vFPGAs, which are identical among each other by excluding these areas in all vFPGAs as shown in Figure 16. The advantage of this approach is that only one mask file is necessary to extract the content of the different vFPGAs. Furthermore, it allows the provision of almost identical vFPGAs. Figure 17 shows the breakdown of the FPGA resources to the three different areas: static infrastructure, partial reconfigurable vFPGAs and unusable due to the homogeneity.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

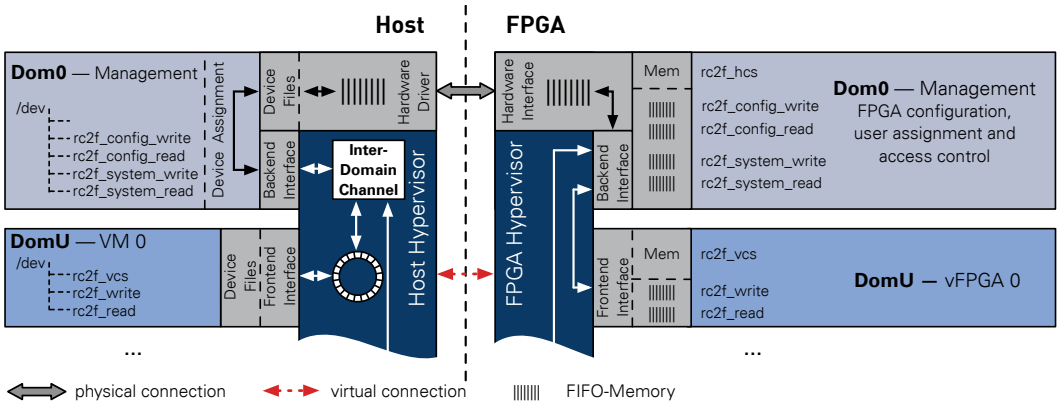


Figure 15. System architecture on the hypervisor level of the host system. FIFOs (rc2f_write, rc2f_read) and configuration memories (rc2f_cs) are displayed in the different host memories.

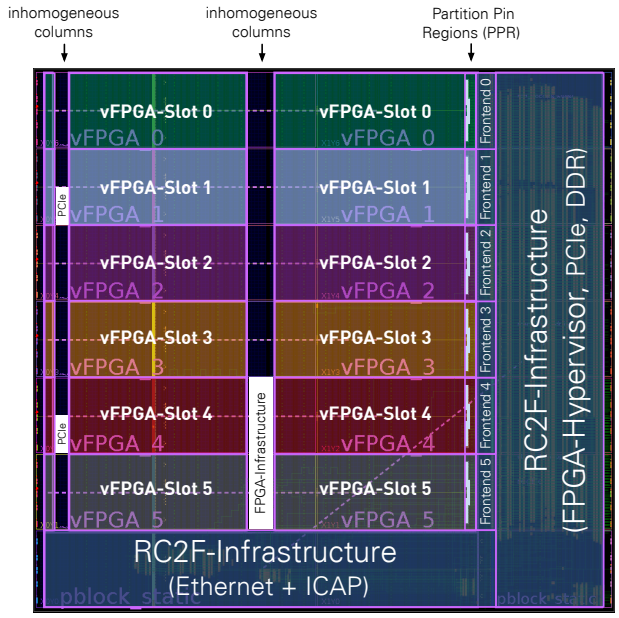


Figure 16. Layout of a Xilinx Virtex-7 XC7VX485T with six vFPGA regions configurable using dynamic partial reconfiguration. The regions and their number are determined by the height of the configuration frames, which consist of one complete column inside a clock region. Regions are homogeneous to allow migration of vFPGAs.

E. vFPGA States

Our FPGA virtualization includes states and transitions similar to traditional VMs. The virtualization of an FPGA requires off-chip monitoring and administration of the vFPGA bitstream database, connected to our cloud management system [3] as well as additional on-chip state transitions. Figure 18 gives an overview of these two parts in our FPGA virtualization. A state with a control flow transition between host and FPGA is called *transition state*.

The global design database and the scheduling of the acceleration tasks (vFPGAs), the allocation to a node and a free region are performed by the cloud system, which also sends commands triggering state transitions on hosts and FPGA devices. In the following, the most important states on host and FPGA – as introduced in Figure 18 – are in detail:

Ready/Shelved: The vFPGA design is located in the global database on a management node.

Booting: First, the node containing the selected vFPGA has to verify if the actual vFPGAs is marked *free*. In a second step the boot process starts, where the partial bitstream is loaded from the database and written into the respective vFPGA location using PCIe and ICAP.

Active: After initialization the vFPGA accelerator is *Active* and the corresponding host application can send/receive application data until a state transition occurs. In case of a *reboot* or *stop* command, the design is halted and reconfigured using the initial or an empty vFPGA design.

Wait for Idle: When a *migration* or *pause* command is received during the *active* state by the host, it forwards the command to the FPGA and both stop the computation and the transmission of further data. Host and vFPGA both wait a limited duration (timeout) until the last data packages are received and stored in the vFPGA's input FIFO and the application's memory.

Snapshot: After the timeout the context of the vFPGA is stable and the actual readback of the vFPGA design is performed by the host using the ICAP. Moreover, the context extraction is performed (see Figure 19) and it is stored in the *virtual register content* file (`.vrc`). At the same time the context of the host application is stored on disk.

Paused: In case of a *pause* command the software and hardware context are stored on disk. If an *abort* command follows, the vFPGA's context in the `.vrc` file becomes invalid (also the host application's context) and the vFPGA gets into the initial *Ready* state. In case of a *resume* command, the initial vFPGA's context is restored by modifying the bitstream using the `.vrc` file as shown in Section VII-F2.

Context Relocation: If the state transition is triggered by a *migration* command, the next vFPGA region is known and the context relocation (bitstream modification) can be performed immediately with the bit positions provided by the `.vrc` file. The modified bitstream and the host application are transferred to the new vFPGA/Node.

Resuming: The modified bitstream with the context from the previous run is used to boot or restore the old context on a different vFPGA.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

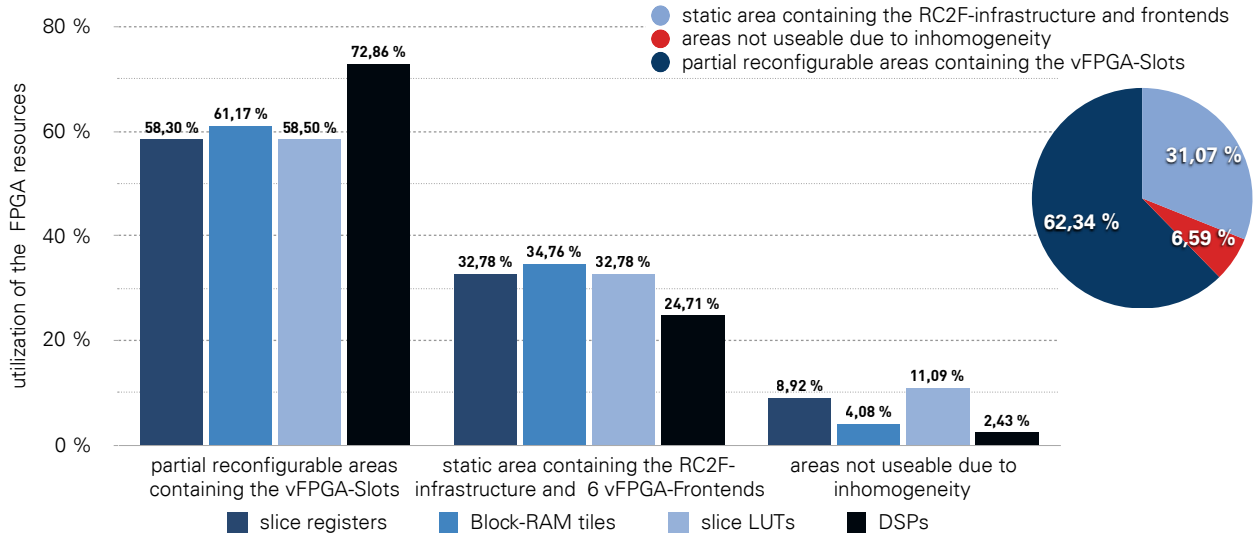


Figure 17. FPGA resources of the three different regions of a Xilinx Virtex-7 XC7VX485T with six vFPGA-Slots, the static region with the FPGA hypervisor and the unusable regions due to homogenization.

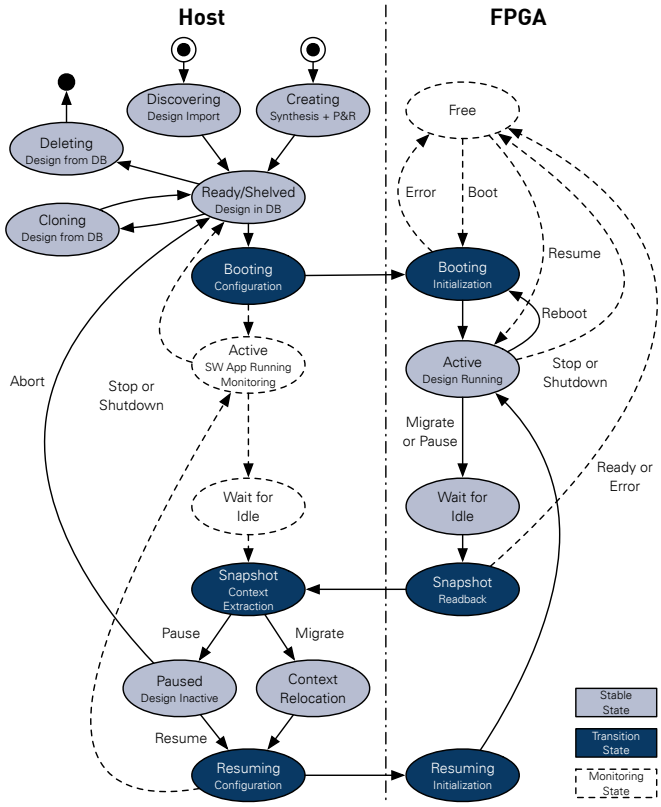


Figure 18. State transitions of a vFPGA (on the host and FPGA).

The context of the vFPGA's DDR3 memory also needs to be saved and restored in the *snapshot* or *resuming* stage using the PCIe connection.

F. Virtualization and Migration Process

Our extended design flow, which generates partial bitstreams and supports vFPGA snapshots as well as the context resumption, is shown in Figure 19. In the following, the components, all additional design flow steps and the generated metadata are described in detail.

For our virtualization we extend the Xilinx Vivado design flow to generate vFPGA bitstreams from user-netlists for every possible vFPGA position. First, directly after synthesis the required region size (single, double, etc.) is chosen (see Table III for appropriate vFPGAs). Afterwards, the design is placed at a first vFPGA region. Before the routing step, the vFPGA region is expanded over the full width of the vFPGA for unlimited routing of the design inside the uninterrupted region. The placements of the same design for all the other vFPGA positions are created by setting the LOC (Location) and BEL (Basic Element Location) information accordingly to the initial placed design. Only the routing is carried out for the additional vFPGA designs to allow static routes inside the different vFPGAs, resulting in designs with identical register and BlockRAM positions for each vFPGA locations on the physical FPGA. After generation of the first bitstream, a mask for extracting the context bits is generated to allow an efficient migration in significantly less time compared to our first approach in [6]. This allows flexible placement of the vFPGA designs at various positions in a cloud system, as well as the migration between vFPGAs on the same or to other physical FPGAs. The bitstreams required for all possible vFPGA positions belonging to a single user design are stored as vRAI as shown in Figure 19.

1) vFPGAs Bitstream Generation and Boot: In the initial step, the full bitstream containing the static design is generated with the traditional Xilinx flow as shown in Figure 19. The bitstream produced contains the basic components, such as PCIe endpoint, memory controller, virtualization layer including the ICAP controller and the static frontend interfaces as well as the local state management for the vFPGAs. The vFPGAs regions themselves are completely empty. The corresponding netlist (`.ngc`) of the static part and all bit positions are stored in the global database and are accessible for the production of partial bitstreams.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

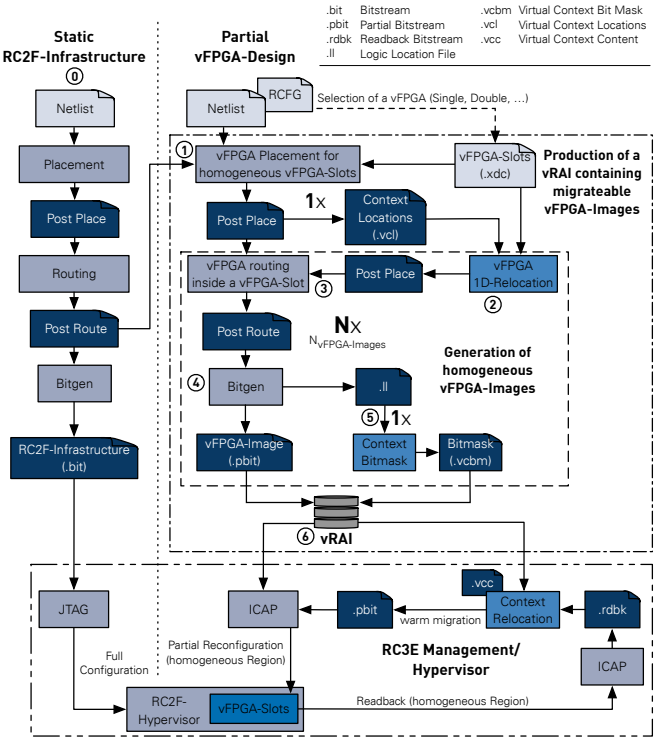


Figure 19. Extended design flow generating partial vFPGA bitstreams and the additional metadata (Beginning with the vFPGA Placement ①).

The following step includes the generation of a partial bitstream based on the static design and a netlist containing the vFPGA design. To achieve an efficient load-balancing and placement on the vFPGA, all possible bitstreams are produced in a single design flow run. For designs which require more than one vFPGA slots, additional partial bitstreams are generated in separate runs. The overall runtime will be reduced in the future by using relocation of placed vFPGAs using homogeneous regions [49]. For the context resumption it is essential to set the option `RESET_AFTER_RECONFIG` for each vFPGA region.

A significant step is the generation of metadata out of these files, which is required to find the register and memory locations in all vFPGA bitstreams. We store the metadata in our *virtual context content* file (`.vcc`) as shown in Figure 19. The information required is extracted from the additional *Logic Location* files (`.ll`) and the *Xilinx Design Language* files (`.xdl`), which are generated during the design flow. The result of this step are partial bitstreams and the corresponding metadata for every possible region. Everything together is stored in the global vFPGA database.

In case of a *pause* or *migration* command, the FPGA is stopped as explained in Section VII-E. After the state became stable, the clock of the corresponding vFPGA is deactivated and the whole context (flip-flops and block RAMs) is frozen. At this point a readback for the `CLB/IO/CLK` and the `BRAM` block is performed and the context is extracted from the bitstream using the `.vcc` file. By the use of the location metadata we only save the registers and the memory used in the design. The readback itself is performed on configuration frame level. In case of a *migration* the location of the new vFPGA is known and the context is written directly into a new bitstream. In case of *pause*, the extracted content is stored in the database as a copy of the `.vrc` file.

2) vFPGA Migration and Context Resumption: In this final step, the relocation and the context resumption are performed. The initial vFPGA bitstream and the corresponding `.vrc` file are used to generate a new bitstream by modifying certain configuration bits. The old flip-flop values are written into the positions of the register initialization bits using the information in the `.vrc` file. To load the values into the flip-flops, the global set/reset (`GSR`) is triggered for the single vFPGA (not global). The Cyclic Redundancy Check (CRC) at the end of the readback bitstream is replaced by a `nop` command to ignore the old CRC.

VIII. RC2F$_{SEC}$ Extension

Security is now more important than ever. Therefore our RC2F$_{SEC}$ extension provides a high level of security for client's data and algorithms. To achieve this, we propose a novel combination of existing security features, a subset of the Transport Layer Security (TLS) protocol and a filter for the partial bitstreams. However, due to various degrees of flexibility show in Figure 1, the extension is only fully available to the production-ready service model RAaaS. But with changes to current FPGA architectures, which will be described later, it would also be available in the BAaaS service model.

A. Security Model

Various adversaries challenge the system's security through multiple vectors. But before these challenges are formalized as requirements for the design, a few assumptions have to be made.

Assumption §A1: *The selected cryptographic algorithms cannot be computationally broken by state-of-the-art attackers. Encrypted data cannot be decrypted or messages signed without access to the keys.*

Assumption §A2: *Naive implementations of cryptographic algorithms are susceptible to side channel attacks, but hardened implementations can withstand better and protect the keys, both shown in [50]. Providing such implementations is not within the scope of this paper. Hence, it is assumed that any cryptographic keys and sensitive intermediate values are secure inside the chip.*

Assumption §A3: *The client's workplace can be trusted and is inaccessible to an attacker.*

Assumption §A4: *The FPGA vendor can be trusted and tries to detect backdoors introduced by manufacturers, tools suppliers or IP vendors, e.g., through analyzing the hardware to find unwanted modifications as shown in [51, 52]. This is the same level of trust the client has to have into hardware in general: CPUs, hard drives and other components might be modified as well.*

Assumption §A5: *Denial of service attacks, interruptions or even physical destruction are secondary and more a concern of the providers, because quality-of-service is an important business factor. The security of data and algorithms has the highest priority.*

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

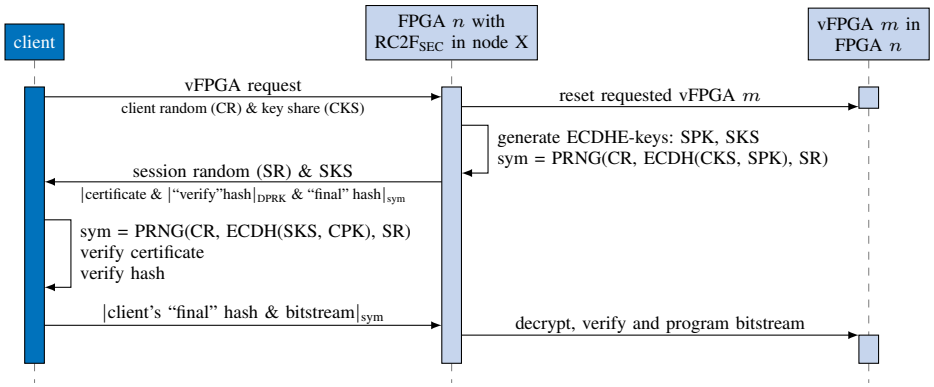


Figure 20. The TLS protocol was adapted to enable the secure and authentic transfers of vFPGA bitstreams. The configuration is protected by symmetric encryption with the key "sym", which is created during the TLS handshake.

Based on theses assumptions, the requirements for the system's design can be defined.

Requirement §R1: *A dedicated third party or trusted authority offering special services only for this system should be avoided.*

Requirement §R2: *A client must be able to establish an authenticated and secured connection from a trusted workplace to the system.*

Requirement §R3: *The FPGA cannot rely on software running on the host machine. The untrusted IaaS provider has direct access and can manipulate anything but the chip.*

Requirement §R4: *The allocation of a vFPGA by an attacker should not interfere with a legitimate client. A strict separation of clients' data is mandatory and the reconfigurable partitions have to be isolated to prevent any interference.*

B. Design

The RC2F$_{SEC}$ extension has to provide two fundamental capabilities:

- Authentication: The FPGA can prove its genuineness to a client.
- Confidentiality: Tamper-proof and secure data transfer.

Microsemi FPGAs already offer similar features, but they do not allow partial reconfiguration preventing virtualization and thus, an efficient cloud deployment. Furthermore, their reputation got a big hit when a backdoor was discovered, which reveals sensitive private keys [51]. The impact of this backdoor could have been minimized, if a more sophisticated protocol offering perfect forward secrecy would have been used. Thus, the RC2F$_{SEC}$ extension implements the well established and thoroughly researched TLS protocol for bitstream and data transfers.

Hence, an embedded TLS processor is required, which is implemented as part of the RC2F$_{SEC}$ extension shown in Figure 11. It does not feature all possible algorithms due to resource constraints, only efficient primitives in terms of performance per logic gate were selected. The initial key exchange follows the Diffie-Hellman algorithm using elliptic curves (ECDH). An advantage over RSA is the cheap generation of new key pairs, making perfect forward secrecy available. Thus, every connection is encrypted with a unique key, compromising one does not affect other connections to the same FPGA. The device's permanent private key is only used to authenticate it through the elliptic curve digital signature algorithm (ECDSA). It reuses the elliptic curve primitives, saving a significant number of resources. Resources can also be saved by combining encryption with authentication. AES128-GCM is an authenticated encryption scheme with high performance hardware implementation, which are also available for CPUs. Finally, SHA256 computes hashes during the handshake.

C. Transfer Protocol

Figure 20 shows the handshake procedure. The client initiated it with a vFPGA request, a key share (CKS) and some random data (CR). The unencrypted request can be used by the cloud provider for billing and scheduling. If the request is unjustly blocked by a cloud provider or evicts another legitimate client, only their quality of service suffers, but not the security of clients' data. Through a complete reset of the vFPGA previous configurations are no longer accessible, and even if data remains in buffers or external memory it is still encrypted.

After the vFPGA reset, a TRNG, which is part of the RC2F$_{SEC}$ extension, generates the session random (SR) and an ephemeral session private key (SPK). With the SPK the public session key share (SKS) is calculated. This new key pair is used to complete the ECDHE key exchange. The resulting shared secret is along with the CR and SR feed into a well defined PRNG. Its output is used to derive various symmetric keys and nonces ("sym"), which are right away utilized to encrypt the FPGA's certificate. Additionally, a hash over the transaction so far is calculated, signed through ECDSA with the device's private key, encrypted and then appended to the certificate. Finally, a second hash over the whole handshake including the CR, CKS, SR, SKS, the encrypted certificate and first hash is calculated, then encrypted and the package is send to the client. Upon receiving it, the unencrypted SR and SKS are used in the same way to derive the symmetric keys and nonces ("sym") through ECDHE and the PRNG. With them, the rest can be decrypted, the certificate and public key verified and the hashes checked. At last, the client also calculates a hash over the whole transaction, now including the second hash, and

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

prepends it to the bitstream. Together they are encrypted with "sym" and transferred to the FPGA. There, after the hash was compared to a locally computed one, the vFPGA bitstream is programmed and the partition ready for use.

If any errors occur or the client uses standardized but not implemented functionalities, the handshake aborts, resets and returns the system into a safe state in which it accepts new connections.

D. Configuration Filter

The configuration filter protects the RC2F as well as other clients' vFPGAs from unwanted modifications, thus, satisfying §R 4. This is possible due to the frame based structure of a bitstream, which is a sequence of commands and data. After a synchronization pattern and some set up, the actual configuration is represented by a repeating series of addresses and data. On a Xilinx 7 Series FPGA each frame consists of 101 words and a full bitstream of a XC7VX485T contains 50 176 frames [53]. A vFPGA is smaller and constraint to a specific area on the chip, which is described by a certain set of frames. The addresses of those frames are extracted during the design phase and do not change later on. They determine the allowed area a client's bitstream can influence.

The configuration filter, shown in Figure 21, acts as a proxy and is located before the ICAP. It receives the decrypted bitstream, scans for interrupting commands like global reset or shut down and blocks them. Its analyzer also detects the command to set the frame address. The address is passed to a set of six detectors, one for each vFPGA slot.

Each checker uses a set of predefined ranges, in Figure 21 five ranges are illustrated, to determine if the current address is within the enabled area. Their results are masked by the slot signal so that the check is only valid for the newly allocated vFPGA. If there is a match, i.e., the address is within the allowed ranges of the current slot, the configuration is passed on to the ICAP. Otherwise, this part of the bitstream is replaced with no-operation (NOP) commands.

The bitstream format is designed as a continuous stream with implicit addresses, in other words not each frame has to have a header specifying its address. A modified bitstream could start at a valid location and write a continuous sequence until the implicit address is outside of the allowed ranges. Thus, the configuration filter cannot only scan for commands to set the frame address. Through an internal counter the end of a 101-word frame is detected and, if another one follows directly afterwards, its address is calculated based on the start address and the current offset. This implicit address is than passed to the range checkers for verification.

E. Implementation

The $RC2F_{SEC}$ extension comprises an elliptic curve multiplier [54], which is shared by the ECDH and ECDSA cores. Furthermore, SHA3 and AES cores, developed by Hsing [55, 56], are used. A so called CMD Decoder handles the handshake and manages the other modules. The resource utilization is shown in Table I.

IX. Implementation Results and Scenario

The resources required for the implementation described in the previous section are shown in the following with a real-world scenario based on our motivation from Section I.

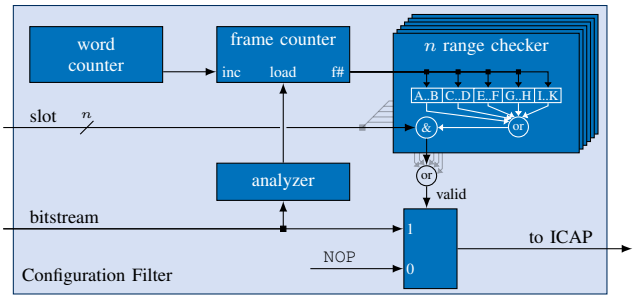


Figure 21. Only if a configuration frame is within the ranges (A..B, ...) of the newly allocated vFPGA slot, it is passed onto the ICAP, otherwise a NOP command is sent instead.

TABLE I. THE $RC2F_{SEC}$ EXTENSION'S RESOURCE UTILIZATION OF A XILINX VIRTEX-7 XC7VX485T.

Submodule	Slice LUTs	Slice Register	BRAM Tile
EC Key Processor[a, b]	30,766	15,158	0
CMD Decoder	7,279	8,714	87
Key Store	269	4,379	0
Configuration Filter	119	99	0
AES De-/Encryption[c]	9,207	11,640	172
Cross clock FIFOs	1,358	3,000	50
Overall	48,878	42,891	309

[a] EC processor by [57] [b] SHA3 core by [55] [c] provided by [56]

A. Implementation

The resource consumption of our prototype introduced in Figure 11 is shown in Table III. Furthermore, the table introduces the size of homogeneous vFPGA regions as outlined in Figure 16. The FPGA resources of every vFPGA can be described with the vector $\vec{\rho}$, which can be defined as shown in Equation (1):

$$\vec{\rho} = \begin{pmatrix} SliceLUTs \\ SliceRegister \\ BlockRAM \\ DSP \end{pmatrix} \quad (1)$$

The vector $\vec{\rho}$ is used in the following to calculate the FPGA resources inside a vFPGA. The aggregated FPGA resources of the homogeneous vFPGAs $\vec{\rho}_{vFPGA}$ can be calculated using Equation (2):

$$\vec{\rho}_{vFPGA}(N_{vFPGA-Slots}, N_{Frontends}) = N_{vFPGA-Slots} \cdot \vec{\rho}_{vFPGA-Slot} + N_{Frontends} \cdot \vec{\rho}_{PPR} \quad (2)$$

In Equation (2), the vector $\vec{\rho}_{vFPGA-Slot}$ describes the resources of a single vFPGA region, $N_{vFPGA-Slots}$ is the number of aggregated vFPGAs, $N_{Frontends}$ is the number of used frontends for the vFPGA and $\overrightarrow{\rho_{ppr}}$ represents the partition

TABLE II. SIZE OF A SINGLE BITSTREAM FOR A VFPGA REGION, NUMBER OF POSSIBLE POSITIONS INSIDE THE FPGA AND SIZE OF THE VRAIS.

	Single	Dual	Triple	Quad	Quint	Hexa
Bitstream (MB)	4.8	9.0	13.0	17.3	21.3	25.3
Locations	6	5	4	3	2	1
vRAI (MB)	33.6	54.0	65.0	69.2	63.9	50.6

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE III. NUMBER OF AVAILABLE RESOURCES INSIDE THE STATIC AND THE AGGREGATED VFPGA REGIONS AND UTILIZATION OF STATIC CONTAINING INFRASTRUCTURE AND HYPERVISOR. THE PARTITION PIN REGION (PPR) IS NECESSARY TO EXCLUDE AND ISOLATE UNUSED PARTITION PINS (PP).

FPGA-Ressource	Static	Utilization of static region					PPR	Into aggregated vFPGA regions and maximal number of frontends					
	region	HF[a]	P[b]	E[c]	M[d]	Total		Single	Dual	Triple	Quad	Quint	Hexa[e]
Slice LUTs	94,824	26%	3%	2%	11%	**42%**	1,200	28,400	56,800	85,200	113,600	142,000	188,400
Slice Register	189,648	11%	2%	1%	4%	**18%**	2,400	59,000	118,000	177,000	236,000	295,000	376,800
Block RAM Tile	369	23%	2%	2%	3%	**30%**	0	105	210	315	420	525	630
DSPs	726	–	–	–	–	**–**	20	340	680	1,020	1,360	1,700	2,040

[a]**HF**: Hypervisor and Frontends [b]**P**: PCIe-Endpoint [c]**E**: Ethernet [d]**M**: DDR3 Memory [e]Largest region without considering homogeneity

TABLE IV. RECONFIGURATION AND MIGRATION TIMES IN SECONDS FOR DIFFERENT SIZED VFPGA-INSTANCES.

Operation	Size of the vFPGAs					
	Single	Dual	Triple	Quad	Quint	Hexa
(1) Readback (s)	0.76	1.43	2.07	2.76	3.39	4.04
(2) Relocate (s)	0.05	0.07	0.10	0.13	0.15	0.18
(3) Configuration (s)	0.04	0.06	0.09	0.11	0.13	0.15
Migration (s)	1.72	3.15	4.51	5.98	7.34	8.79

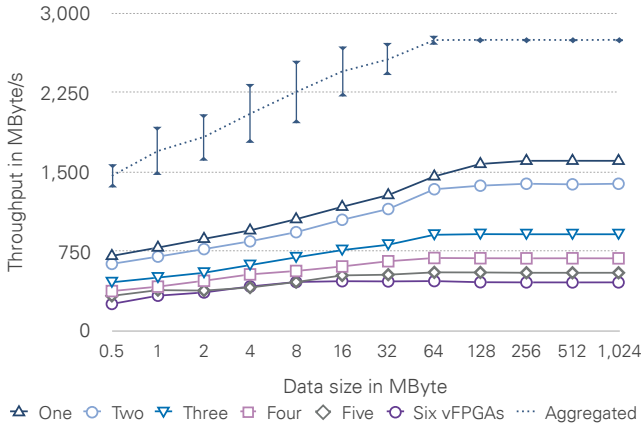


Figure 22. Throughput between host and FPGA with different numbers of concurrent vFPGAs. The diagram shows for each number of vFPGAs the average throughput of one representative vFPGA. The aggregated throughput is thereby the average throughput of all vFPGA compositions on the device.

pin region (PPR) necessary to exclude the unused frontend interfaces from the grouped vFPGAs. When a frontend is used by a vFPGA, the resources inside the PPR are available to the user design inside the vFPGA. The open frontends, which are not used by the vFPGA are therefore treated as stubs and are securely sealed using a partial vFPGA bitstream. The resources of the corresponding PPR are not available inside a vFPGA. All regions except the largest one (Hexa), which has only one possible position, are homogeneous.

The throughput between vFPGAs and host (PCIe Gen2 8x on a Xilinx VC707) with different numbers of concurrently active vFPGAs is shown in Figure 22. The throughput of a single design is limited by a user clock of 100 MHz and a 64-bit data interface. Starting from three vFPGAs, a limitation due to the concurrent users occurs. The throughput shown in Figure 22 is the minimal guaranteed throughput for each vFPGA.

The size of the vRAI packages and the number of possible locations on the physical device are shown in Table II. With 69.2 MByte, a quad vFPGA with bitstreams for three possible positions and a mask file for context migration is the largest vRAI package. Table IV shows the times for configuration, design readback and relocation, as well as a complete migration process for different sized vFPGAs.

B. Scenario

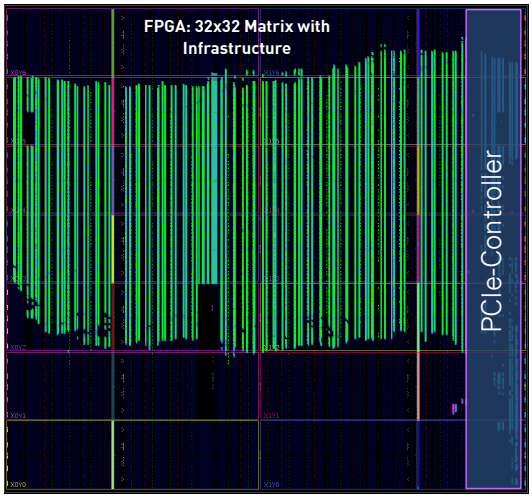
In the following, we show a scenario based on a typical real-world application for our virtualization approach. The goal is to migrate vFPGA designs to achieve a high utilization as shown in Figure 23(e). In a system with jobs arriving and being finished at different points in time, situations as shown in Figure 23(c) can occur. The fragmentation of the physical FPGA restricts only one small vFPGA and one aggregated double sized vFPGA. By migrating the design from user 3 from vFPGA 5 to vFPGA 0 as shown in Figure 23(d), an area for a group of three vFPGAs (triple) becomes available and makes higher utilization of the physical device possible.

C. RC3E Large-Scale Datacenter Simulation

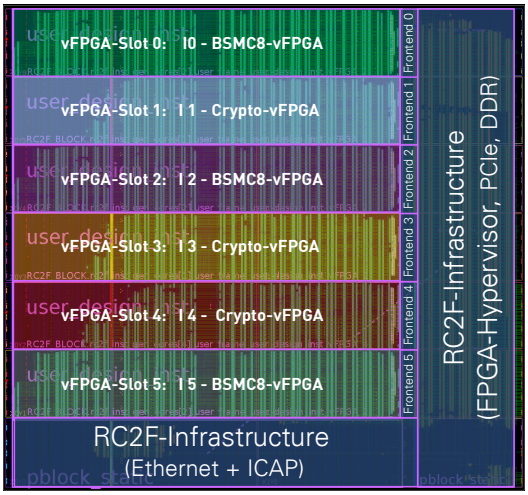
To evaluate the behavior of resource management, and in particular the benefits of virtualized FPGAs as well as their migration in a cloud, the RC3E simulator was developed. The results of the simulation are shown in Figure 24. In addition to the average number of allocated compute nodes, the table shows their energy requirements and the utilization of the FPGA resources available to the user. The Service Level Agreement (SLA) also specifies what proportion of the work packages will be processed within a certain time period (2.5 s) in order to be able to assess the system behavior with regard to the quality of the provision of the resources.

The energy requirement of the cloud is reduced to 69.35 % in the RC3E simulation in the reference scenario in load scenario (I) through the use of FPGAs and the utilization of the physical FPGAs is 27.34 %. RC2F virtualization reduces energy consumption to 24.43 % and increases the physical FPGA utilization to 78.14 %. An additional migration of the vFPGA instances to defragment the system increases utilization to 85.07 % and reduces energy consumption to 22.99 %. The SLA increases slightly by 0.04, or 0.02, as the virtualized resources are available faster than a re-allocating compute node. The additional migration contributes only marginally to saving resources and energy. However, the process of migration

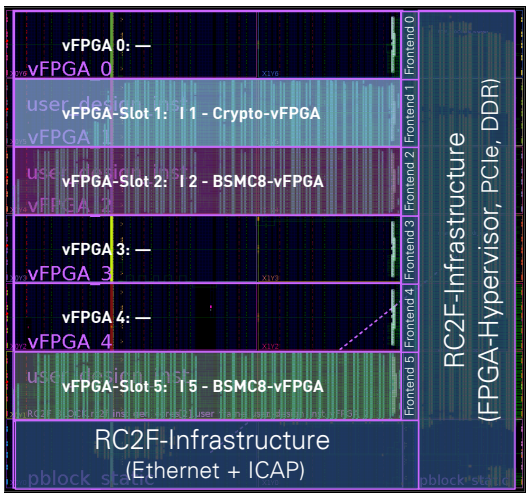
2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



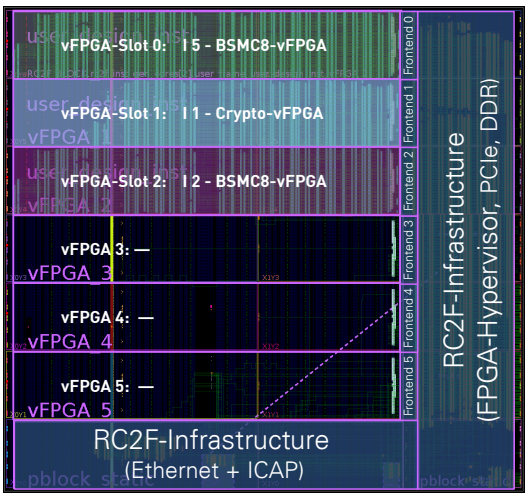
(a) Single userdesign in the classic RSaaS model, which allocates a full physical FPGA without utilizing the entire FPGA.



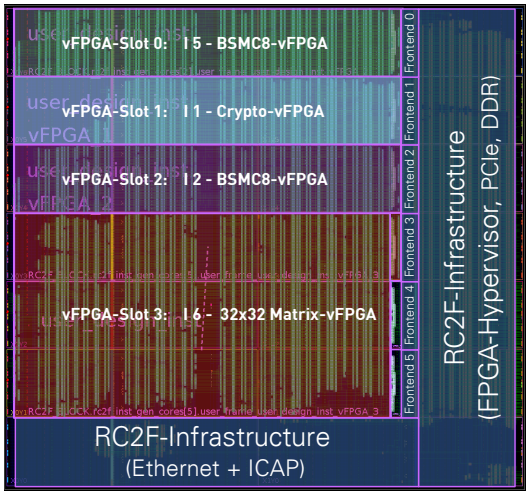
(b) Approximate full utilization of the FPGA with six independent users and designs.



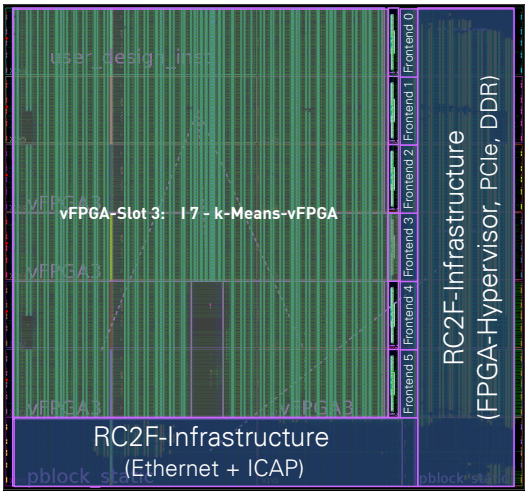
(c) Fragmentation of the physical FPGA caused by dynamic de- and allocation.



(d) Defragmentation providing aggregated vFPGA regions for larger designs.



(e) Utilization of the free region with a design using three aggregated vFPGAs (Triple).



(f) Example of a k-Means design using the largest vFPGAs with six Slots (Hexa).

Figure 23. Scenario with different users and designs on a Xilinx Virtex-7 XC7VX485T with six (vertically) scalable vFPGAs.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

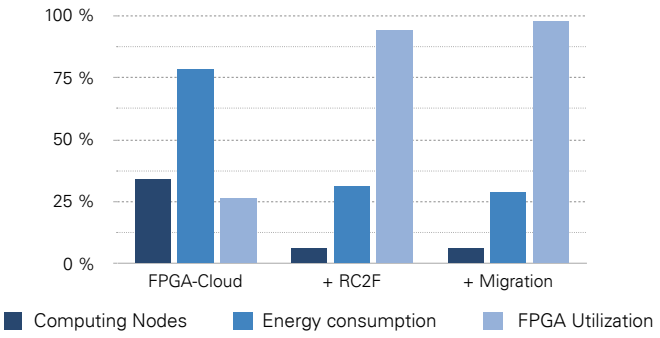


Figure 24. Comparison of the different system configurations within the RC3E simulation.

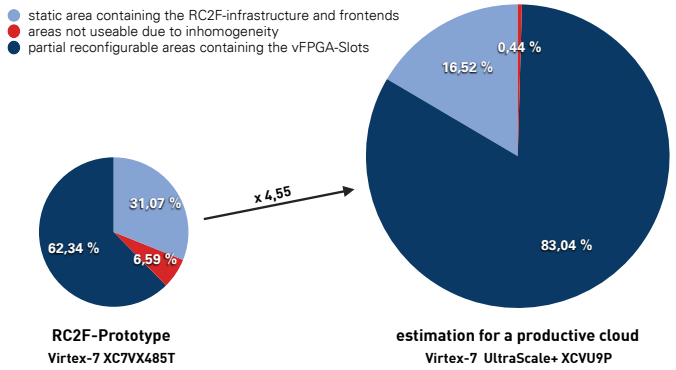


Figure 25. Size of the different regions of the RC2F virtualization transferred to a Xilinx Ultrascale+ FPGA.

TABLE V. RESULTS OF THE RC3E SIMULATION FOR THE SYSTEM CONFIGURATIONS WITH (1) SIMPLE COMPUTE NODES WITHOUT FPGAS, (2) ADDITIONAL FPGAS WITHOUT VIRTUALIZATION, (3) RC2F FPGA VIRTUALIZATION, AND (4) ADDITIONAL MIGRATION. THE SCENARIOS ARE THE LOAD DATA OF A REAL WEB SERVER [58] WITH 47,748 WORK PACKAGES OVER 1,440 MINUTES.

	Cloud (1)	**+FPGA (2)**	**+RC2F (3)**	**+Mig(4)**
Compute Node[a]	376	128	25	24
FPGA Utilization (%)	—	26.74	94.24	97.82
Energy Demand (kWh)	287.37	225.12	89.48	83.06
Energy Demand (%)	100.00	78.34	31.14	28.90
SLA[b]	0.96	0.90	0.92	0.91

[a] Average number of allocated compute nodes.
[b] SLA: Share of work packages being processed within 2.5 s.

adversely affects the SLA because migration is a high priority and new resources are delayed.

Based on the results of the RC3E simulation, it can be expected that both virtualization and the associated migration of vFPGAs can result in resource savings and thus energy without significantly reducing the SLA. The high savings can be explained by the chosen demonstrators and the work packages based on them. If the vFPGA designs completely expose the physical FPGA, virtualization can not save compute nodes and reduce power consumption. However, the migration allows the migration of the vFPGA images to other compute nodes, providing the ability to move parts of the system locally for maintenance, for example.

In addition to evaluating how virtualization and migration affect the optimization of utilization and energy consumption, the RC3E simulator also validated the mapping of vFPGAs to physical FPGAs and compute nodes.

X. CONCLUSION AND OUTLOOK

This paper presented a comprehensive virtualization concept for reconfigurable hardware and its integration into a cloud environment. Our definition of the term *virtualization* is inspired by traditional VMs whose functionalities are transferred to reconfigurable hardware. We develop a paravirtualized infrastructure on a physical FPGA device with multiple vFPGAs. The concept is integrated into a framework, which allows for interaction with the vFPGAs similar to traditional VMs. We create homogeneous regions for the vFPGAs on the physical FPGA to optimize the process of vFPGA migration between different physical FPGAs. Implementation details are described, the necessary resources and the virtualization overhead are presented.

The hardware accelerators used by Amazon in the EC2-F1 instances are Virtex-7 UltraScale+ FPGAs [59] on a VCU1525 Acceleration Development Kit with an XCVU9P [60]. A prognostic transfer of RC2F virtualization to an UltraScale+ (XCVU9P) FPGA provides the partitioning of FPGA resources into the different domains shown in Figure 25. The usable range for the vFPGAs is therefore 83.04 % and does not scale linearly with the size of the FPGA, which is 4.55 times larger than the Virtex-7 XC7VX485T. Due to the homogeneous structure of the UltraScale+ FPGAs, the unusable area has dropped to 0.44 % but still exists, so homogenization is still required.

One significant result of this paper is that the provision of homogeneous FPGA resources is possible with state-of-the-art FPGAs. We think that such approaches are necessary for establishing FPGAs in modern data centers housing clouds. Certainly, when cloud providers like Amazon expand their cloud architectures with high-end FPGAs, such as Xilinx Virtex-7 UltraScale devices [59] it is necessary to utilize the hardware efficiently with multiple designs in a scalable frame inside one physical FPGA. Such kind of flexible approach allows for adaption the individual resources to the users' requirements.

In the future, we plan to establish a productive cloud environment based on RC3E and RC2F at the Helmholtz-Zentrum Dresden-Rossendorf. The system should serve for background acceleration (BAaaS) of scientific applications like [61] and also for FPGA-prototyping (RSaaS) in combination with continuous integration (CI) [62] to optimize the process of hardware design and to satisfy the demands for automated tested FPGA designs for advanced research applications such as [63]. Other promising application areas are the mapping of applications and their distribution on a scalable FPGA cluster [64] and the evaluation of dynamic task offloading from CPUs to (virtualized) FPGAs during run-time, which will be developed on a similar system located at the chair of adaptive dynamic systems at Technische Universität Dresden. Furthermore, the systems are used to investigate economic impacts on hybrid (FPGA) cloud systems.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

REFERENCES

[1] O. Knodel, P. R. Genssler, and R. G. Spallek, "Virtualizing reconfigurable hardware to provide scalability in cloud architectures", *Reconfigurable Architectures, Tools and Applications, RECATA 2017, ISBN: 978-1-61208-585*, vol. 2, 2017.

[2] O. Knodel, A. Georgi, P. Lehmann, W. E. Nagel, and R. G. Spallek, "Integration of a highly scalable, multi-fpga-based hardware accelerator in common cluster infrastructures", in *42nd International Conference on Parallel Processing, ICPP 2013, Lyon, France, October 1-4*, IEEE, 2013, pp. 893–900.

[3] O. Knodel and R. G. Spallek, "RC3E: provision and management of reconfigurable hardware accelerators in a cloud environment", *CoRR*, vol. abs/1508.06843, 2015. [Online]. Available: http://arxiv.org/abs/1508.06843.

[4] ——, "Computing framework for dynamic integration of reconfigurable resources in a cloud", in *2015 Euromicro Conference on Digital System Design, DSD 2015*, IEEE, 2015, pp. 337–344.

[5] O. Knodel, P. Lehmann, and R. G. Spallek, "Rc3e: Reconfigurable accelerators in data centres and their provision by adapted service models", in *9th Int'l Conf. on Cloud Computing, Cloud 2016, June 27 - July 2, San Francisco, CA, USA*, IEEE, 2016.

[6] O. Knodel, P. Genßler, and R. Spallek, "Migration of long-running tasks between reconfigurable resources using virtualization", in *ACM SIGARCH Computer Architecture News Volume 44, HEART 2016*, ACM, 2016.

[7] P. Genssler, O. Knodel, and R. G. Spallek, "A New Level of Trusted Cloud Computing - Virtualized Reconfigurable Resources in a Security-First Architecture", in *Informatik 2017, 47. Jahrestagung der Gesellschaft für Informatik, 25.-29. September 2017, Chemnitz, Deutschland*, 2017.

[8] M. Armbrust, A. Fox, R. Griffith, *et al.*, "A view of cloud computing", *Communications of the ACM*, vol. 53, pp. 50–58, 2010.

[9] P. Mell and T. Grance, "The NIST definition of cloud computing, Revised", *Computer Security Division, Information Technology Laboratory, NIST Gaithersburg*, 2011.

[10] 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, 2008.

[11] J.-A. Mondol, "Cloud security solutions using FPGA", in *PacRim, Pacific Rim Conf. on*, IEEE, 2011, pp. 747–752.

[12] A. Putnam, A. M. Caulfield, E. S. Chung, *et al.*, "A reconfigurable fabric for accelerating large-scale datacenter services", in *Computer Architecture (ISCA), 41st Int'l Symp. on*, 2014.

[13] W. Fornaciari and V. Piuri, "Virtual FPGAs: Some steps behind the physical barriers", in *Parallel and Distributed Processing*, Springer, 1998, pp. 7–12.

[14] Xilinx Inc., *Vivado Design Suite User Guide – Partial Reconfiguration*, UG909 (v2017.1), April 5, 2017.

[15] C. Kachris and D. Soudris, "A survey on reconfigurable accelerators for cloud computing", in *Field Programmable Logic and Applications (FPL), 26th Int'l Conf. on*, 2016.

[16] K. Eguro and R. Venkatesan, "FPGAs for trusted cloud computing", in *Field Programmable Logic and Applications (FPL), 22nd Int'l Conf. on*, IEEE, 2012, pp. 63–70.

[17] V. Kulanov, A. Perepelitsyn, and I. Zarizenko, "Method of development and deployment of reconfigurable FPGA-based projects in cloud infrastructure", in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, May 2018, pp. 103–106. DOI: 10.1109/DESSERT.2018.8409108.

[18] J. Dondo Gazzano, F. Sanchez Molina, F. Rincon, and J. C. López, "Integrating reconfigurable hardware-based grid for high performance computing", *The Scientific World Journal*, 2015.

[19] S. A. Fahmy, K. Vipin, and S. Shreejith, "Virtualized FPGA accelerators for efficient cloud computing", in *Cloud Computing Technology (CloudCom), Int'l Conf. on*, IEEE, 2015.

[20] M. Asiatici, N. George, K. Vipin, S. A. Fahmy, and P. Ienne, "Designing a virtual runtime for FPGA accelerators in the cloud", in *Field Programmable Logic and Applications, Int'l Conf. on*, 2016.

[21] J. Weerasinghe, F. Abel, C. Hagleitner, and A. Herkersdorf, "Enabling FPGAs in Hyperscale Data Centers", in *Cloud and Big Data Computing (CBDCom), Int'l Conf. on*, IEEE, 2015.

[22] A. Vaishnav, K. D. Pham, and D. Koch, "A survey on fpga virtualization", *28th FPL*, 2018.

[23] R. Kirchgessner, G. Stitt, A. George, and H. Lam, "VirtualRC: a virtual FPGA platform for applications and tools portability", in *FPGAs, Proc. of the ACM/SIGDA Int'l Symp. on*, 2012.

[24] H. K.-H. So and R. Brodersen, "A unified hardware/software runtime environment for FPGA-based reconfigurable computers using BORPH", *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, no. 2, p. 14, 2008.

[25] W. Wang, M. Bolic, and J. Parri, "pvFPGA: Accessing an FPGA-based hardware accelerator in a paravirtualized environment", *Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2013 Int'l Conf. on*, pp. 1–9, 2013.

[26] F. Chen, Y. Shan, Y. Zhang, *et al.*, "Enabling FPGAs in the cloud", in *Computing Frontiers, Proc. of the 11th ACM Conf. on*, ACM, 2014, p. 3.

[27] S. Byma, J. G. Steffan, H. Bannazadeh, A. L. Garcia, and P. Chow, "FPGAs in the Cloud: Booting Virtualized Hardware Accelerators with OpenStack", in *Field-Programmable Custom Computing Machines (FCCM), 22nd Annual Int'l Symp. on*, IEEE, 2014, pp. 109–116. DOI: 10.1109/FCCM.2014.42.

[28] Q. Chen, V. Mishra, J. Nunez-Yanez, and G. Zervas, "Reconfigurable Network Stream Processing on Virtualized FPGA Resources", *International Journal of Reconfigurable Computing*, vol. 2018, 2018.

[29] M. Asiatici, N. George, K. Vipin, S. A. Fahmy, and P. Ienne, "Virtualized Execution Runtime for FPGA Accelerators in the Cloud", *IEEE Access*, vol. 5, pp. 1900–1910, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2661582.

[30] M. Happe, A. Traber, and A. Keller, "Preemptive Hardware Multitasking in ReconOS", in *Applied Reconfigurable Computing*, Springer, 2015, pp. 79–90.

[31] J. Rettkowski, K. Friesen, and D. Göhringer, "RePaBit: Automated generation of relocatable partial bitstreams for Xilinx Zynq FPGAs", in *ReConFigurable Computing and FPGAs (ReConFig), 2016 International Conference on*, IEEE, 2016, pp. 1–8.

[32] E. Rossi, M. Damschen, L. Bauer, G. Buttazzo, and J. Henkel, "Preemption of the Partial Reconfiguration Process to Enable Real-Time Computing With FPGAs", *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 2, 10:1–10:24, Jul. 2018, ISSN: 1936-7406. DOI: 10.1145/3182183.

[33] S. Rachana and H. Guruprasad, "Emerging security issues and challenges in cloud computing", *International Journal of Engineering Science and Innovative Technology*, vol. 3, 2 2014, ISSN: 2319-5967.

[34] J. Ryoo, S. Rizvi, W. Aiken, and J. Kissell, "Cloud security auditing: Challenges and emerging approaches", *IEEE Security Privacy*, vol. 12, no. 6, pp. 68–74, Nov. 2014, ISSN: 1540-7993. DOI: 10.1109/MSP.2013.132.

[35] M. A. Will and R. K. L. Ko, "Secure FPGA as a Service - Towards Secure Data Processing by Physicalizing the Cloud", in *2017 IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 449–455.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[36] B. Hong, H.-Y. Kim, M. Kim, L. Xu, W. Shi, and T. Suh, "FASTEN: An FPGA-based Secure System for Big Data Processing", *IEEE Design & Test*, 2017.

[37] OpenStack. (2017). OpenStack - Open Source Cloud Computing Software, [Online]. Available: http://www.openstack.org/ (visited on 2018-11-25).

[38] J. E. Smith and R. Nair, *Virtual machines - versatile platforms for systems and processes*. Elsevier, 2005, ISBN: 978-1-55860-910-5.

[39] ——, "The architecture of virtual machines", *Computer*, vol. 38, no. 5, pp. 32–38, 2005.

[40] R. P. Goldberg, "Survey of virtual machine research", *Computer Journal*, vol. 7, no. 6, pp. 34–45, 1974.

[41] M. Rosenblum, "The Reincarnation of Virtual Machines", *ACM Queue*, vol. 2, no. 5, pp. 34–40, 2004.

[42] Xillybus Ltd. (2017). An FPGA IP core for easy DMA over PCIe, [Online]. Available: http://xillybus.com (visited on 2018-11-25).

[43] T. B. Preußer, M. Zabel, P. Lehmann, and R. G. Spallek, "The portable open-source ip core and utility library poc", in *2016 Int'l Conf. on ReConFigurable Computing and FPGAs (ReConFig)*, Nov. 2016, pp. 1–6. DOI: 10.1109/ReConFig.2016.7857191.

[44] Xilinx Inc., *7 Series FPGAs Integrated Block for PCI Express v3.3 – LogiCORE IP Product Guide*, PG054, 5. April, 2017.

[45] ——, *7 Series FPGAs Memory Interface Solutions – User Guide*, UG586, 18. Januar, 2012.

[46] ——, *LogiCORE IP Tri-Mode Ethernet MAC v5.2 – User Guide*, UG777, 18. Januar, 2012.

[47] H. Zimmermann, "Osi reference model - the iso model of architecture for open systems interconnection", *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, Apr. 1980, ISSN: 0090-6778. DOI: 10.1109/TCOM.1980.1094702.

[48] X. Zhang, S. McIntosh, P. Rohatgi, and J. L. Griffin, "Xensocket: A high-throughput interdomain transport for virtual machines", in *Middleware 2007*, Springer, 2007, pp. 184–203.

[49] R. Backasch, G. Hempel, S. Werner, S. Groppe, and T. Pionteck, "Identifying homogenous reconfigurable regions in hetero"-gene"-ous fpgas for module relocation", in *ReConFigurable Computing and FPGAs (ReConFig), Int'l Conf. on*, IEEE, 2014, pp. 1–6.

[50] H. Gross, S. Mangard, and T. Korak, "An efficient side-channel protected aes implementation with arbitrary protection order", in *Cryptographers' Track at the RSA Conference*, Springer, 2017, pp. 95–112.

[51] S. Skorobogatov and C. Woods, "Breakthrough silicon scanning discovers backdoor in military chip", in *Cryptographic Hardware and Embedded Systems – CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 23–40, ISBN: 978-3-642-33027-8. DOI: 10.1007/978-3-642-33027-8_2.

[52] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using ic fingerprinting", in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 296–310. DOI: 10.1109/SP.2007.36.

[53] Xilinx Inc., *7 series fpgas configuration, User guide 470*, 1.11, Sep. 27, 2016.

[54] C. Rebeiro and D. Mukhopadhyay, "High Speed Compact Elliptic Curve Cryptoprocessor for FPGA Platforms", in *Indocrypt*, Springer, vol. 5365, 2008, pp. 376–388. DOI: 10.1007/978-3-540-89754-5_29.

[55] H. Hsing. (Jan. 29, 2013). Opencores - sha3 core, [Online]. Available: https://opencores.org/project,sha3 (visited on 2018-11-25).

[56] ——, (Dec. 14, 2015). Opencores - tiny aes, [Online]. Available: https://opencores.org/project,tiny_aes (visited on 2018-11-25).

[57] D. Mukhopadhyay, C. Rebeiro, and S. Roy. (Dec. 9, 2008). Elliptic Curve Crypto Processor for FPGA Platforms, [Online]. Available: http://cse.iitkgp.ac.in/~debdeep/osscrypto/eccpweb/index.html (visited on 2018-11-25).

[58] ITA – The Internet Traffic Archive, *EPA-HTTP – A day of HTTP logs from a EPA WWW server.* 2016. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html (visited on 2018-11-25).

[59] Amazon Inc. (2018). Amazon EC2 F1 Instances – Run Custom FPGAs in the AWS Cloud, [Online]. Available: https://aws.amazon.com/ec2/instance-types/f1/ (visited on 2018-11-25).

[60] Xilinx Inc., *VCU1525 Reconfigurable Acceleration Platform – User Guide*, UG1268 (v1.0), 13. November, 2017.

[61] H. Burau, R. Widera, W. Honig, *et al.*, "Picongpu: A fully relativistic particle-in-cell code for a gpu cluster", *IEEE Transactions on Plasma Science*, vol. 38, no. 10, pp. 2831–2839, 2010.

[62] A. Schaefer, M. Reichenbach, and D. Fey, "Continuous integration and automation for devops", in *IAENG Transactions on Engineering Technologies*, Springer, 2013, pp. 345–358.

[63] R. Steinbrück, M. Kuntzsch, M. Justus, T. Bergmann, and A. Kessler, "Trigger generator for the superconducting linear accelerator elbe", 2016. DOI: 10.18429/JACoW-IBIC2015-MOPB011.

[64] L. Kalms and D. Gohringer, "Clustering and Mapping Algorithm for Application Distribution on a Scalable FPGA Cluster", in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, 2016, pp. 105–113.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Investigating Stochastic Dependencies Between Critical Infrastructures

Sandra König
Austrian Institute of Technology GmbH, Center for Digital Safety & Security
Vienna, Austria
Email: sandra.koenig@ait.ac.at

Stefan Rass
Universität Klagenfurt, Institute of Applied Informatics, System Security Group
Klagenfurt, Austria
Email: stefan.rass@aau.at

***Abstract*—Critical infrastructures (CIs) are essential for the welfare and prosperity of a society, and failure of one infrastructure has a significant impact on our everyday life. However, a problem in one critical infrastructure is rarely local but often affects other infrastructures, e.g., limited availability of electricity affects hospitals, water providers and food suppliers. Even a partial failure of critical infrastructures has consequences that are hard to predict unless under stringent assumptions. Among other things, the damage on another infrastructure depends on the availability of substitutes. While such factors are mostly known, many external factors such as weather, temporary demand or load peaks are not precisely predictable so that a stochastic model is required to describe the state of an infrastructure. The state of each infrastructure is described by a random variable and changes its state according to a transition regime that depends on the state of other CIs but also the type of dependency. This yields a model of complex interdependencies with unknown dynamics where the state of a CI is determined by several Markov chains. Several ways exist to determine the actual state of the CI under several influences; the most conservative one is to assume the worst case (by applying the maximum principle). In this work, we provide a more general view that allows incorporating dependencies between input providers. Further, we discuss practical issues such as assessments from several experts and investigate chances for healing and total failure. An implementation of the model in R is used to illustrate how the model may be used in practice to estimate the states of a dependent CI due to limited availability of a provider. This paper describes a stochastic model of dependencies between CIs and discusses issues that arise when applying it.**

Keywords-critical infrastructure; stochastic dependencies; Markov chain; risk propagation; copula.

I. Introduction

Many Critical infrastructures (CIs) are supply networks satisfying the basic needs of society, such as power, water, food, health care or transportation. These CIs naturally depend on one another, and recent developments such as the increased use of control systems increase these interdependencies. The type of dependency is manifold. For example, a hospital depends on water supply as it needs drinking water for staff and patients but also cooling water is necessary for smooth operation. A water provider needs electricity to keep its pumps running but also for the operation of a Supervisory Control and Data Acquisition (SCADA) system. These complex interdependencies are not exactly predictable a can thus be regarded as stochastic [1]. A core characteristic of today's CIs is the fact that a failure or limited availability of one CI often has a considerable impact on CIs depending on it. This has shown in recent years, for example, the disruption of electric power in California in 2001 [2] affected several other CIs, a significant power outage in Italy of about 12 hours [3] resulted in a financial damage of over one billion euros or the most recent hacking of the Ukrainian power grid caused a power outage of several hours [4]. Generally, such dependencies between CIs can be either *continuous*, as it is the case of electricity where a stable supply is required, or *instantaneous*, for example, if the CI's support is just required in an emergency (e.g., police or fire brigade).

In this work, we consider structures that mutually and *continuously* depend on input from several providers, such as water or electricity (see [5][6] for a more detailed discussion). Reduced or even missing supply from a critical provider may cause significant problems for an infrastructure. The actual damage naturally depends on the degree of failure of the provider but is also influenced by many other factors such as availability of substitutes (see [7] for work related to water supply). Especially consequences of reduced support are usually not precisely predictable, which is why we use a stochastic model to describe the condition (state) of a CI based on the states of its providers. These dependencies may be grouped depending on nature or importance of the relation. Such an abstract model can be applied to any infrastructure, as long as the dependency structure is known and can be classified qualitatively in terms of "how severe" a provider's outage is on a finite scale (say, from 1 to 5; see [8] for a discussion of this requirement in light of compliance, auditing, and monitoring). The model thus speaks about different "degrees of failure," where the particular meaning of such a "degree" is up to the specific characteristics of the CI (e.g., status 3 may represent different things or problems for a water provider than for a hospital). The basic model is not too complex by considering only dependencies between two infrastructures at a time and by grouping infrastructures into different classes with different

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

characteristics. However, it is also possible to take into account dependencies between providers of the same type that could be used as a substitute in case of limited availability. Making such a dependence among providers explicit is, for instance, doable with help of copulas that give the joint probability distribution as a function of the individual distributions. This method keeps the complexity still manageable while allowing for higher flexibility and more accuracy of the model.

Paper Outline

The remainder of this article is organized as follows. After a recap of selected related work in Section II, Section III introduces a stochastic model for dependencies between critical infrastructures and its use in practice. Section IV analyzes the chances of total failure or normal functionality based on this model, and Section V extends the basic model towards dependencies between providers of a CI. Section VI shows a small example, and Section VII provides concluding remarks.

II. Related Work

Several models exist on dependencies among critical infrastructures. In [9], a framework for addressing infrastructure interdependencies is presented that distinguishes five different classes of critical infrastructure interdependencies (including dependencies of information and communication technologies). Recent models consider random failure and stochastic dependencies, for example, a multi-graph model that analyze random failures and their effects on critical infrastructures [10]. Other models explicitly look at interdependencies of higher order to identify and assess the effect of failures not only for "consumers" but also for subsequent infrastructures in the dependency chain [11][12]. Cascading effects have been investigated in [13] using an Input-Output Inoperability Model (IIM) based on financial data and Hierarchical Holographic Modeling (HHM) [14] has been used to describe the diverse nature of CI networks and to analyze failures therein. Interdependency graphs are another popular tool for development of methodologies to describe the propagation of failures and cascading effects. Examples include the Cross Impact Analysis (CIA) [15], [16] or the Cross Impact Analysis and Interpretative Structural Model (CIS-ISM) [17]. The Input-Output Inoperability Model (IIM) [18], [19], [20] also provides a detailed view on interdependencies between CIs where linear equations model the consequences. However, these effects are in general measured according to economic aspects, which is only one (and not always the most important) point of view, especially when looking at critical infrastructures. Models that include a more detailed description of the infrastructures are based on Bayesian networks [21] as, for example, the Hierarchical Coordinated Bayes Model (HCBM) [22], [23], [24] or other approaches (cf. [25] and references therein). These models take into account effects of extreme events as well as events with only spars data but can also focus on technical dependencies, see, e.g., [26]. Our basic model is related to various approaches by simulation and co-simulation [27][28][29][30][31]. Typically, these are applicable when the analyst is much more informed about the infrastructure in question since the simulation depicts the internal dynamics (even up to the level of actual network packets to be exchanged). Our perspective is much more high-level and assumes the absence of this detailed information but rather assumes categorical valuations of interdependencies (cf. [5][32][33][34] for more comprehensive overviews), as it is often done in risk management. The stochastic dependency model used here can also be understood as a part of a classical risk management analysis [35].

More formal models include coupled complex systems [36] that are more exposed to large-scale failures or models as described in [37] that describe the increase and the decrease of random failures. Most popular among the stochastic models are Markov chain models. The Interdependent Markov Chain (IDMC) model describes cascading failures in interdependent infrastructures [38]. Conditional Markov transition models are applied in electric power grids [39] and a model including higher orders by adding memory to the Markov chain is presented in [40].

III. Stochastic Dependencies Between Critical Infrastructures

We first describe a basic probabilistic model of dependencies between critical infrastructures in Section III-A and then show some issues when applying it. Potential application to the problem of measuring the resilience of critical infrastructures is given in [41]. Here we show that this model is capable of incorporating assessments from several experts, see Section III-B, and how it can be of use when applying risk management best practices to increase security in Section III-C. Finally, we sketch how the model can be implemented in Section III-D.

A. The Model

Dependencies are often modeled through a directed graph where the nodes represent the various components, and a directed edge describes that the target node depends on the start node. This simple model can also be used to describe interdependencies between critical infrastructures. A high-level view on a system of CIs lets nodes represent the different CIs and a directed edge from CI 1 to CI 2 indicates that CI 2 depends on CI 1, e.g., a hospital depends on a water provider for drinking, but also waste management and fire extinguishing. A more detailed analysis lets nodes represent components of a CI, e.g., a pump or a well as parts of a water utility, and investigates how these depend on one another and other CIs. In either case, the visualization of dependencies provides a basis to understand how limitations in one provider affect dependent CIs and how this effect changes over time. A simulation tool for these cascading effects that is also able to distinguish between short- medium and longtime dependencies is presented in [42]. CIs that support other CIs are called *provider* in the following. In the basic model of stochastic dependencies among CIs [1] we represent the CIs as directed graphs whose nodes, also called *components*, can be in various

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

states that represents their degree of functionality. Whenever one component changes its change from 1 ("working properly") into a state that represents limited functionality (up to state k that represents total failure), this may cause a state change in every CI depending on it.

More formally, suppose that a CI S depends on a set of n providers, enumerated as $P_1, \ldots, P_n$ (all being perhaps themselves CIs). We assume that S will not endogenously experience any state changes since keeping S up and running is a matter of S's business continuity management. The model we describe thus centers on *exogenous* triggers for S to change its state, namely upon problems with one of S's providers $P_1, \ldots, P_n$, drawn as the lower layer of nodes in the bipartite right graph shown in Figure 1. Specifically, we let each of them maintain its own state of functionality, which is communicated (or generally observed) by S, and S can react on a change. This change is governed probabilistically by the state of the provider, or more formally, let S_i be the condition, i.e., state, that P_i can cause for S. This is a random variable distributed over the state space $\{1, 2, \ldots, k\} \simeq \{\text{ok}, \ldots, \text{total failure}\}$, and described as a conditional distribution $\Pr(S_i = x | P_i = y)$, where $P_i = y \in \{1, \ldots, k\}$ is the current state of provider P_i, and $x \in \{1, 2, \ldots, k\}$ is the state that P_i may drive S into. The exact way in which S now depends on the provider P_i can then be specified by a value $p_{i,x,y}$, which can, for example, be set following considerations like these:

- P_i is of vital importance for S, so if i is in a bad condition, S is highly likely to be in trouble as well. Thus, $\Pr(S_i = 5 | P_i = 5) \approx 1$, indicating that S_i will become unavailable if P_i becomes unavailable, since there may be no compensation for P_i's service.
- P_i may only be of minor importance, and an outage of P_i can be bridged by backup resources that S maintains. In that case, we could define $\Pr(S_i = 5, P_i = 5) \approx 0$, modeling that an outage of S is very unlikely even if P_i no longer provides its service.

More fine-grained considerations like the above examples are possible and in a practical instance depend on the application at hand. We leave the two examples here only for illustration and now turn back to the formal description of the model.

Since the dependence of S on two providers may be quite individually different, we internally let S's state be a vector of random states $(S_1, \ldots, S_n)$, each S_i determined by the corresponding provider P_i. These nodes $S_1, \ldots, S_n$ correspond to the colored upper layer in the bipartite graph shown in Figure 1, and the actual state of S that it communicates as a provider to other CIs is a single value in $\{1, 2, \ldots, k\}$ compiled (aggregated) from the vector, i.e., node states, S_i.

This aggregation follows the maximum principle of system security, defining the risk in a system by the highest individual risk therein. Likewise, we compute the state of S as $\max\{S_1, \ldots, S_n\}$, corresponding to S being in trouble if at least one of its providers reports a bad condition (by having a state close or equal to k). In terms of Boolean or fuzzy logic, we would thus define S's state as the (logical) OR of its provider's states. Adopting this view but changing the perspective, any more complex aggregation function to define the state of S from the variables $S_1, \ldots, S_n$ is imaginable, including the use of copulas [43] or triangular norms (from multivalued- and fuzzy logic [44]); we postpone this discussion until Section V. The simulation model studied in this work uses the max-aggregation hereafter.

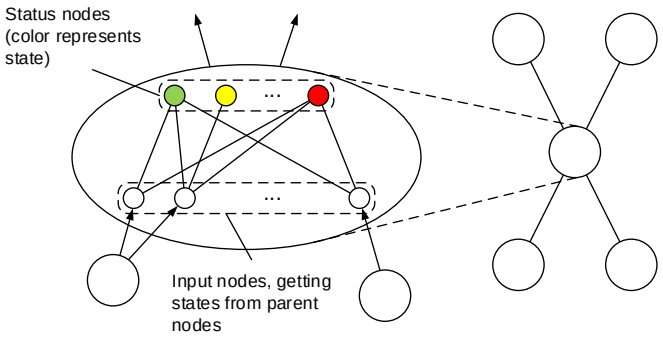


Fig. 1. Model of the inner structure of a critical infrastructure [1]

With k states for each of the n providers, and k states of S itself, the overall specification of the state transition is a set of n stochastic $(k \times k)$ transition matrices with entries being the conditional likelihoods as described above. According to these transition matrices, provider i yields a state S_i of the dependent CI. In case different providers yield different states, the final state S is determined by $S = \max\{S_1, \ldots, S_n\}$, i.e., we consider the worst case.

While a stochastic model is convenient (actually natural) to describe uncertain consequences, it is often challenging to put it in practice. The main issue in this regard is estimation of transmission probabilities since experts often feel uncomfortable or feel unable to provide concrete and reliable values. Several aids exist, however:

1) allow an expert to give qualitative values, e.g., on a 5-tier scale
2) ask an expert to tell a level of confidence with every estimate (also qualitative, such as "very sure", "somewhat unsure" or "'just guessing')
3) ask several experts for their individual (subjective) assessments

The last point may allow experts assess only some transitions (depending on their expertise) but also raises the question of how to deal with several expert opinions. We will focus on this below. The second point seems to blow up the data needed but can be put into practice in a way that actually in most cases reduced the amount of input data. For each state of the provider, it is necessary to find an entire distribution over all possible states of the dependent CI. This can be done by determining the most likely value as a subjective prediction, and interpreting the level of confidence in the told opinion as some kind of variance. Practically, that means that for confidence "very high" we choose a distribution that puts mass 1 on the predicted value and 0 elsewhere, while we choose a uniform distribution over all state for a confidence "just guessing". Intermediate assessments such as "somewhat unsure" put some probability mass on states close to the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

specified one but no positive probability on values that are too far from the opinion. For the case of three different states of a CI this mapping is illustrated in [45], including a short discussion on the interpretation of "totally sure" assessments. For the case of five possible states (as used in our example later on) we get the mapping given in Table I, where the resulting vectors make up one row of the corresponding transitions matrix.

TABLE I. DISTRIBUTION OVER STATES OF DEPENDENT CI BASED ON EXPERT ASSESSMENT (PREDICTION, CONFIDENCE)

prediction	very sure	somewhat unsure	just guessing
1	(1,0,0,0,0)	(2/3, 1/3, 0, 0, 0)	(1/5,1/5,1/5,1/5,1/5)
2	(0,1,0,0,0)	(1/4, 2/4, 1/4, 0, 0)	(1/5,1/5,1/5,1/5,1/5)
3	(0,0,1,0,0)	(0, 1/4, 2/4, 1/4, 0)	(1/5,1/5,1/5,1/5,1/5)
4	(0,0,0,1,0)	(0, 0, 1/4, 2/4, 1/4)	(1/5,1/5,1/5,1/5,1/5)
5	(0,0,0,0,1)	(0, 0, 0, 1/3, 2/3)	(1/5,1/5,1/5,1/5,1/5)

Assessments of this kind yield an entire row of the transmission matrix (i.e., a discrete distribution) and we will denote it by F to represent this fact.

B. Several Expert Opinions

So far we have assumed that each dependency has been assessed by one expert only. We stress that, however, not necessarily the same person is required to rate all dependencies. In case we have more than one expert assessment for a connection, all opinions should be taken into account to increase the data quality underlying the subjective assessments, as well as to reduce the pressure on each expert to be responsible solely for the given assessment. In our setting, a number of K experts opinions yield to multiple distributions over the possible states of the dependent CI. We denoted these as $F_1, \ldots, F_K$ and aggregate the distribution to find the estimate

$$F = \alpha_1 \cdot F_1 + \ldots + \alpha_K \cdot F_K$$

for one edge in the bipartite graph in Figure 1, where $\alpha_1, \ldots, \alpha_K$ satisfy

$$\alpha_1 + \ldots + \alpha_K = 1.$$

The parameter α_i can be interpreted as the weight (influence) of expert i's opinion in the overall assessment. In case we do not distinguish between different expertise, we choose uniform weights, i.e., $\alpha_i = 1/K$ for all i.

C. Impact Estimation as a Part of Risk Management

One step in risk analysis and risk management [46][35] is to estimate the impact due to a security incident. There exist several ways to do that, and the choice of a specific method depends on the situation at hand, see, e.g., [47] for co-simulation applied to power distribution grids, [48] for an application of agent-based simulation or [49] for the spreading of ransomware. This impact estimation is not only needed for an analysis of effects of an incident (such as a malware attack) but can also help to test the use of countermeasures (such as patching a computer). Whenever actions are taken to reduce the damage on a CI due to a problem in a provider, this changes the dependency between the two, in particular, it reduced the probability that the dependent CI changes into a severe state (if the countermeasure is effective). The network itself does not change but the effect of reduced availability of a provider on the dependent CI changes, which yields a different transitions matrix. The simulation is then be rerun with a different set of transition matrices to see if the resulting losses reduce. These estimates of damages for various scenarios build up a generalized payoff matrix that allows finding an optimal way to protect the system at hand. The important point is that the assessment as we outline here is exactly the same as what is done along a conventional risk management process anyway: following standard frameworks like the IT Grundschutz (by the German federal office for information security (BSI) [50]), or the ISO 27k standard, a typical step is an assessment of how assets or components depend on one another. The pure information of a dependence then naturally defines the dependency graph topology. Our method then goes further in asking what would happen to one component if another component fails. That is, we propose a mere "additional use" of the artifacts from risk management in the here proposed simulation framework to aid the impact assessment to gain some "objectivity."

D. Simulation

The stochastic dependency model between critical infrastructures has a straightforward implementation in software such as the freeware `R`. This makes it handy to use in any field due to the high interoperability of `R` with other systems.

The simulation starts with an incident affecting one node, which subsequently (directly and indirectly) triggers descendant CIs to change their status according to the likelihoods in their inner bipartite graphs. In that way, the simulation reveals how far an incident will propagate through the network of CIs (within the runtime of the simulation), and can thus be used to estimate the effect a problem in one component has on a specific critical infrastructure or generally on other components. Further, it allows an empirical estimation of the number of components that are in a critical state (i.e., reach the highest status k) or the relative frequency of one specific CI being in a critical state.

More explicitly, we model the network of infrastructures as a graph $G = (V, E)$ with vertices $v \in V$ that represent the infrastructures and edges $e \in E$ representing the dependencies between them. A common difficulty in specifying such probabilistic models is the issue of where to get the conditional probabilities from (that we already mentioned in Section III-A, along with hints on how to think about these values). To relieve this practical challenge, we let the conditional likelihood specification be discrete and replace the poll for probabilities by the question to specify, resp. assign, a certain *edge class* c instead (the edge again being one in the bipartite inner graph modeling a CI; cf. Figure 1). An edge classification is hereby chosen from a set of candidates $\{1, 2, \ldots, C\}$. Each edge class represents a fixed type of inner or mutual dependency which carries the sought probabilities with them. The information in

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

the edge class can also include different levels of importance of a CI for its successor CI (provider consumer dependency), and other explanatory information or data useful for the simulation. Each edge $v \to w$ is then associated with a representative number for its class c that carries an attribute being the probability for the simulation.

This allows the model parameterization to be done upfront and independently of the concrete CI, and eases matters of model parameterization in the absence of empirical data to estimate conditional probabilities. Depending on this class c the state i of v influences the state of w through a multinomial distribution MN($p_{i,c}$). That is, the j-th component of the vector $p_{i,c}$ gives the probability that w will be in state j in this situation. Figure 2 shows an algorithm in pseudo-code, which simulates T time steps.

```
1: t ← 0
2: while t < T
3:   for each node v, set N(v) = {w ∈ V : (v, w) ∈ E}
4:     for each neighboring node w ∈ N(v)
5:       let c be the class of v → w,
6:       let i be the current state of node v,
7:       draw the status of w from MN(p_{i,c})
8:       t ← t + 1.
9:     endfor
10:  endfor
11:endwhile
```

Fig. 2. Simulation algorithm

Just as the input, the result of this simulation is a network of connected critical infrastructures where each CI is in one specific state. For a better understanding of the results, visualization with use color codes (e.g., ranging from green to indicate a working state to red, alerting about a critical condition) is helpful. Numerically, the results of the simulation can be summarized as a table that lists how many components are on average in any of the possible states.

An implementation of the dependency model in the event simulation tool `OMNeT++` is presented in [42]. The prototype described therein enables modeling the network of CIs as a directed graph whose nodes are colored to represent its state (ranging from green to red to represent several levels of functionality). External events can trigger the simulation by changing the states of one or more components. The propagation through the network is implemented as a message exchange over a fictitious communication channel. After a predefined running time, the tool yields a chronological record of the state changes for each component of the network. An illustrative example is included.

IV. Chances of Healing and Total Failure

Technically speaking, the changes between the states of the CIs based on the state of its provider is described by a Markov chain, whose states correspond to the states of the CI. We adopt an ordered numeric representation for the nominal scale of health, ranging from "good" $\simeq$ state 1, up to state "failure" $\simeq$ state k. The rich theory related to Markov chains then enables us to compute the chances that a CI fails completely (i.e., is in state k) or remaining in good shape (i.e., is in state 1) for a certain period. We will denote the i-th unit vector by u_i to represent the situation where an asset is in state i with likelihood 1.

As before, let S_i denote the state of the dependent CI due to the state of its i-th provider. In the classical model from Section III-A, we assumed a worst case scenario, i.e., the overall state S of the dependent CI is $S = \max\{S_1, \ldots, S_n\}$. Under this assumption, a CI is in state 1 only if every provider causes a switch to state 1 (or a stay in state 1). That is:

$$\Pr(S = 1) = F_{S_1,\ldots,S_n}(1, \ldots, 1), \tag{1}$$

where F is the joint distribution over all CI states. Unfortunately, this only simplifies in the case of i.i.d. variables (where we get a product). However, it can generally be decomposed into the individual, i.e., marginal, distributions unconditionally describing the state of each CI, plus an outer function capturing the interdependence. This outer function is a *copula* and essentially is the mathematical function describing the dependencies visualized in the graph $G = (V, E)$. Formally, we have $F_{S_1,\ldots,S_n}(1, \ldots, 1) = C(F_{S_1}(1), \ldots, F_{S_n}(1))$. In case of stochastic independence, the last term simplifies to $\prod_{i=1}^{n} \Pr(S_i = 1)$, i.e., we have $C(x_1, \ldots, x_n) = x_1 \cdot x_2 \cdots x_n$.

Since we assume that at the beginning every asset is working properly (i.e., in state 1) we know that the likelihood of returning to that state after t time steps is $u_1 P^t$. Further, the Markov Chain returns to the starting distribution if it is a limiting distribution, i.e., if $u_1 = u_1 P$ holds. So, equation (1) is fulfilled if the vector u_1 is a stationary distribution of each of the involved Markov chains.

In case k is an absorbing state (i.e., if there is no recovery from a failure, say, if the CI is irreparably destroyed) for all involved Markov chains, we find that $\Pr(S_i = k) = 1$ and thus $\Pr(S = k) = 1$. Otherwise, the probabilities $\Pr(S_i = k)$ can be determined by the law of total probability

$$\Pr(S_i = k) = \sum_{j=1}^{k} \Pr(S_i = k | v_i = j) \cdot \Pr(v_i = j)$$

where v_i is the i-th provider that yields to state S_i according to the transition probability $p_{jk} = \Pr(S_i = k | v_i = j)$. The probabilities $\Pr(v_i = j)$ depend in turn on the providers of the provider v_i, which makes an explicit analysis challenging.

V. Dependencies Between Providers

The basic model introduced in Section III assumes that providers are independent and the effect of a limited availability is in not influenced by the state of other providers. However, the effect of a problem in one provider might be limited as long as there is another one of the same type that is fully working. The basic model can be extended to capture a certain degree of dependency between providers, e.g., if they are of the same or very similar kind and can be used as substitutes. This particularly applies to the situation where

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

a CI has contracts with several providers to reduce the damage in case the provider is not available.

Let $S_1, \ldots, S_n$ denote the states of a CI due to the state of the corresponding providers according to the transition matrices. Until now we have assumed that the state S communicated to other CIs is determined by the maximum principle, yielding

$$S = \max\{S_1, \ldots, S_n\}. \quad (2)$$

However, this is a very conservative view, as it ignores the fact that due to dependencies between providers the overall state may be better than caused by a single provider. For example, some critical infrastructures use several providers of the same type to avoid this strong dependency, e.g., they have contracts with more than one telecommunication provider. In that case, we may replace the relation given in (2) by the more general form

$$S = f(S_1, \ldots, S_n)$$

where f is any function that aggregates the n values into one "overall" state. If we know about dependencies between providers of the same type (e.g., one may be a substitute) we can reformulate this as

$$S = f(C(S_{t_1}, \ldots, S_{t_y}), S_{r_1}, \ldots, S_{r_m}) \quad (3)$$

where providers $t_1, \ldots, t_y$ are of the same type and thus assumed to be dependent to some extent (while providers $r_1, \ldots, r_m$ are not of this type). We model their joint distribution with a suitable copula or a more general function. Possibilities include the following here at least:

- $\min$-operator: this is a copula, and in setting the overall state of a CI to the minimum state of all its providers, our convention that "healthy" has a state representation number less than that for "failure", we end up with the following semantic:
  A CI will not change into failure state unless *all* its providers have failed. This is an "OR-aggregation" since the CI remains intact if any of its providers is intact.
- $\max$-operator: this provides the reverse semantic as above since a CI will go into failure state if at least one of its providers fails. Logically, they are in that sense "AND-connected".
- Combinations of the two, where a complex Boolean term can have each its connectives represented by an artificial intermediate node with min or max aggregations to model AND or OR operations therein. We leave this as a simple extension and not go into formal details at this point[1].

The decomposition in equation (3) can be further refined by using several copulas for several types of providers, i.e.,

$$S = f(C_t(S_{t_1}, \ldots, S_{t_y}), \ldots, C_s(S_{s_1}, \ldots, S_{s_z}))$$

for copulas $C_t, \ldots, C_s$.

[1]Extending this Boolean approach further, we can even model a logical negation by setting the conditional probabilities for a state change accordingly: suppose that S has only a single provider P, then we can have S to "invert" the state of P by specifying that $\Pr(S = 5|P = 1) = 1$ and $\Pr(S = 1|P = 5) = 1$, meaning that S is in a good/bad condition if and only if P is in a bad/good condition. Intermediate states $k = 2, \ldots, 4$ would herein be triggered with the likewise defined conditional probabilities.

VI. An Illustrative Example

To illustrate the application of the presented model, we evaluate a small example of high-level dependencies between a few critical infrastructures. When applying the model to real use cases, one needs to decide on the granularity of the network representation that corresponds to the degree of detail in the description. If the aim of the analysis is getting an overview of dependencies between critical infrastructures (as we do in this small example), then each node represents a single CI. Dependencies are assessed on a general level, and the results are accordingly general. If more data on a CI are available, then this CI can, in turn, be represented as an entire network where components represent significant components of the network (this approach has been illustrated in [45]).

Let us consider a subnetwork of dependent CIs, which consists of a hospital that depends on a water provider, an electricity provider as well as transportation infrastructures (roads). The dependencies between the different components in the network are classified as either "minor", "normal" or "critical" depending on how important the service provisioning is for the CI. In this small example, we classified input from the electricity provider as "normal" (as we assume existence of an emergency power system), input from a water provider as "critical" (substitution by bottled water is usually just possible for a limited period of time, but substituting water for waste management or fire extinguishing systems is even more problematic to replace; we do not cover these details hereafter), and the transport connection as "minor", since even if roads are temporarily congested or blocked, aerial transportation remains possible for critical patients.

The effects of an outage of each provider may be different, and to ease matters as before with the edge classes, we propose specifying transition matrices *per dependency criticality level*, i.e., transitions from working into a failure state become more likely the more critical the dependency on the respective provider is. Consequently, we will below specify three transition matrices for dependencies of levels "minor", "normal" and "critical".

In [1], we considered transmission matrices that are estimated with certainty. Dropping this assumption now, we show a case where experts are not certain about their assessments, i.e., they provide information as described in Section III-B. Further, we assume that two experts do the assessment for the critical dependency on water with potentially different backgrounds. For this example, we consider 5 possible states for each node, where 1 represents the situation where everything works smoothly, while 5 stands for serious problems including total failure.

Suppose for example that data, as shown in Table II, has been collected, where we represent the confidence levels by numbers ranging from 1 ("totally unsure") to 3 ("totally sure").

We assign the same weight (importance) to both experts doing assessments of the dependency on water, that is we chose $\alpha_1 = \alpha_2 = 0.5$. This yields individual transmission matrices $T_{\text{dependency-criticality-level}} = (t_{ij})_{i,j=1}^{5}$, in which the ij-th

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

(a) EXPERT ASSESSMENT FOR ELECTRICITY

State of Provider	Predicted Value	Confidence Level
1	1	2
2	1	1
3	1	1
4	4	1
5	4	1

(b) EXPERT ASSESSMENT FOR TRANSPORT

State of Provider	Predicted Value	Confidence Level
1	1	2
2	1	2
3	1	2
4	1	1
5	1	1

(c) FIRST EXPERT ASSESSMENT FOR WATER

State of Provider	Predicted Value	Confidence Level
1	1	1
2	2	1
3	4	1
4	4	2
5	5	2

(d) SECOND EXPERT ASSESSMENT FOR WATER

State of Provider	Predicted Value	Confidence Level
1	1	2
2	2	1
3	4	1
4	5	2
5	5	3

TABLE II. ASSESSMENTS FOR ELECTRICITY AND TRANSPORT

entry corresponds to the conditional likelihood $t_{ij} := \Pr(\text{CI gets into state } j \mid \text{provider is in state } i$ and given the respective criticality of the dependency). We choose

$$T_{minor} = \begin{pmatrix} 2/3 & 1/3 & 0 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{pmatrix},$$

$$T_{normal} = \begin{pmatrix} 2/3 & 1/3 & 0 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{pmatrix}$$

and

$$T_{critical} = \begin{pmatrix} 13/30 & 8/30 & 3/30 & 3/30 & 3/30 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 8/80 & 8/80 & 18/80 & 28/80 & 18/80 \\ 0 & 0 & 3/24 & 10/24 & 11/24 \\ 0 & 0 & 0 & 1/6 & 5/6, \end{pmatrix},$$

where $T_{critical}$ is the linear combinations of the two matrices induced by the two expert assessments.

Figure 3 (left side) displays the dependencies graphically, with arrows annotated according to the criticality of the dependency. The right part of Figure 3 shows how the inner model of Figure 1 corresponds to a dependency and is instantiated according to the matrices above. For example, if the dependency's criticality is "minor" and the respective provider is in state 4 (i.e., it has rather serious problems), this will yield to a state 5 of the critical infrastructure that depends on it with a likelihood of $1/5 = 0.2$.

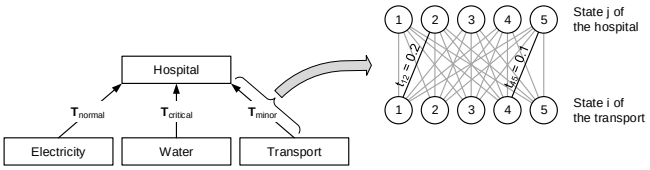


Fig. 3. Example instance

The results of our analysis can be interpreted in (at least) two ways. On one hand, it provides information on a specific node in the network (such as it frequency of failure), including information about which node caused the failure. On the other hand, it provides an overview on the average number of nodes in a specific state, so in particular on the expected number of failing components.

Initially, we assumed that all components are in state 1 (i.e., operate smoothly) except for the water provider that is in state 2 facing some (maybe temporary) problems. This scenario yielded to a critical state for the hospital in 201 out of 1000 cases. Note that in this example, this critical state can only be caused by the state of the water provider since a CI of normal or even minor importance will never cause a critical level while being in state 1 (i.e., both entries in the transition matrices are zero). In a more elaborated example with numerous dependencies and other components facing problems state changes may be caused by other components as well. The simulation may then be used to track which provider caused the worse state (if this information is stored as well). This information may help providers to identify critical dependencies and indicates where future investments may be useful (e.g., it might make sense to have a substitute for a provider that often causes problems).

Table III shows the average number of nodes (CIs) that are in each of the 5 possible states. This gives a general overview on the situation of the entire network, e.g., it can be seen that on average 1.921 nodes are in a state worst than 1 (indicating that they have a problem). Further, it gives an estimate of the number of components in the worst case, which might help planing resources needed to fix problems. Additionally, such information may be used to measure resilience of a component [41].

TABLE III. AVERAGE NUMBER OF AFFECTED NODES DUE TO INCREASED LEVEL OF CRITICALITY

Criticality	1	2	3	4	5
Nodes	2.079	1.313	0.201	0.206	0.201

VII. CONCLUSION

A basic stochastic model of dependencies between critical infrastructures can capture issues such as the impossibility of exact predictions in a network of interdependent critical infrastructures. It describes the degree of availability of a provider by different states and let this potentially cause a state change in the dependent CI. In the simplest case, the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

actual state of a CI depending on many providers is assumed to be the worst of all states caused by any supporting CI, corresponding to a worst-case view.

In this work, we addressed practical issues when applying the model by describing how experts assessment can be incorporated without the need of an agreement between several experts. Instead, opinions of different experts might be combined, and each expert is asked to rate the confidence in his prediction. Further, we illustrated how to implement the simulation either in the statistical software `R` or in the event simulation tool `OMNeT++` and exemplified the approach with a small example.

The basic model can be extended to take into account dependencies between providers. Further, we explained how the model fits into a risk analysis as a tool to estimate the impact of an incident affecting parts of a network of CIs and how it can be used to compare the current situation with a scenario in which countermeasures have been implemented before.

ACKNOWLEDGMENT

This work was done in the context of the project "Cross Sectoral Risk Management for Object Protection of Critical Infrastructures (CERBERUS)", supported by the Austrian Research Promotion Agency under grant no. 854766.

REFERENCES

[1] S. König and S. Rass, "Stochastic dependencies between critical infrastructures," in *SECURWARE 2017: The Eleventh International Conference on Emerging Security Information, Systems and Technologies*, IARIA, Ed., 2017, pp. 106–110.

[2] S. Fletcher, "Electric power interruptions curtail California oil and gas production," *Oil Gas Journal*, 2001.

[3] M. Schmidthaler and J. Reichl, "Economic Valuation of Electricity Supply Security: Ad-hoc Cost Assessment Tool for Power Outages," *ELECTRA*, no. 276, pp. 10–15, 2014.

[4] J. Condliffe, "Ukraine's Power Grid Gets Hacked Again, a Worrying Sign for Infrastructure Attacks," 2016, URL: https://www.technologyreview.com/s/603262/ukraines-power-grid-gets-hacked-again-a-worrying-sign-for-infrastructure-attacks/ [accessed: 2017-07-26].

[5] *Managing the Complexity of Critical Infrastructures: A Modelling and Simulation Approach*, ser. Studies in Systems, Decision and Control. Cham: Springer, 2016, vol. 90.

[6] R. Klein, E. Rome, C. Beyel, R. Linnemann, W. Reinhardt, and A. Usov, "Information modelling and simulation in large interdependent critical infrastructures in irriis," in *Critical Information Infrastructure Security: Third International Workshop, CRITIS 2008, Rome, Italy, October13-15, 2008. Revised Papers*. Berlin, Heidelberg: Springer, 2009, pp. 36–47.

[7] E. Luiijf, M. Ali, and A. Zielstra, "Assessing and improving SCADA security in the Dutch drinking water sector," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 3-4, pp. 124–134, 2011.

[8] A. Abou El Kalam and Y. Deswarte, "Critical infrastructures security modeling, enforcement and runtime checking," in *Critical Information Infrastructure Security: Third International Workshop, CRITIS 2008, Rome, Italy, October13-15, 2008. Revised Papers*. Berlin, Heidelberg: Springer, 2009, pp. 95–108.

[9] S. Rinaldi, J. Peerenboom, and T. Kelly, "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies," *IEEE Control Systems Magazine*, pp. 11–25, 2001.

[10] N. K. Svendsen and S. D. Wolthusen, "Analysis and Statistical Properties of Critical Infrastructure Interdependency Multiflow Models," in *2007 IEEE SMC Information Assurance and Security Workshop*, June 2007, pp. 247–254.

[11] M. Theoharidou, P. Kotzanikolaou, and D. Gritzalis, "Risk assessment methodology for interdependent critical infrastructures," *International Journal of Risk Assessment and Management*, vol. 15, no. 2-3, pp. 128–148, 2011, URL: http://www.inderscienceonline.com/doi/abs/10.1504/IJRAM.2011.042113 [accessed: 2018-11-24].

[12] P. Kotzanikolaou, M. Theoharidou, and D. Gritzalis, "Assessing n-order dependencies between critical infrastructures," *International Journal of Critical Infrastructures*, vol. 9, no. 1-2, pp. 93–110, 2013, URL: http://www.inderscienceonline.com/doi/abs/10.1504/IJCIS.2013.051606 [accessed: 2018-11-24].

[13] R. Setola, S. De Porcellinis, and M. Sforna, "Critical Infrastructure Dependency Assessment Using the Input-Output Inoperability Model," *International Journal of Critical Infrastructure Protection (IJCIP)*, vol. 2, pp. 170–178, 2009.

[14] Y. Y. Haimes, "Hierarchical Holographic Modeling," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 9, pp. 606–617, 1981.

[15] T. J. Gordon and H. Hayward, "Initial experiments with the cross impact matrix method of forecasting," *Futures*, vol. 1, no. 2, pp. 100–116, 1968.

[16] M. Turoff, "An alternative approach to cross impact analysis," *Technological Forecasting and Social Change*, vol. 3, pp. 309–339, 1971.

[17] V. A. Bañuls and M. Turoff, "Scenario construction via Delphi and cross-impact analysis," *Technological Forecasting and Social Change*, vol. 78, no. 9, pp. 1579–1602, 2011.

[18] Y. Y. Haimes and J. Pu, "Leontief-Based Model of Risk in Complex Interconnected Infrastructures," *Journal of Infrastructure Systems*, vol. 7, no. 1, pp. 1–12, 2001.

[19] J. R. Santos and Y. Y. Haimes, "Modeling the demand reduction input-output (I-O) inoperability due to terrorism of interconnected infrastructures," *Risk Analysis: An Official Publication of the Society for Risk Analysis*, vol. 24, no. 6, pp. 1437–1451, 2004.

[20] R. Setola, S. De Porcellinis, and M. Sforna, "Critical Infrastructure Dependency Assessment Using the Input-Output Inoperability Model," *International Journal of Critical Infrastructure Protection*, vol. 2, pp. 170–178, dec 2009.

[21] M. I. Jordan, Ed., *Learning in graphical models*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1999.

[22] Y. Y. Haimes, J. Santos, K. Crowther, M. Henry, C. Lian, and Z. Yan, "Risk Analysis in Interdependent Infrastructures," in *Critical Infrastructure Protection*, ser. IFIP International Federation for Information Processing. Springer, Boston, MA, 2007, pp. 297–310.

[23] Z. Yan, Y. Y. Haimes, and M. G. Wallner, "Hierarchical coordinated Bayesian model for risk analysis with sparse data," Baltimore, USA, 2006.

[24] Y. Y. Haimes, J. Santos, K. Crowther, M. Henry, C. Lian, and Z. Yan, "Risk Analysis in Interdependent Infrastructures," in *Critical Infrastructure Protection*. Boston, MA: Springer US, 2007, vol. 253, pp. 297–310.

[25] T. Schaberreiter, S. Varrette, P. Bouvry, J. Röning, and D. Khadraoui, *Dependency Analysis for Critical Infrastructure Security Modelling: A Case Study within the Grid'5000 Project*. Berlin, Heidelberg: Springer, 2013, pp. 269–287.

[26] T. Schaberreiter, S. Varrette, P. Bouvry, J. Röning, and D. Khadraoui, "Dependency Analysis for Critical Infrastructure Security Modelling: A Case Study within the Grid´5000 Project," in *Security Engineering and Intelligence Informatics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 8128, pp. 269–287.

[27] R. Caire, J. Sanchez, and N. Hadjsaid, "Vulnerability analysis of coupled heterogeneous critical infrastructures: A Co-simulation approach with a testbed validation," in *IEEE PES ISGT Europe 2013*. IEEE, 2013, pp. 1–5.

[28] R. Jaromin, B. Mullins, J. Butts, and J. Lopez, "Design and Implementation of Industrial Control System Emulators," in *Critical Infrastructure Protection VII*, ser. IFIP Advances in Information and Communication Technology. Berlin, Heidelberg: Springer, 2013, vol. 417, pp. 35–46.

[29] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *ISGT 2011*. IEEE, 2011, pp. 1–6.

[30] M. Faschang, F. Kupzog, R. Mosshammer, and A. Einfalt, "Rapid control prototyping platform for networked smart grid systems," in *Proceedings IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*. Vienna, Austria: IEEE, 2013, pp. 8172–8176.

[31] M. Findrik, P. Smith, J. H. Kazmi, M. Faschang, and F. Kupzog, "Towards secure and resilient networked power distribution grids: Process and tool adoption," in *Smart Grid Communications (SmartGridComm), 2016 IEEE International Conference on*. Sidney, Australia: IEEE Publishing, 2016, pp. 435 – 440.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[32] J. Butts, *Critical Infrastructure Protection VII: 7th IFIP WG 11. 10 International Conference, ICCIP 2013, Washington, DC, USA, March 18-20, 2013, Revised Selected Papers*, ser. IFIP Advances in Information and Communication Technology. Berlin/Heidelberg: Springer, 2013, vol. v.417.

[33] S. M. Rinaldi, "Modeling and simulating critical infrastructures and their interdependencies," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*. IEEE, 2004, pp. 1–8.

[34] E. Wiseman, "Critical Infrastructure Protection and Resilience Literature Survey: Modeling and Simulation," URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/1003598.pdf [accessed: 2018-11-24].

[35] S. Rass and S. Schauer, Eds., *Game Theory for Security and Risk Management: From Theory to Practice*, ser. Static & dynamic game theory : foundations & applications. Cham, Switzerland: Birkhäuser, 2018.

[36] B. A. Carreras, D. E. Newman, P. Gradney, V. E. Lynch, and I. Dobson, "Interdependent Risk in Interacting Infrastructure Systems," in *40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007*, Hawaii, USA, 2007, pp. 112–112.

[37] N. Svendsen and S. Wolthusen, "Analysis and Statistical Properties of Critical Infrastructure Interdependency Multiflow Models." IEEE, 2007, pp. 247–254.

[38] M. Rahnamay-Naeini and M. M. Hayat, "Cascading Failures in Interdependent Infrastructures: An Interdependent Markov-Chain Approach," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 1997–2006, 2016.

[39] Z. Wang, A. Scaglione, and R. J. Thomas, "A Markov-Transition Model for Cascading Failures in Power Grids," in *2012 45th Hawaii International Conference on System Sciences*, 2012, pp. 2115–2124.

[40] S. Wu and M. T. Chu, "Markov chains with memory, tensor formulation, and the dynamics of power iteration," *Applied Mathematics and Computation*, vol. 303, pp. 226–239, 2017.

[41] S. König, T. Schaberreiter, S. Rass, and S. Schauer, *A Measure for Resilience of Critical Infrastructures*, 2018, (in press).

[42] T. Grafenauer, S. König, S. Rass, and S. Schauer, "A simulation tool for cascading effects in interdependent critical infrastructures," in *International Workshop on Security Engineering for Cloud Computing (IWSECC 2018)*, 2018, (in press).

[43] R. B. Nelsen, *An Introduction To Copulas*, ser. Lecture Notes in Statistics 139. Springer, 1999.

[44] T. Ross, J. M. Booker, and W. J. Parkinson, *Fuzzy Logic and Probability Applications: Bridging the gap*. ASA SIAM, 2002.

[45] S. König, T. Grafenauer, S. Rass, and S. Schauer, "Practical risk analysis in interdependent critical infrastructures - a How-To," in *SECURWARE 2018: The Twelfth International Conference on Emerging Security Information, Systems and Technologies*. IARIA, 2018, pp. 150–157.

[46] S. Schauer, "A Risk Management Approach for Highly Interconnected Networks," in *Game Theory for Security and Risk Management*, ser. Static & dynamic game theory : foundations & applications. Cham, Switzerland: Birkhäuser, 2018, pp. 285–311.

[47] M. Findrik, P. Smith, J. H. Kazmi, M. Faschang, and F. Kupzog, "Towards secure and resilient networked power distribution grids: Process and tool adoption," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2016, pp. 435–440.

[48] B. S. Onggo, J. Busby, and Y. Liu, "Using agent-based simulation to analyse the effect of broadcast and narrowcast on public perception: A case in social risk amplification," in *Proceedings of the Winter Simulation Conference 2014*, Dec 2014, pp. 322–333.

[49] S. König, A. Gouglidis, B. Green, and A. Solar, *Assessing the Impact of Malware Attacks in Utility Networks*. Cham: Springer International Publishing, 2018, pp. 335–351.

[50] Bundesamt für Sicherheit in der Informationstechnik, "BSI-Standard 100-2: IT-Grundschutz Methodology," https://www.bsi.bund.de/cln_156/ContentBSI/grundschutz/intl/intl.html, May 2008, version 2.0, english.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

On the Design, Construction and Calibration of Dual-Probe Heat-Pulse Soil Moisture Sensor: Towards an Industrial Solution

Antonio Valente*, Arata Andrade Saraiva†, N. M. Fonseca Ferreira‡, and Salviano Soares§

*INESC TEC - INESC Technology and Science
and University of Trás-os-Montes and Alto Douro
Vila Real, Portugal
Email: avalente@utad.pt
ORCID: 0000-0002-5798-1298

†CEUMA
and State University of Piaui-Piripiri, Brazil
and University of Trás-os-Montes and Alto Douro,
Vila Real, Portugal
Email: aratasaraiva@gmail.com
ORCID: 0000-0002-3960-697X

‡INESC TEC - INESC Technology and Science
and Institute of Engineering of Coimbra,
Polytechnic Institute of Coimbra,
Coimbra, Portugal
Email: nunomig@isec.pt
ORCID: 0000-0002-2204-6339

§IEETA - UA
and University of Trás-os-Montes and Alto Douro,
Vila Real, Portugal
Email: salblues@utad.pt
ORCID: 0000-0001-5862-5706

***Abstract*—There is a need for a multi-functional probe for small-scale measurements of different soil properties measured within identical soil volumes. Dual Probe Heat Pulse (DPHP) sensors are an economical solution for this since they measure simultaneously temperature, volumetric water content, and soil thermal properties: diffusivity and volumetric heat capacity. However, all DPHP sensors to date have very complex manufacturing processes. This paper aims to design and build a DPHP sensor based only on a Printed Circuit Board (PCB) board, which comprises the probes and all supporting electronics leading to a low cost and simple manufacturing process. The proposed system includes signal-processing circuits, a microcontroller, and communicates by a Serial Digital Interface at 1200 baud protocol (SDI-12). The sensor needs a one-time two calibrations: distance from heater to temperature sensor calibration using agar solution; and sensor calibration in soil using Tottori Dune sand. Both calibration processes showed a reasonably good agreement between measured and fitted data. In conclusion, results also show that it is possible to build the multi-functional DPHP sensor in a low cost process, and this was the first time that a multi-functional probe was build using a Printed Circuit Board (PCB) as support.**

Keywords–soil moisture sensor; soil thermal properties; dual-probe; heat-pulse sensor; thermal conductivity; thermal sensors.

I. INTRODUCTION

The heat pulse-based soil moisture sensors, of which stand out Dual Probe Heat Pulse (DPHP) sensors, are an economical solution for soil moisture measurements. This paper is an extension of [1] and focuses the construction and calibration of PCB based soil moisture sensor based on the DPHP method.

The DPHP-based sensor has two elements, the heater and the temperature sensor, separated by a short distance, r, as depicted on Figure 1. The heater and the temperature sensor are placed in a physical support, the heater probe and the temperature sensor probe, respectively. A voltage pulse, with finite duration t_0, is applied to the heater element causing the heat to flow in all directions around the heater element. Heat 'traverses' the surrounding soil and the temperature rise will be measured by a sensor on the temperature probe.

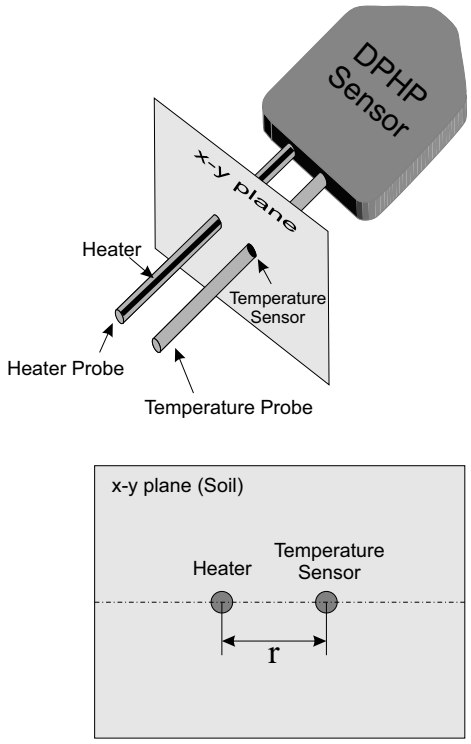


Figure 1. Dual Probe Heat Pulse method.

Most sensors use needles as heating element and temperature sensing element. There are very few cases where the sensor is not directly connected to a data-logger, that is, all the inspection, control and processing is not done in

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

the sensor itself, but in an external element - the data-logger (which implies, almost always, a dedicated data-logger for each sensor).

The use of needles, and the control, acquisition and processing, external to the sensor, has made this method difficult to use in the field. Therefore, in this work we resolve these shortcomings.

In particular, we make the following contributions:

- Place all the electronics near the probes (heating and temperature probes). This contribution (Section IV), although already achieved by [2], uses only a temperature probe, making the construction of the sensor simpler at the expense of measuring the flow of water in the soil.
- We present a new form of sensor construction which uses the PCB, also used for the rest of electronics, as a support for the temperature and heating probes (Section IV). This is a great achievement as it will enable the industrial production of the sensor at a reduced cost.
- All calculations of the model used (Section III) are performed inside of the on-board microcontroller.
- Two calibration types: calibration of the distance between the heater element and the temperature sensor (for this calibration process it was used agar); and calibration of the sensor using Tottori Dune sand as soil type whose physical characteristics are widely reported [3]–[5]. The results of the calibrations (Section V) showed good results of the model used.

The rest of this paper is organized as follows. Section III describes the models implemented for the calculation of the soil thermal properties and volumetric water content. Section IV details all the fabrication process of the DPHP sensor, calibration method of the distance between heater and temperature sensor, and SDI-12 communication commands implemented. Section V addresses the sensor calibration results and shows sensor prototypes. The acknowledgement and conclusions close the article.

II. Related Work

Since Campbell [6], much research has been done and reported that the heat pulse technique is an economical technique to measure soil thermal properties and soil moisture content [2], [4]–[21]. Although it is an economical method, for several reasons the sensors that use the DPHP method have not yet left the research laboratories.

Some works mention field use, such the work described in [11]. However, the sensors developed were prepared for use in the laboratory and were subsequently used in the field. This solution leads to some problems such as accidental and continuous heating of the dual-probe heat-pulse sensors during a field test, causing the heater element resistance to be damaged, as reported in [11].

As mentioned, in almost all developed sensors [2], [4]–[16] uses needles as housings for the heat source and the temperature sensor. Normally, the heating probe is composed of a needle where a conductive wire (with a high resistivity, suitable for producing resistances for heating - $0.062\,\mathrm{mm}$ Evanohm enameled wire) is inserted inside. A single very fine wire is threaded into the needle four times creating two turns. The heater wire was spliced onto copper leads to form the connections to the resistance forming the heater element.

The probe with the temperature sensor is elaborated in the same way, by inserting inside the needle and the longitudinally centering the temperature sensor (thermocouple or thermistor). Both needles are then filled with an epoxy glue of high thermal conductivity and low electrical conductivity. Therefore, the use of this process is laborious and difficult, if not impossible, for industrialization.

Other authors have attempted, without known success, to develop soil moisture sensors based on the heat pulse without the use of needles. In this group, there are the works of [17]–[19], [21] who developed microelectronics to try to improve the manufacturing process and in other works a button-shaped sensor was developed [16], [20]. The button-shaped sensor also use the same kind of wire (very thin) and in one of the works [16] a needle is used which is folded into a ring type shape. This needle, which is the heater element, uses the same assembly method described above.

Most of the work done with DPHP sensors uses data-loggers to acquire data from temperature probe, to control heat pulse duration and to perform calculations. A recent work [2] that presents a novel design of a multifunctional penta-needle thermo-dielectric sensor with advantage for in-situ and simultaneous determination of water content (θ_v), thermal conductivity (λ), and thermal diffusivity (κ) in porous media, overcomes this problem but still uses needle type probes.

In conclusion, of all the literature consulted, no sensor was found based on the heat pulse method that uses the supporting PCB for the reading, control and communication electronics, as well as support of the elements heater and temperature sensor. This fact allows us to think that this is the first time that this process is used, being a good basis for industrial production of this type of sensors.

III. Theory

There are some models to describe the operation of sensors based on the DPHP method. The simpler model, described by Campbell [6], considers that the heat source is an infinite line and the heat pulse is released instantaneously. Other models [7] increase the constraints of the model of Campbell, considering that the heat pulse is finite, that the source line is finite and, finally, that it is not a source line, but a cylinder. Of these models the most common is the model which considers the finite heat pulse generated by an infinite source line [7]. In the present work, this will be the adopted model due to its precision and relative ease implementation in a microcontroller-based system. The solution for conducting radial heat from a short-duration, t_0 , heat pulse away from an infinite source line, for $t > t_0$ is

$$\Delta T(r,T) = \frac{q'}{4\pi\kappa\rho c}\left[\mathrm{E_i}\left(\frac{-r^2}{4\kappa(t-t_0)}\right) - \mathrm{E_i}\left(\frac{-r^2}{4\kappa t}\right)\right] \quad (1)$$

where, ΔT is change in temperature ($^\circ C$), r is radial distance from the line source (m), t is time (s), q' is the energy input

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

per unit length of heater per unit time ($\mathrm{W\,m^{-1}}$), ρc is the volumetric heat capacity ($\mathrm{J\,m^{-3}\,^{\circ}C^{-1}}$), κ is the thermal diffusivity ($\mathrm{m^2\,s^{-1}}$) of the medium surrounding the heater, and $-\mathrm{Ei}(-x)$ is the exponential integral [22]. The thermal diffusivity (κ) and volumetric heat capacity (ρc) can be determined from heat-pulse measurements using the single-point method [8], or a nonlinear curve fitting where (1) is fitted to measured $\Delta T(r,t)$ data [10]. It was showed that accuracy in ρc will be limited by the accuracy of r, and accuracy in κ will be limited by the accuracy of both r and time of maximum temperature change, t_{M} [8]. The single-point method makes use of the fact that the temperature response at some distance r from the heater displays a maximum, so that we can take the derivative of (1) with respect to time, set the result equal to zero, and obtain the time, t_{M}, to the maximum temperature change (ΔT_{M}). This yields an expression for estimating κ where

$$\kappa = \frac{r^2}{4}\left[\frac{\frac{1}{(t_{\mathrm{M}}-t_0)}-\frac{1}{t_{\mathrm{M}}}}{\ln\left(\frac{t_{\mathrm{M}}}{t_{\mathrm{M}}-t_0}\right)}\right] \tag{2}$$

which is a function of r, t_{M} and t_0. Rearrangement of (1) yields an expression for estimating ρc, for $t > t_0$,

$$\rho c = \frac{q'}{4\pi\kappa\Delta T_{\mathrm{M}}}\left[\mathrm{E_i}\left(\frac{-r^2}{4\kappa(t_{\mathrm{M}}-t_0)}\right)-\mathrm{E_i}\left(\frac{-r^2}{4\kappa t_{\mathrm{M}}}\right)\right] \tag{3}$$

where κ is obtained from (2). To minimize errors, the single-point method requires an accurate measurement of r and times t_{M} and t_0. For ρc estimation, in addition to κ from (2) and r, are needed accurate measurements of q' and ΔT_{M}. Volumetric heat capacity ρc of soil can be determined as the sum of the heat capacities of the individual constituents and considering that the air is ignored and solids defined to include the mineral and organic matter fractions then soil water content θ_{v} can be defined as a function of volumetric heat capacity [23],

$$\theta_{\mathrm{v}} = \frac{\rho c - \rho_{\mathrm{b}} c_{\mathrm{s}}}{(\rho c)_{\mathrm{w}}} \tag{4}$$

where ρ is density, c specific heat, θ_{v} volumetric water content, and the subscripts b, s and w indicates bulk, average properties of the solids (minerals+organic matter), and water, respectively. Since $(\rho c)_{\mathrm{w}}$ is known, measurements of ρc obtained with the sensor can be used together with estimates (or preferably measurements) of soil bulk density (ρ_{b}) and specific heat (ρ_{s}) to obtain θ_{v}.

IV. Materials and Methods

The developed sensor is a complete solution and this approach, together with the non-use of needles for the heating and temperature probes, is new. Next, all details of the sensor construction, as well as its firmware, the communication protocol and as referred to in the previous section the knowledge of the true value of r (r_{eff}) through calibration in agar, will be explained in detail.

A. Sensor System Description

In Figure 2 is depicted the electronic layout of the sensor system, the core unit is on-board $16\,\mathrm{bit}$ microcontroller (PIC24F32KA301) with very low power consumption, 12-channel $12\,\mathrm{bit}$ Analog-to-Digital Converter (ADC), serial communications modules (UART - Universal Asynchronous Receiver-Transmitter, SPI - Serial Peripheral Interface, and $\mathrm{I^2C}$ - Inter-Integrated Circuit), and hardware Real-Time Clock (RTC) Calendar with alarms. The temperature sensor probe consists of a $10\,\mathrm{k\Omega}$ (NCP15XH103F03RC) thermistor, a precision ($0.1\,\%$) voltage reference of $2.048\,\mathrm{V}$ (LM4128) and a $24\,\mathrm{bit}$ ADC (MCP3421). The heating probe consists of a series of 15 resistors of $1\,\Omega$, controlled by an electronic switch composed of transistors. The system power is from the SDI-12 power ($6\,\mathrm{V}$ to $12\,\mathrm{V}$) that feeds a dual DC regulator: $2\,\mathrm{V}$ to $5\,\mathrm{V}$ for powering the heat pulse (LM1117), and $3.3\,\mathrm{V}$ for the rest of the system (MIC5219). The voltage control of the first DC-DC converter is achieved with a pulse (Pulse With Modulation - PWM pulse) controlled by the microcontroller.

The microcontroller controls the heat pulse through the transistors switch to enable/disable the power to the heater. To determine accurately the value of q' (heat input per unit length per unit time), average current through the heater was determined by sampling the voltage across a $0.18\,\Omega$ resistor ($1\,\%$) in series with the heater $15\,\Omega$ resistor ($15 \times 1\,\Omega$), and the voltage across resistance series of the heater and current measure resistor. The value q' is determined by

$$q' = V_{\mathrm{heater}} \times I_{\mathrm{heater}} \times \frac{1}{l_{\mathrm{heater}}} \tag{5}$$

where V_{heater} is the measured voltage across the heater, I_{heater} is the current through the heater given by $I_{\mathrm{heater}} = V_{\mathrm{R=0.18}}/0.18\,\Omega$ ($V_{\mathrm{R=0.18}}$ is the measured voltage across the $0.18\,\Omega$ resistor), and l_{heater} is the length of the heater ($0.0353\,\mathrm{m}$). All voltages measured for q' calculations, are performed by microcontroller internal ADC ($12\,\mathrm{bit}$). Temperature from the temperature probe was measured by sampling the voltage drop across the thermistor in series with a $10\,\mathrm{k\Omega}$ ($0.1\,\%$) resistor. This was done on the $24\,\mathrm{bit}$ ADC to ensure sufficient sampling accuracy for determining temperature.

B. Sensor Construction

The prototype of the developed sensor, as shown in Figure 3, is based on the printed circuit board (PCB) as substrate. In the PCB are welded all the components necessary for the operation of the sensor. The design of the PCB was made in the form of a fork with two 'rods' that form the

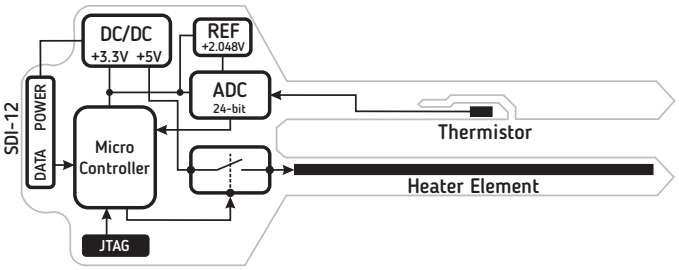


Figure 2. Sensor overview

heating probe and the temperature probe. The thermistor in the temperature probe is placed in a thermally insulated tab. The distance between these two elements were designed to be $6 \times 10^{-3}\,\mathrm{m}$. However, this distance has to be calibrated because as previously described an error in this parameter contributed significantly to the error in the determination of κ and ρc [9].

The process of mounting the sensor is as follows:

- Manufacture of PCB board;
- Assembly of all electric components, except for the first resistance of the heating element due to the mold;
- Sealing, using a mold as depicted on Figure 4, with thermoplastic molding resin to achieve high quality sealing and protection of components of the main circuit. Overtec 820 15 Hotmelt Glue Gun, from Techsil Limited, UK, was used with the respective polyamide resin OverTec 5 FR;
- Missing resistor placement;
- Placement of an epoxy adhesive in the heating element (in all resistors forming it) and in the thermistor.

As can be seen from the description of the process used for the elaboration of the prototype, it can be turned into an industrial process.

C. Firmware

The developed firmware, after all the initialization, enters into sleep mode waiting for a SDI-12 command. After a `aM!` command the firmware will perform a complete measurement as presented on Figure 5.

For the implementation of the exponential integral in (3) in order to obtain the ρ_c value, the Chebyshev approximations for the Exponential Integral $\mathrm{E_i}(x)$ were used [24]. The implementation was elaborated in C and can be executed in the microcontroller used (16 bit with 32 kB of program memory and 2 kB of SRAM).

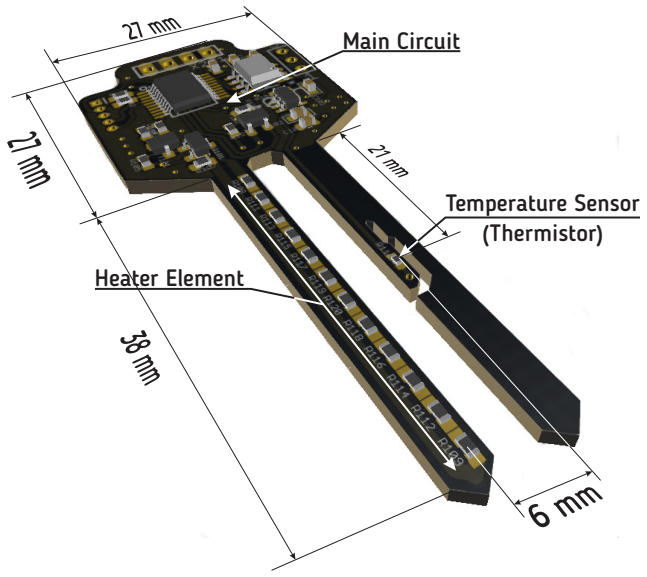


Figure 3. Sensor 3D view.

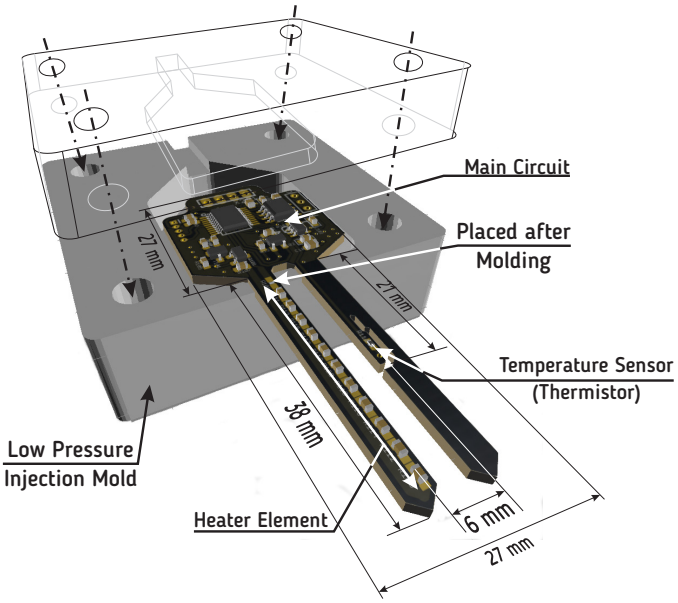


Figure 4. 3D view of the construction details.

1: **repeat** ▷ Read soil temperature before applying heat
2: $StartTemperature \leftarrow temperature$
3: **until** $time < 12\,\mathrm{s}$
4: HeatPulse ON (5 V)
5: **repeat** ▷ Read heat power
6: $HeatVoltage \leftarrow voltage$
7: $HeatCurrent \leftarrow current$
8: **until** $time < 8\,\mathrm{s}$
9: HeatPulse OFF
10: Calculate q' using (5)
11: **repeat** ▷ Read Temperature Increase
12: ▷ (3 samples/second)
13: $\Delta T \leftarrow temperature - StartTemperature$
14: DetectMaxTemperature (ΔT_M, t_M)
15: **until** $time < 120\,\mathrm{s}$
16: Calculate κ using (2)
17: Calculate ρc using (3)
18: Calculate θ_v using (4)
19: Enter SLEEP mode (760 s)

Figure 5. Firmware algorithm

The presented values for the heater voltage (5 V), the heating duration (8 s) and the time allowed for the next measurement are the defaults. These values can be change as described next.

D. Digital Communications

The sensor uses SDI-12 protocol for digital communication of sensor data and change sensor parameters. SDI-12 is a standard to interface battery powered data recorders with micro-processor based sensors designed for environmental data acquisition. All SDI-12 communications are transmitted using American Standard Code for Information Interchange (ASCII) at 1200 baud with 7 data bits and an even parity bit. The standard also specifies a communications protocol that allows sensors to remain in a low-power sleep state until awoken by

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. SDI-12 SENSOR COMMANDS

Command	*Description*	*Command*	*Description*
?!	Return address of the sensor	aI!	Return sensor identification
aAb!	Change actual address to new 'b' address	aXGQ!	Return heating power, q', value
aM!	Start all measurements	aXGV!	Return heater applied voltage
aM1!	Start soil temperature measurement	aXSVn.n!	Assign heater voltage (from 2.0 to 5.0 V)
aD0!	Read measurements data	aXGT!	Return maximum temperature rise, ΔT_{M}
aHB!	Raw data of temperature curve	aXGH!	Return heat duration, t_0
aXGP1!	Return soil volumetric water content, θ_{v}	aXGI!	Return heater current
aXGP2!	Return soil thermal diffusivity, κ	aXGR!	Return spacing, r, value
aXGP3!	Return soil volumetric heat capacity ρc	aXSRn.nnn!	Assign new spacing value (from 5.500 to 6.500 mm)
aXGB!	Return soil bulk density, ρ_{b}	aXSHnn!	Change heat duration (from 6 to 30 s)
aXSBm.nn!	Set soil bulk density (from 0.90 to 2.00 $10^3\,\mathrm{kg\,m^{-3}}$)		

a serial break signal sent by the master. The first character of each command is a unique sensor address, `a`, that specifies with which sensor the recorder wants to communicate.

Table I lists all the commands that the sensor will respond to. In addition to the standard commands it was necessary to implement a set of extended commands (with an `X` after the address) in order to configure the sensor and obtain individual readings of some parameters. These extended commands use letter `G` after the `X` to get extra data from the sensor (heater voltage, heating power, heater current, spacing between heater and temperature sensor, maximum temperature rise, time of the maximum temperature rise, soil bulk density, and individual sensor parameters readings: soil volumetric water content, soil thermal diffusivity, and volumetric heat capacity), and a letter `S` to set new values for heater applied voltage and heat duration, to new spacing value, and to set new bulk density.

Access to metadata is also available as presented in version 1.4 of the SDI-12 specification [25]. For calibration and whenever necessary, the raw values (16 bit unsigned floating point format) of the temperature curve can be requested using the High Volume Binary Command (`aHB!`).

E. Probe Distance Calibration

The measurement of κ and ρc are highly sensitive to effective separation distances, r, between the heater element on heater probe and the thermistor on temperature sensor probe [9]. Thus, calibration of this distance was crucial for accurate measurements. The calibration was performed by inserting the sensor in a gel that was made from a $4 \times 10^{-3}\,\mathrm{kg\,L^{-1}}$ agar solution, Figure 6. The agar has equal thermal properties as water, yet does not create heat convection as would occur by heating liquid water [6].

The measured temperature response in agar solution was used to calibrate the sensor separation distance for each thermistor by fitting the measured temperature change to (1) using known values of volumetric heat capacity ($4174\,\mathrm{kJ\,m^{-3}\,K}$) and thermal diffusivity ($1.436 \times 10^7\,\mathrm{m^2\,s^{-1}}$) of water, as showed on Table II.

TABLE II. THERMAL PROPERTIES OF WATER AT 20 °C.

Physical Parameter	Value
ρ_w ($\mathrm{kg\,m^3}$)	998.2
c_w ($\mathrm{J\,kg^1\,K^{-1}}$)	4181.6
C_w ($\mathrm{kJ\,m^3\,K^{-1}}$)	4174
κ_w ($\mathrm{m^2\,s^{-1}}$)	$1.4{\times}10^{-7}$

F. Sensor Calibration

In this study, the developed soil moisture sensor has calibrated in laboratory conditions with a constant temperature of 20 °C using Tottori Dune sand because it allows for rapid saturation and drainage across a wide water content range. Physical properties are listed in Table III. In this study the method used for water content determination was the gravimetric method. The mass of water present in the sample is given

Figure 6. Probe distance calibration in agar solution

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Figure 7. Sensor calibration with Tottori sand.

by the difference in mass between the wet and dry sample.

Water content by mass is defined as the mass of water divided by the mass of dry soil. The conversion into volumetric equivalent can be done by multiplying the gravimetric water content by the dry bulk density of the sample (ρ_s), with a knowledge of water density (ρ_w). The volumetric water content was calculated using

$$\theta = \frac{m_w - m_d}{m_d}\rho_b \tag{6}$$

where ρ_b ($\mathrm{kg\,L^{-3}}$) is the dry bulk density to which the material is packed, and m_w (kg) and m_d (kg) are the wet and dry soil.

TABLE III. PHYSICAL PROPERTIES OF TOTTORI DUNE SAND. AFTER [4]

Physical Parameter	Value
ρ_b ($\mathrm{kg\,m^3}$)	1630
c_s ($\mathrm{J\,kg^{-1}\,K^{-1}}$)	795.0

The soil, Tottori Dune sand, is oven dried (105 °C) for 24 hours in order to get the *dry soil*. Then, soil was packed uniformly into a plastic container and was weighted (m_d=0.600 kg) and a measure was taken (dry soil measurement, θ_v =0.00 $\mathrm{m^3\,m^{-3}}$), as shown in Figure 7. After that, water is applied until saturation was achieved (weight of the wet soil was 0.732 kg). For this type of soil, saturation was achieved with $\theta_v = 0.36\,\mathrm{m^3\,m^{-3}}$ using (6) and data from Table III. With the soil sample at saturation a measurement was made, followed by three more measurements as the soil sample was drying in the air.

The values of sample weight, soil gravimetric water content (θ_G), and soil volumetric water content (θ_v) are summarized on Table IV.

TABLE IV. SOIL WEIGHT VALUES AND CALCULATED VOLUMETRIC WATER CONTENT

Soil Weight (kg)	θ_G ($\mathrm{kg\,kg^{-1}}$)	θ_v ($\mathrm{m^3\,m^{-3}}$)
0.600	0.000	0.000
0.630	0.050	0.082
0.650	0.083	0.136
0.700	0.167	0.273
0.732	0.220	0.359

V. RESULTS AND DISCUSSION

Since this work is only about a description of the design and the construction of the sensor, only the results about the sensor assembly and the calibration of the distance between the probes will be discussed.

A. Printed Circuit Board

The developed Printed Circuit Board (PCB) is shown in Figure 8. In order to optimize the space provided by the supplier (Seeed Studio, China) that is 100 mm × 100 mm, it was possible to place three sensors with mechanical joints that after welding all components will be cut. This process also reduces costs by 1/3 of PCB production, with the production of 30 PCB for about 15 €.

B. Assembled Sensor

Figure 9 shows the assembled sensor without the thermoplastic molding resin and the final version of the sensor. The sensor is small in size, compact, robust and easy to use. The final product is comparable to commercial versions of other soil moisture sensors using other measuring methods, such as the EC-5, 5TE and 5TM probes from Decagon Devices Inc., USA.

This type of sensor (DPHP sensor) compares with capacitive type of sensor in terms of accuracy (±3 % [26]), and is very different from very low-cost resistive sensor. Resistive sensors give only qualitative estimation of the moisture content [26].

C. Probe Distance Calibration

The probe distance calibration was determined by obtaining five sensor readings in agar solution with an interval of 1 h between readings and with heating power of $q' = 51.65\,\mathrm{W\,m^{-1}}$. Figure 10 presents the measured temperature response as a function of measurement time of one of these readings. The measured temperature data was fitted using the non-linear least-squares Marquardt-Levenberg algorithm. There is a good agreement between measured and fitted data. The fitted values of effective separation distance, r_{eff}, for all 5 readings are presented in Table V. The average value of r_{eff} is 5.534×10^{-3} m. The differences between the designed PCB layout value (6×10^{-3} m) and effective distances are likely caused by unprecise placement of the components (heater and/or thermistor) because they are hand welded. In an industrial process this error could be minimized and lead to two kinds of sensors: agar calibrated and uncalibrated lowering the costs.

It can also be observed in Figure 10 that, compared to other works [4], [8], [27], just to mention a few, which uses

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

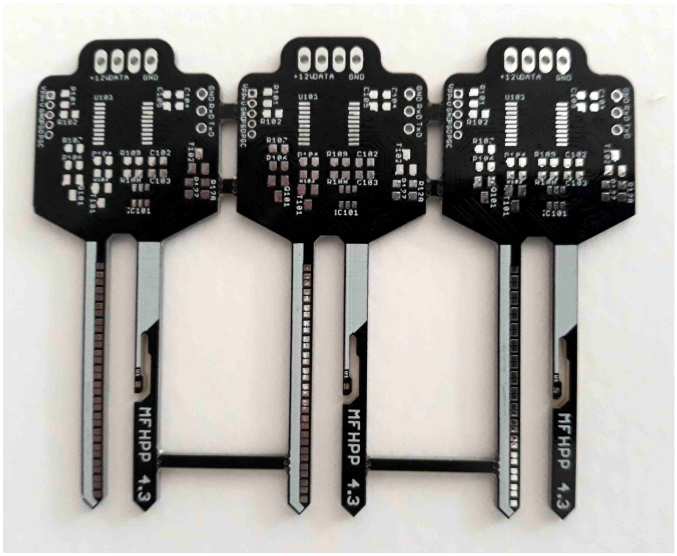


Figure 8. Developed Printed Circuit Board

data-logger for reading and control of the DPHP sensor, this prototype presents a better resolution (18 bit against 13 bit of must data-loggers) and a better sampling rate (1 sample/s against 3 samples/s). This will give a better precision on calculating ΔT_{M} and t_{M}. We can also observe a good signal to noise ratio ($\Delta T_{\mathrm{M}} = 0.3728\,^{\circ}\mathrm{C}$ for a value of 100 % of θ_{v}, agar $\Rightarrow$ water) even for a lower heat power ($51.65\,\mathrm{W\,m^{-1}}$ against $60\,\mathrm{W\,m^{-1}}$ - minimum found in literature [8]). This low power (could be less and adjusted depending on θ_{v} values - lower values less power) is very important in order to use the sensor in wireless systems (Internet of Things - IoT).

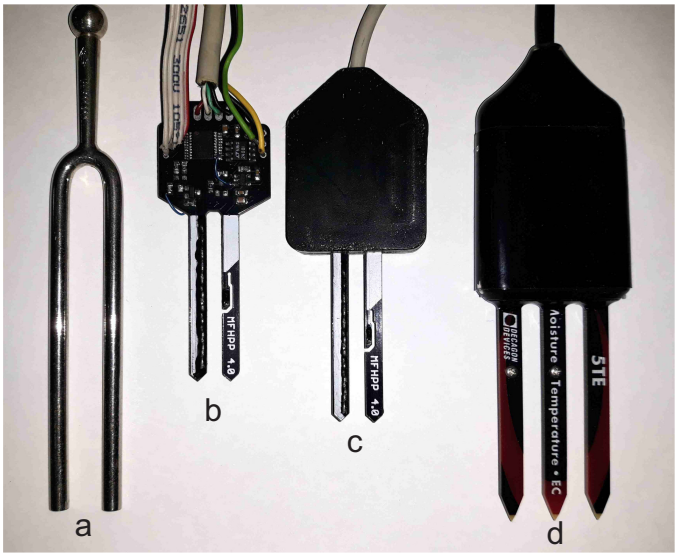


Figure 9. Developed prototypes of the DPHP sensor. a) Diapason; b) Prototype for debug without protective resin; c) Final prototype with protective resin; d) Decagon 5TE.

TABLE V. CALIBRATION OF THE EFFECTIVE SPACING, r_{eff}

Sensor Readings	$r(\mathrm{m})$	$r_{\mathrm{eff}}(\mathrm{m})$
#1	0.006	0.005566
#2		0.005522
#3		0.005521
#4		0.005512
#5		0.005551
Average		**0.005534**

D. Sensor Calibration

For sensor calibration the sensor was connected to an Arduino development board according to the wiring diagram of Figure 11. The SDI-12 listed commands from Table I can be sent through the serial port (USB) and the Arduino software (using the Arduino library for SDI-12 communications [28]) is responsible for converting to the SDI-12 protocol for the sensor.

The calibration of the sensor was done according to the method described in Section IV-F. When the values indicated on the scale were in accordance with the desired values, three sensor readings were taken with half an hour interval between

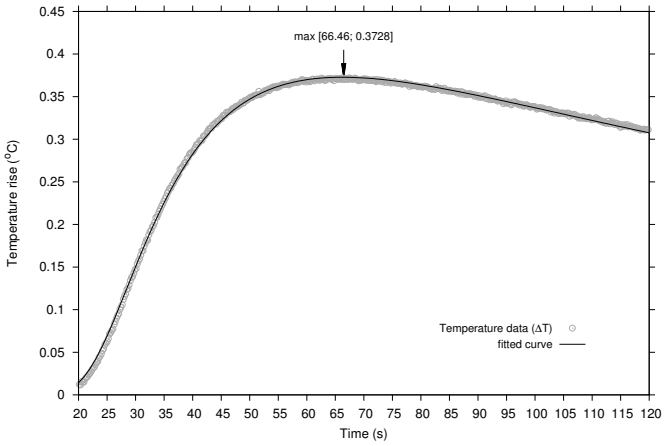


Figure 10. Measured temperature rise after the heat pulse in agar solution compared to the analytic modeled data (solid line).

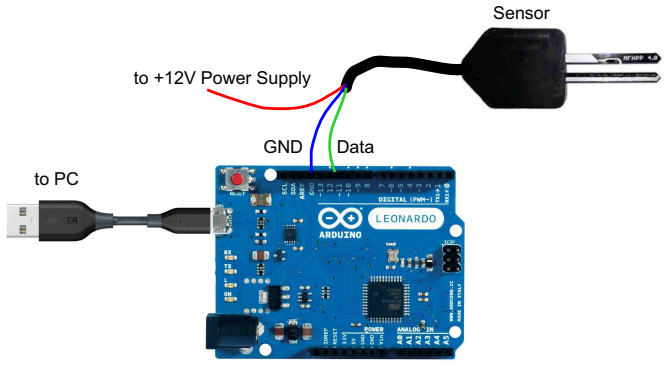


Figure 11. Sensor to Arduino connections for SDI-12 protocol conversion.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE VI. OBTAINED AND CALCULATED VALUES FROM FIRMWARE IN THE SENSOR

	Obtained Values		*Calculated Values*		
θ_v from calibration ($\mathrm{m^3\,m^{-3}}$)	ΔT_M (°C)	t_M (s)	κ ($\mathrm{m^2\,s^{-1}}$)	ρc ($\mathrm{J\,m^{-3}\,°C^{-1}}$)	θ_v from sensor ($\mathrm{m^3\,m^{-3}}$)
0.000	1.130410			1.3150	0.0042
0.082	0.908076			1.6370	0.0816
0.136	0.788853	17.8317	6.8981	1.8844	0.1396
0.265	0.613252			2.4239	0.2684
0.359	0.525757			2.8273	0.3668

each one. Readings are made by sending the command `0M!` followed by the command `0XGP2!`, sent after the time indicated in the response to the command `0M1` (130 s).

The results obtained for the five calibration points presented in Table IV, are represented in Figure 12.

As can be seen, for the Tottori Dune sand measurement range ($0\,\mathrm{m^3\,m^{-3}}$ to $36\,\mathrm{m^3\,m^{-3}}$), the temperature rise value is only $0.65\,°\mathrm{C}$. In order to obtain, in comparative terms, a precision close to commercial capacity-based sensors ($3\,\%$), the accuracy of the temperature reading should be better than $0.05\,°\mathrm{C}$ which is the accuracy achieved by the developed sensor.

During the acquisition of temperature, and according to the algorithm presented in Figure 5, the maximum value of the temperature rise, ΔT_M, and the time at which this maximum happens, t_M, were obtained from temperature reads and microcontroller timer. With these values and using the equations (2), (3) and (4), the values presented in Table VI were calculated o microcontroller firmware.

It is important to note that the values of t_M, and consequently the value of κ (by (2)), can be verified that κ is only dependent on t_M since t_0, the time of the heat pulse, and r, the distance between the heater element and sensor are constant and this last was subjected to a calibration process to minimize errors) are constant for the same type of soil. In this way, the firmware obtains the t_M value through a sliding average.

To verify the accuracy of the water content measurements, Figure 13 compares the results of water content from the sensor and calibration samples obtained with the method explained on Section IV-F. It is possible verify that the values for the sensor tted a linear relationship with $R^2 = 0.9995$ and $RMSE = 0.003\,\mathrm{m^3\,m^{-3}}$ over the entire range of θ_v from dry to saturation. With the mention two-step calibration the developed sensor achieved high-accuracy water content measurements.

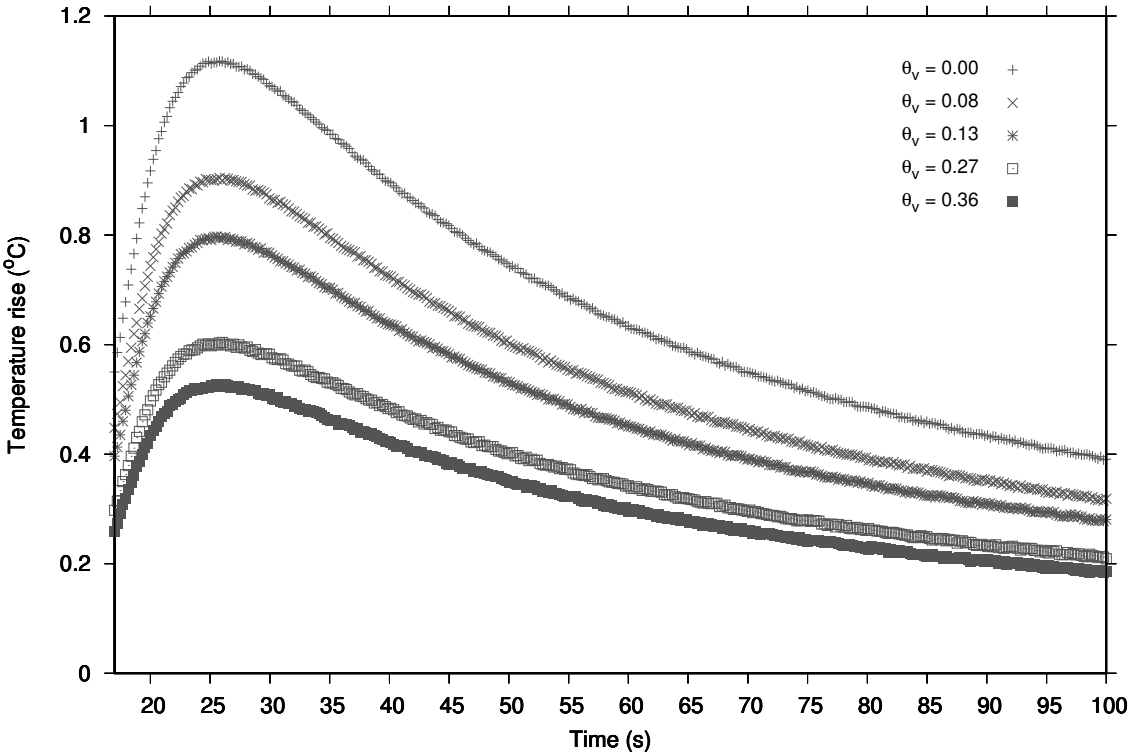


Figure 12. Measured temperature rise after the heat pulse in Tottori sand for five moisture contents.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

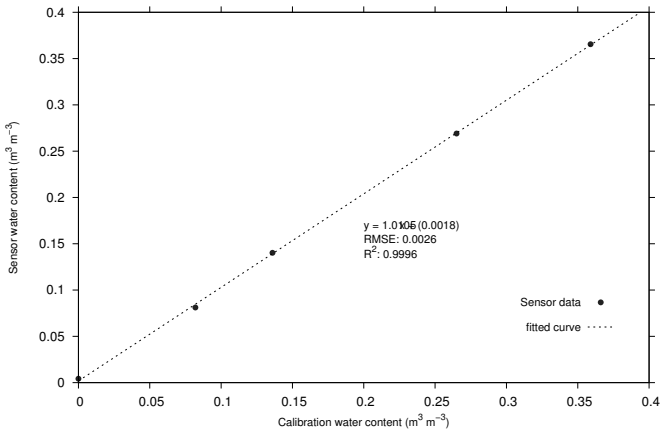


Figure 13. Comparison of volumetric water content measured by the sensor and the calibration water content.

VI. Conclusion and Future Work

This paper presents a novel design and an industrial process to build a DPHP sensor based on PCB board as substrate. The process is based on four steps and two-step calibration process: in agar for probe distance determination, and using Tottori Dume sand for sensor validation. The prediction accuracy of soil volumetric water content, θ_v by the developed sensor was validated as $0.003\,\mathrm{m}^3\,\mathrm{m}^{-3}$ for Tottoti Dune sand with θ_v varying from dry to saturation.

Results also show that it is possible to build the multi-functional DPHP sensor in a low cost industrial process (PCB and assembly). This was the first time that a soil moisture sensor, based on heat-pulse method, was build using a PCB as support. Further work must be done to find out the sensor accuracy, test sensor readings variations with soil type, and perform in-field and long-term stability tests.

Future work must needed to study the thermal properties determined under various thermal conditions to find any impact of the ambient temperature on the performance of the sensor. Also, the use of several sensors connected on the same data-logger must be evaluated on power consumption and communications stability.

The authors are producing about 48 sensors for use in the field and connected through SDI-12 to two data loggers with battery and solar support. The results of this study will be published in future work.

In addition, as the sensor has a PCB as substrate, pads can be placed to measure soil moisture using the capacitive method and to measure soil Electrical Conductivity (EC). This process may lead to a system with sensory fusion. The capacitive method, with lower energy consumption, would be the main method to be used and once or twice a day an acquisition by the heat pulse method would be made. By using, for example, a complementary filter, sensory fusion would be performed. In this way, it could have a precise sensor, of low cost and with less energy consumption.

Acknowledgment

This work is financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project “POCI- 01-0145-FEDER-006961”, and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

[1] A. Valente, A. A. Saraiva, N. F. Ferreira, and S. Soares, “On the design and construction of dual-probe heat-pulse soil moisture sensor: Towards an industrial solution,” in ALLSENSORS 2018, vol. 1, 2018, pp. 43–48.

[2] W. Sheng, K. Rumana, M. Sakai, F. Silfa, and S. B. Jones, “A Multi-Functional Penta-Needle Thermo-Dielectric Sensor for Porous Media Sensing,” IEEE Sensors Journal, vol. 16, no. 10, 2016, pp. 3670–3678.

[3] M. Inoue, J. imnek, S. Shiozawa, and J. Hopmans, “Simultaneous estimation of soil hydraulic and solute transport parameters from transient infiltration experiments,” Advances in Water Resources, vol. 23, no. 7, 2000, pp. 677 – 688. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0309170800000117

[4] Y. Mori, J. W. Hopmans, A. P. Mortensen, and G. J. Kluitenberg, “Multi-functional heat pulse probe for the simultaneous measurement of soil water content, solute concentration, and heat transport parameters,” Vadose Zone Journal, vol. 2, no. 4, 2003, p. 561.

[5] Y. Mori, J. Hopmans, A. Mortensen, and G. Kluitenberg, “Estimation of vadose zone water flux from multi-functional heat pulse probe measurements,” Soil Sci. Soc. Am. J., vol. 69, no. 3, 2005, pp. 599–606.

[6] G. S. Campbell, K. Calissendorff, and J. H. Williams, “Probe for measuring soil specific heat using a heat pulse method,” Soil Sci. Soc. Am. j., vol. 55, 1991, pp. 291–293.

[7] G. J. Kluitenberg, J. Ham, and K. L. Bristow, “Error analysis of the heat-pulse method for measuring soil volumetric heat capacity,” Soil Sci. Am. J., vol. 57, 1993, pp. 1444–1451.

[8] K. Bristow, G. Kluitenberg, and R. Horton, “Measurement of soil thermal properties with a dual-probe heat-pulse technique,” Soil Sci. Soc. Am. j., vol. 58, 1994, pp. 1288–1294.

[9] G. J. Kluitenberg, K. L. Bristow, and B. S. Das, “Error analysis of the heat pulse method for measuring soil heat capacity, diffusivity, and conductivity,” Soil Sci. Soc. Am. J, vol. 59, 1995, pp. 719–726.

[10] S. Welch, G. Kluitenberg, and K. Bristow, “Rapid numerical estimation of soil thermal properties for a broad class of heat-pulse emitter geometries,” Measurement Science & Technology, vol. 7, no. 6, 1996, pp. 932–938.

[11] J. Tarara and J. Ham, “Measuring soil water content in the laboratory and field with dual-probe heat-capacity sensors,” Agronomy Journal, vol. 89, no. 4, 1997, pp. 535–542.

[12] T. Ren, K. Noborio, and R. Horton, “Measuring soil water content, electrical conductivity, and thermal properties with a thermo-time domain reflectometry probe,” SOIL SCIENCE SOCIETY OF AMERICA JOURNAL, vol. 63, no. 3, 1999, pp. 450–457.

[13] K. Bristow, G. Kluitenberg, C. Goding, and T. Fitzgerald, “A small multi-needle probe for measuring soil thermal properties, water content and electrical conductivity,” Computers and Electronnics in Agriculture, vol. 31, no. 3, 2001, pp. 265–280.

[14] T. Ren, T. E. Ochsner, and R. Horton, “Development of thermo-time domain reflectometry for vadose zone measurements,” Vadose Zone Journal, vol. 2, 2003, pp. 544–551.

[15] A. Valente, R. Morais, C. Couto, and J. Correia, “Modeling, simulation and testing of a silicon soil moisture sensor based on the dual-probe heat-pulse method,” Sensors and Actuators A:Physical, vol. 115, no. 2-3, 2004, pp. 434–439.

[16] T. Kamai, G. J. Kluitenberg, and J. W. Hopmans, “Design and Numerical Analysis of a Button Heat Pulse Probe for Soil Water Content Measurement,” Vadose Zone Journal, vol. 8, no. 1, 2009, pp. 167–173.

[17] A. Valente, J. Cunha, J. Correia, and C. Couto, “A silicon probe with integrated microelectronics for soil moisture measurements,” in Proc. of WCCA 2001, 2001, pp. 440–446.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[18] A. Valente, C. Couto, and J. Correia, “On-chip integrated silicon bulk-micromachined soil moisture sensor based on the DPHP method,” in Transducers ’01: Eurosensors XV, vol. 1, 2001, pp. 316–319.

[19] A. Valente, R. Morais, C. Couto, and J. H. Correia, “Modeling and simulation of a silicon soil moisture sensor based on the DPHP method for agriculture,” EUROPEAN CONFERENCE ON SOLID-STATE TRANSDUCERS, no. 2, 2003, pp. 564–567.

[20] A. Valente, S. Soares, R. Morais, J. M. Baptista, and M. Cabral, “Button heat-pulse sensor for soil water content measurements,” Proceedings - 1st International Conference on Sensor Device Technologies and Applications, SENSORDEVICES 2010, 2010, pp. 180–182.

[21] A. Valente, R. Morais, J. Boaventura, and C. José, “A multi-chip-module micro-system for soil moisture,” in EFITA 2003 Conference, no. July, Debrecen, Hungary, 2003, pp. 547–551.

[22] M. Abramowitz and I. Stegum, Handbook of Mathematical Functions. Dover Publ., 1972.

[23] G. Campbell, “Soil physics with basic, transport models for soil plant systems,” Elsevier, 1985.

[24] W. J. Cody and H. C. Thacher, “Chebyshev approximations for the exponential integral ei(x),” Mathematics of Computation, vol. 23, no. 106, 1969, pp. 289–303. [Online]. Available: http://www.jstor.org/stable/2004423

[25] SDI-12 Support Group, “SDI-12 a Serial-Digital Interface standard for microprocessor-based sensors,” SDI-12 Support Group, Version 1.4, May 2017, [retrieved: January, 2018]. [Online]. Available: http://www.sdi-12.org/specification.php?file_id=1

[26] P. Aravind, M. Gurav, A. Mehta, R. Shelar, J. John, V. S. Palaparthy, K. K. Singh, S. Sarik, and M. S. Baghini, “A wireless multi-sensor system for soil moisture measurement,” in 2015 IEEE SENSORS, Nov. 2015, pp. 1–4.

[27] J. M. Tarara and J. M. Ham, “Measuring soil water content in the laboratory and field with dual-probe heat-capacity sensors,” Agronomy Journal, vol. 89, no. 4, 1997, pp. 535–542.

[28] S. W. R. Center. Arduino library for SDI-12 communications. [Online]. Available: https://github.com/EnviroDIY/Arduino-SDI-12/issues/17 (2013) Last visited on 2018-11-26.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A Flexible QoS Measurement Platform for Service-based Systems

Andreas Hausotter, Arne Koschel
University of Applied Sciences & Arts Hannover
Faculty IV, Department of Computer Science,
Hannover, Germany
email: Andreas.Hausotter@hs-hannover.de
email: Arne.Koschel@hs-hannover.de

Johannes Busch, Malte Zuch
University of Applied Sciences & Arts Hannover
Faculty IV, Department of Computer Science,
Hannover, Germany
email: Johannes.Busch@stud.hs-hannover.de
email: Malte.Zuch@hs-hannover.de

***Abstract*—The transfer of historically grown monolithic software architectures into modern service-oriented architectures creates a lot of loose coupling points. This can lead to an unforeseen system behavior and can significantly impede those continuous modernization processes, since it is not clear where bottlenecks in a system arise. It is therefore necessary to monitor such modernization processes with an adaptive monitoring concept to be able to correctly record and interpret unpredictable system dynamics. This contribution presents a generic QoS measurement framework for service-based systems. The framework consists of an XML-based specification for the measurement to be performed – the Information Model (IM) – and the QoS System, which provides an execution platform for the IM. The framework will be applied to a standard business process of the German insurance industry, and the concepts of the IM and their mapping to artifacts of the QoS System will be presented. Furthermore, design and implementation of the QoS System's parser and generator module and the generated artifacts are explained in detail, e.g., event model, agents, measurement module and analyzer module.**

Keywords–Quality of Service (QoS); Indicator Measurement; XML-Model; Service-orientation; SOA; Complex Event Processing (CEP)

I. INTRODUCTION

The background of this work is a cooperation with a partner from the German insurance industry and its IT-Architecture department. Many IT-driven and data-driven companies face the challenge of continually modernizing their infrastructure, technologies, systems and processes. The insurance industry in particular is characterized by the fact that extensive digitization of processes took place very early. This was done well before researching modern service-based approaches, such as 'traditional' service-oriented architectures (SOA) or even microservices (MS) and without the use of distributed infrastructures such as cloud computing. Historically grown software monoliths were state of the art. The modernization of such monoliths in the direction of service-based architectures is a major challenge. This conversion process is the main motivation of this work and will be explained in more detail below.

A. Motivation

Systems cannot be abruptly switched off and replaced by new architectures but must be continuously transformed into modern architectural forms. In this continuous modernization process, monolithic structures are broken down and distributed into services. This gives companies more agility and adaptability to changing business requirements. However, a decentralized and service-oriented system architecture is usually quite fine-granular and loosely coupled. Generally, this provokes an unpredictable dynamic system behavior. This also applies to our partner in the insurance industry. In order to remain competitive, the insurance industry has to respond quickly to customer information portals, such as check24.de, where different insurance companies competitively can offer, e.g., car insurances. This scenario motivates the need for a holistic measurement concept and defines the general application scenario of this work.

So, there is a fundamental need for information about the system behavior. Relevant information is collected in the 'Information Product', which represents the output of the 'Core Measurement Process' (cf. Fig. 1). The 'Information Need' provides the input for the subprocess 'Plan the Measurement Process', the subprocess 'Perform the Measurement Process' generates the Information Product'. The process goal is to satisfy the 'Information Need'.

Nowadays it is normal that customers are demanding online services unpredictably and with high volatility. These volatile demands may lead to bottlenecks in distributed service-oriented architectures. Therefore, a reliable measurement of the whole system behavior is necessary in order to eliminate any bottlenecks. Such a measurement concept and its prototypical implementation are the core contributions of our work.

B. Contribution

In order to monitor individual system components with respect to time behavior, fixed time limits have so far been used. These fixed time limits are often used in historically grown software systems of the German insurance industry. If a system component (service) could not respond within these time limits, this was interpreted as a bad quality feature. However, with these static limits, a dynamic system behavior can be poorly monitored and interpreted. The challenge is to determine, when dynamic systems are overloaded. In this respect, a partner company of the insurance industry demands to integrate a metric, which could replace their static time limits in the future with a more dynamic metric. The general requirements lead to the following questions:

- How could static rules and timeouts be supplemented by a dynamic measurement metric?
- How could the measuring system be built on existing XML-Standards?

In previous work [1] [2] [3], we have already developed a framework for dynamically measuring the service response

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

time as a Quality of Service (QoS) Parameter within service-oriented architectures. Especially in [1] we provided initial implementation details of the dynamic measuring system. Our measuring system considers existing XML standards and can flexibly record the load behavior of a software system. This measuring system should measure the response time as a particular QoS parameter as an example. The measuring system should be able to consider both dynamic limits as well as static limits (optional). Normally, only the dynamic limits should be considered. But, if a service exceeds a fixed limit of, e.g., 5 seconds, then this should also be recognized. Another requirement is that the measuring system should 'inject' measurement agents into a software system as flexible and automated as possible.

As a significant extension to our previous articles, here we contribute in more details in important areas of our work, namely in Section IV and in Section V we show:

- a detailed design model of our QoS generator including an in depth look at its general parser classes as well as its derived measure parser classes,
- our in-memory model for our base XML model,
- much more implementation details on the measurement module and our event model (EventModel),
- and a comprehensive summary of the overall QoS platform based on our previous articles.

In total, our additional contributions provide a much deeper look at our work with respect to design and implementation of our overall system.

The remainder of this article is organized as follows: In Section II, related work concerning the topic of measurement models of service-based systems is explained. The measurement process with its core concepts and the information model are described in Section III and more implementation details in Section IV. Some mathematical equation explains the general measurement concept. After clarifying the general measurement plan, Section VI shows how the planned measurement concept can be applied for detecting the so called 'Spikes', situations of high system-loads. The final Section VII will summarize this work. The different advantages and disadvantages of the described measurement model will be discussed. Also, an outlook to future work will show how the results of this work will be used in upcoming work in Section VII.

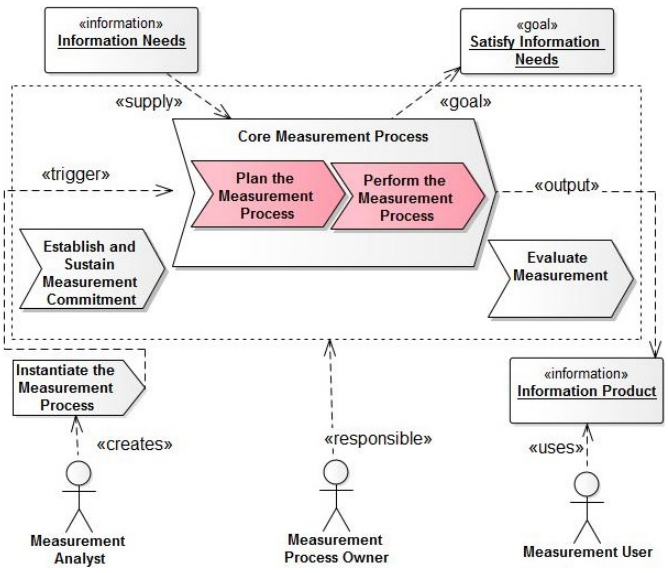


Figure 1. The Core Measurement Process

II. Prior and Related Work

In prior work, we already discussed several aspects of the combination of SOA, Business Process Management (BPM), Workflow Management Systems (WfMS), Business Rules Management (BRM), and Business Activity Monitoring (BAM) [4][5][6] as well as Distributed Event Monitoring and Distributed Event-Condition-Action (ECA) rule processing [7][8]. Building on this experience, we now address the area of QoS measurement for combined BRM, BPM, and SOA environments, mainly but not limited to, within the (German) insurance domain.

Work related to our research falls into several categories. We will discuss these categories in sequence.

General work on (event) monitoring has a long history (cf. [9][10] or the ACM DEBS conference series for overviews). Monitoring techniques in such (distributed) event-based systems are well understood, thus such work can well contribute general monitoring principles to the work presented here. This also includes commercial solutions, such as the Dynatrace [11] system or open source monitoring software like, for example, the NAGIOS [12] solution. In these systems there is generally no focus on QoS measurement within SOAs. Also, they usually do not take application domain specific requirements into account (as we do with the insurance domain).

Active Database Management Systems (ADBMS) offer some elements for use in our work (see [13][14] for overviews). Event monitoring techniques in ADBMSs are partially useful, but concentrate mostly on monitoring ADBMS internal events, and tend to neglect external and heterogeneous event sources. A major contribution of ADBMSs is their very well defined and proven semantics for definition and execution of Event-Condition-Action (ECA) rules. This leads to general classifications for parameters and options in ADBMS core functionality [14]. We may capture options that are relevant to event monitoring within parts of our general event model. QoS aspects are handled within ADBMS, for example, within the context of database transactions. Since ADBMSs mostly do not concentrate on heterogeneity (and distribution), let alone SOAs, our research work extends into such directions.

The closest relationship to our research is the work, which directly combines the aspects QoS and SOA. As many as 2002 several articles fall into this category. However, in almost all known articles the SOA part focuses on WS-* technologies. This is in contrast to our work, which takes the operational environment of our insurance industry partners into account.

Examples of Webservice (WS-*) related QoS work include QoS-based dynamic service bind [15][16], related WS-* standards such as WS-Policy [17], and general research questions for QoS in SOA environments [18]. Design aspects and models for QoS and SOA are, for example, addressed in [15][19][20][21][22]. As for WS-* Web services, we also take XML as foundational modelling language for our work. SOA performance including QoS is discussed in articles [23], and monitoring for SOA in articles such as [24][25][26][27].

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Uniqueness of our research is that it takes all the above-mentioned aspects into account. We provide a detailed XML based measurement model, as well as a generator-supported, generic SOA monitoring framework. All of it takes especially the operational environment of our insurance industry partners into account, which is a large-scale SOA, but only partially WS-* technology based. This makes our work highly relevant in practice. Even more, since we base our modelling on standards, which are highly relevant for German insurance businesses (cf. VAA [28], ISO/IEC 9126 [29][30]), our work is of a quite general nature and thus can be transferable (at least within the insurance domain).

III. Plan the Measurement Process

The Core Measurement Process can be divided into two parts. First of all, the planning of the measurements takes place, which determines how the Information Need can be answered. In the second part, the planned methods of measurement will be implemented.

A. Core Concepts of the Abstract Information Model

To measure the response time behavior of a dynamic system, the definition of static response time limits is often not sufficient. When a system component (service) is deployed in a different hardware environment or in a different cloud environment, this will affect the response time of this system component. Static limits would have to be adapted manually to the new execution environment of the services. Furthermore, individual services share hardware resources with many other services. This can lead to an unpredictable system behavior, especially in complex business processes. Therefore, static limits are not sufficient, but a more flexible solution is required. The approach of this work is the investigation of a measurement concept, which is more flexible and based on the standard deviation of system load of a specifiable measuring period.

The insurance industry in particular is characterized by strong seasonal fluctuations. Towards the end of the year, many customers switch their insurance contracts and are provoking high system loads. In times of such high system loads, the mentioned static limits would be continuously exceeding. The information would be lost at the time when high loads are peaking in such a strongly demanded period. It is important to know when the current system is heavily loaded. Knowledge about this information represents the so-called Information Need (Fig. 1) of our partner from the insurance industry.

To answer this Information Need, the average response time behavior μ of a system component is firstly computed for a freely definable time period. For example, on the basis of the last $n = 500$ measured response times of the services. On the basis of this, the standard deviation is calculated within this period, shown in (1):

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1}} \tag{1}$$

After this calculation, the current response time r of a service is set in relation to this standard deviation s. If the response time r of a currently requested service exceeds this standard deviation by the factor of 2x then this is considered as an overload situation:

$$\text{Spike detected: } r > \mu + 2 * s \tag{2}$$

This calculation takes place continuously. As soon as a service is requested again, its response time is recorded and set in relation to the last one (e.g., the last 500 measured values). It is therefore a continuous and rolling measurement. This measuring system can be applied both for very slow system components on a daily base and also to very fine-granular services that interact in the range of milliseconds.

The important fact is that the standard deviation is calculated continuously over a defined time period, and the current response time of a service is set in relation to this. Therefore, the measuring system adapts to seasonal fluctuations, and it is possible to identify, which user requests (service calls) are currently very critical with respect to the general response time behavior, independently of the prevailing current load situation. This allows fast and more precise analysis of systems and less misinterpretation due to incorrectly set static time limits. This dynamic measurement concept can give a more reliable answer to the Information Need of our project partners.

B. Mapping of the Concepts of the Information Model

In this subsection, a QoS Information Model (QoS IM) is presented in a more detailed manner. The QoS IM is a XML document that includes values of the concepts for a given application scenario. The concepts and their relationships with each other are introduced in [3]). Here we focus on the implementation of the concepts.

The QoS IM is created during the planning stage when executing the subprocess 'Plan the Measurement Process', cf. Fig. 1). The XML document is used to automatically generate the QoS Platform's artefacts. The measurements results (i.e., the output of 'Perform the Measurement Process') are produced by the QoS Platform. They are persistently stored for subsequent analysis, typically in a database system.

We opted for XML as universally accepted standard which is highly flexible, platform and vendor independent and supported by a wide variety of tools. Furthermore, XML comes with a standardized schema definition language, namely XML Schema. This is a big advantage against other languages such as JSON for example.

In the QoS IM, we specify the measurement concepts for the check24.com scenario, or the Proposal Service respectively. Due to space limitation, the discussion is restricted to the following concepts (cf. [3]):

- Measurable Concept – outlines in an abstract way, how the Quality Attributes are determined to satisfy the Information Need,
- Base Measure – specifies by its Measurement Method how the value of Quality Attribute is to be determined,
- Derived Measure – uses one or more Base Measures or other Derived Measures, whilst the Measurement Function specifies the calculation method and thus the combination of the Measures used,
- Indicator – is a qualitative evaluation of Quality Attributes, which directly addresses the issue raised in the Information Needs.

The Measurable Concept `Processing_Time` references

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
<MeasurableConcept Name="Processing_Time">
 <SubCharacteristic Name="Performance"/>
 <BaseMeasure Name="t_inst"/>
 <BaseMeasure Name="t_term"/>
 <DerivedMeasure Name="t_proc"/>
 <DerivedMeasure Name="Count_StdDev_Calls"/>
 <DerivedMeasure Name="Count_Calls"/>
 <DerivedMeasure Name="StdDev_Calls_Percentage"/>
 <DerivedMeasure Name="Failed_Calls"/>
</MeasurableConcept>
```

Listing 1. Calculation of the Proposal Service's Processing Time

```
<BaseMeasure Name="t_inst">
 <Scale TypeOfScale="Rational" Type="R"/>
 <Attribute ServiceID="BAS_001" AttributeName="ServiceCallID"/>
 <MeasurementMethod Name="recordTimeOfServiceCall">
 <Implementation>
  <Agent Class="ServiceAgent">
  <Method>
  <Attribute Name="ServiceCallID" Type="xs:integer"/>
  <Attribute Name="Time" Type="xs:string" Computed="time"/>
  <Event Name="ServiceStartEvent"/>
  </Method>
  </Agent>
 </Implementation>
 </MeasurementMethod>
</BaseMeasure>
```

Listing 2. Start Time of a Proposal Service Call

by name all necessary Base and Derived Measures (cf. listing 1).

The definition of the Base Measure `t_inst` is shown in listing 2. Its task is to capture the start time of a Proposal Service call (by a user request). The element `Attribute` specifies the attribute of the Proposal Service to be observed. The element hierarchy of `Implementation` defines all platform specific information to automatically generate all artefacts needed for the measurement, i.e., the agent class with attributes and the measurement method (cf. subsection IV).

The Derived Measure `Count_StdDev_Calls` presented in listing 3 calculates the number of Proposal Service calls that exceeds twice the standard deviation (cf. subsection IV, (2)).

`Count_StdDev_Calls` is based on a different Derived Measure, namely `t_proc`, which computes the processing time of a Proposal Service call (cf. `Uses` element, line 2). The element `Implementation` comprises of all information that is used to generate the analyzer class (cf. subsection IV). The analyzer executes the SQL Select statement (cf. lines 8 to 16), which represents the content of the element `Plain`. This is done by the measurement function `calculateNumberOfCallsAboveSTDDEV`, shown in line 3, whenever an event `ServiceDurationEvent` has been fired (cf. line 6).

```
<DerivedMeasure Name="Count_StdDev_Calls">
 <Uses><DerivedMeasure Name="t_proc"/></Uses>
 <MeasurementFunction Name="calculateNumberOfCallsAboveSTDDEV">
  <Implementation>
   <Analyzer>
    <Query Class="ServiceDuration" Type="xs:long">
     <Plain>
     SELECT COUNT(*) FROM serviceduration
     WHERE
     TINTS > TIME_SECS(DATEADD('DAY',
                 -30,NOW()))
     AND TPROC > (SELECT AVG(TPROC)
                       + (2 * STDDEV(TPROC))
     FROM serviceduration
       WHERE TINTS > TIME_SECS(
           DATEADD('DAY',-30, NOW())))
     </Plain>
    </Query>
   </Analyzer>
  </Implementation>
 </MeasurementFunction>
 <UnitOfMeasurement>ms</UnitOfMeasurement>
 <TargetValue>1</TargetValue>
</DerivedMeasure>
```

Listing 3. Compute the Number of Proposal Service Calls that Exceed Twice the Standard Deviation

Finally, the Indicator `SLoT_proc`, shown in listing 4, evaluates the adequacy of the processing time of all Proposal Service calls.

`SLoT_proc` is based on two different Derived Measures, namely `StdDev_Calls_Percentage`, and `Failed_Calls` respectively (cf. `Uses` element, lines 4 to 7). The first measure, `StdDev_Calls_Percentage`, takes `Count_StdDev_Calls` and `Count_Calls` and does some basic arithmetic computation.

The element `DecisionCriteria` specifies a decision table, so that a value, computed by the Derived Measures, can be mapped to the entry of the given nominal scale (i.e., high, medium, low). The element `Implementation` comprises all information to generate the analyzer class (cf. subsection IV), which implements the decision table and the mapping.

IV. Design of the QoS Generator

The initial phases of applying an IM (cf. Fig. 2) were shown in Section III-B. This section discusses subsequent phases (especially about generators, artefacts, etc.) in detail. Please note, although its concepts are transferable, our QoS Generator aims not to be of generic nature but is tailored specifically towards our XML based IM and needs of our partner companies. Furthermore, the generated artefacts are specific to our current QoS Platform. Both offer the flexibility to tailor each part to the specific needs of each of our partner companies.

Several different artefacts have to be generated to apply a specific IM. The basic design of the QoS Generator is given in

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Figure 2. Phase from IM to QoS System.

```
<Indicator Name="SLoT_proc">
 <AnalysisModel Name="computeAdequacyOfProcessingTime">
 <Scale TypeOfScale="Nominal" Type=.../>
 <Uses>
 <DerivedMeasure Name="StdDev_Calls_Percentage"/>
 <DerivedMeasure Name="Failed_Calls"/>
 </Uses>
 <DecisionCriteria>
 <Implementation>
  <Analyzer>
  <IndicatorTable Class="IndicatorController" Type="HMN">
  <IndicatorEntry>
   <Input>devPercentageCount < 5 && badCount == 0</Input>
   <Result>low</Result>
  </IndicatorEntry>
  <IndicatorEntry>
   <Input>devPercentageCount >= 5 && badCount == 0</Input>
   <Result>mittel</Result>
  </IndicatorEntry>
  <IndicatorEntry>
   <Input></Input>
   <Result>hoch</Result>
  </IndicatorEntry>
  </IndicatorTable>
  </Analyzer>
 </Implementation>
 </DecisionCriteria>
 </AnalysisModel>
</Indicator>
```

Listing 4. Compute the Adequacy of the Processing Time of all Proposal Service Calls

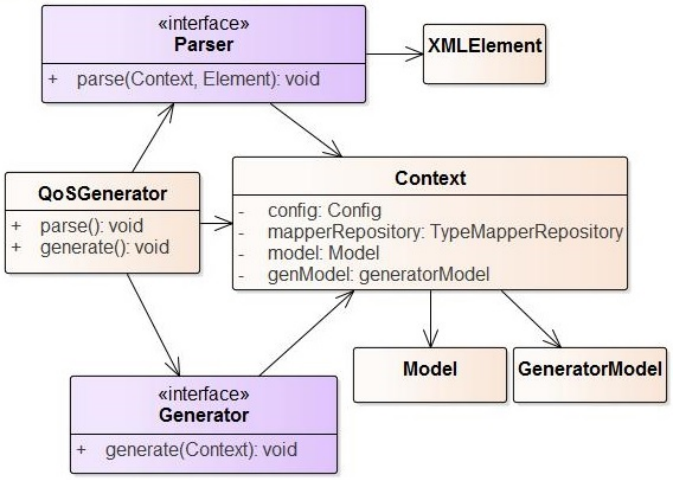


Figure 3. Design of the QoS Generator.

Fig. 3. In general, it consists of a parser step and a generator step. Purpose of the first step (parser) is to build an optimized in-memory model of an given IM. A specific parser gets the XML root element and parses an abstract or concrete part. The second step (generator) consists of different generators reading the in-memory model to generate specific artefacts (e.g., classes, rule files, etc.). This distinction is necessary for the desired flexibility of the QoS Platform itself and follows the single responsibility principle. Further explanations of the in-memory model, the QoS-Generator parsers and generators are given in this section. At the end a brief overview of the execution of these components is described.

a) In-memory model: The optimized model is part of the `Context` class. Furthermore, it contains general configuration information, a `TypeMapperRepository` and a `GeneratorModel`. The TypeMapperRepository contains mappers to translate XML Schema types into implementation specific types (e.g., SQL, Java, etc.). The GeneratorModel contains specific information (package definitions, etc.) for the generators and is shared between them by the Context class.

An overview of the in-memory model is given in Fig. 6. The model contains class representations of all IM concepts like Services or Events. A major part is the representation of Measure concepts which is combined in the MeasurableConcept class. It contains references to BaseMeasure and DerivedMeasure classes. Also it contains helper methods to access these (e.g., based on the name) or `Measurement Methods` and `Measurement Functions` directly. Each IM concept class contains all attributes and elements as shown in the XML. Instead of accessing the XML directly, this approach offers helper methods and direct references to other parts of the IM. Where possible attributes are not represented in simple String values but instead by more specialized types like enums or else. Further optimizations can be implemented upon this basis.

b) QoS-IM parsers: A brief overview of all parsers is given in Fig. 4. As stated before each IM concept has a corresponding parser class. The Context and XML root element is given to each parser. The QoS Generator defines the order in which these parsers have to be executed. Basis for all parsers is the minimal Parser interface which defines only the `parser(...)` method. Thus, all parsers can be implemented completely independent, which allows to implement more complex optimizations or more in-depth evaluations of an IM.

An overview of the `DerivedMeasureParser` class and its basic dependencies is given in Fig. 5. While the Context contains the Model (and thus all previously parsed elements), the XML Element class usually contains the Information Need concept of the given QoS IM. Parsers are divided into classes for abstract and concrete parts. As shown before the abstract part is parsed first and thus this parser calls their concrete part parsers. In this case the concrete parts are parsed by the `DerivedMeasureParser` and `MeasurementFunctionParser`. The latter one handles all needed subsequent parsers calls. A Derived Measure can

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

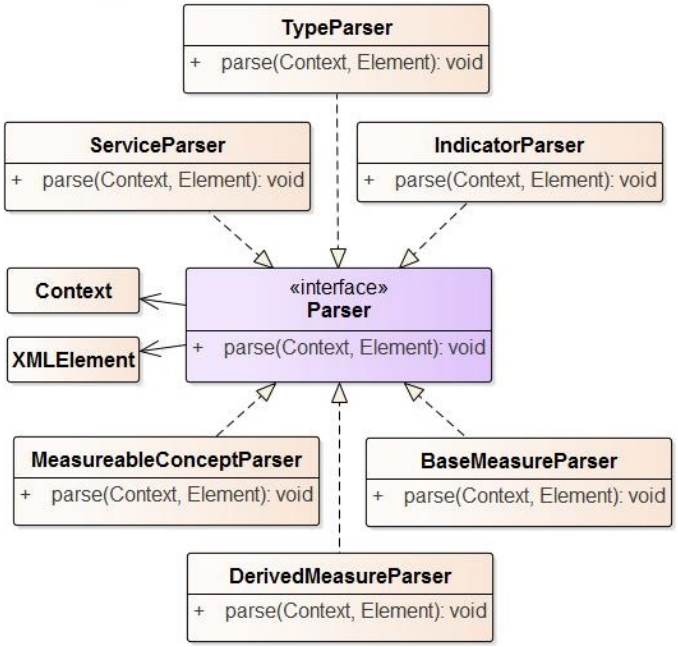


Figure 4. Overview of the parser classes.

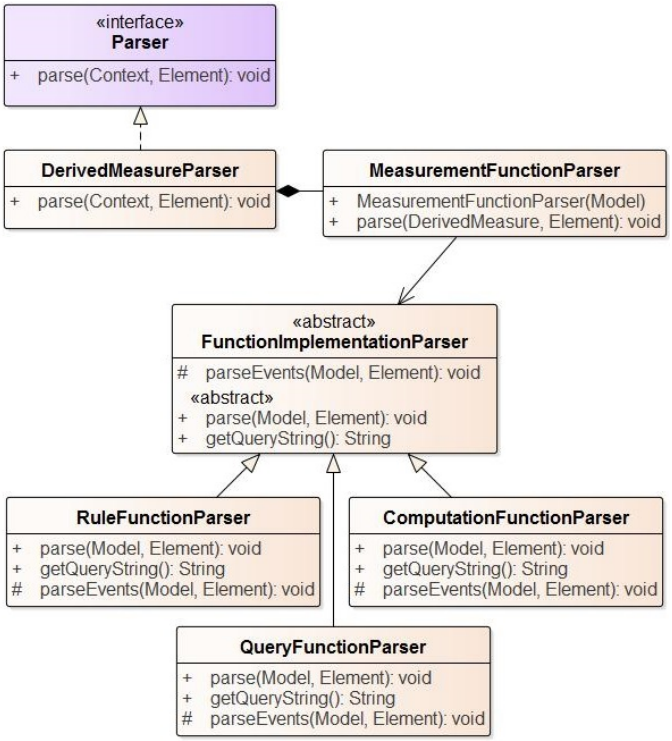


Figure 5. Overview of the Derived Measure parser class.

contain a concrete part for rule-, query- or computation-functions. To determine these each parser defines a unique XML query string based on XPath.

c) QoS-Platform generators: While the parsers are tailored towards the IM model, the generators are tailored towards implementation artefacts or QoS Platform concepts. There are generators for the QoS Agent, Indicator implementation or complex event processing (CEP) rules. An overview of the different generators is given in Fig. 7. Each generator has a specific task concluding in the generation of certain artefacts. This further supports the flexibility of the QoS Platform itself. All generators implement the `Generator` interface and thus get the Context and GeneratorModel objects. Purpose of the latter is to share certain information between different generator in a defined and consistent way. In this case only three Java package definitions are shared. The Velocity template engine (cf. [31]) was chosen because of its ease of use and simplicity. It offers access to Java objects through a template language which can be used to generate HTML or Java code. To further simplify the generator implementation several abstract base classes were developed. While the `VelocityShellGenerator` only offers basic initializations, the `VelocityGenerator` prepares nearly everything to generate artefacts. Only the file name and the specific arguments for the template have to be provided. This differentiation was needed because the QueryGenerator creates several different artefact types (e.g., SQL files, Java classes) and thus uses several different templates. The other generators only have to use one template in the moment.

A detailed view of the `QueryGenerator` is given in Fig. 8. Different SQL files and Java query classes are generated by this generator. To initialize several Velocity Template classes, direct access to the VelocityEngine is needed. Several `TypeMapper` classes are retrieved from the TypeMapperRepository. Actual generation is delegated to several `generate...` methods. Each one builds the corresponding `VelocityContext` and calls the `writeToFile(...)` method to render the template. The VelocityContext is basically a key value map which can be accessed inside of a template.

d) QoS-Generator execution: An example of a QoS Generator parser execution is given in Fig. 9. After the IM is parsed by a SAX compatible parser, the XMLElements are given to each one separately. Every IM concept is parsed from the abstract part to the concrete part. In this case, the `Uses`, `UnitOfMeasurement` and `TargetValue` elements are parsed by the `DerivedMeasureParser`. After the `MeasurementFunctionParser` parses the relevant attributes (e.g., the Name attribute) and creates the corresponding Model class, it determines which type of implementation it contains. This is done by a query string that each `FunctionImplementationParser` offers. In this specific case, the `MeasurementFunction` contains a query and thus the `QueryFunctionParser` is executed. Each query implementation uses certain events to execute a query. To parse these, the inherited method `parseEvents` is given the Event elements.

A brief overview of the a query generator execution is given in Fig. 10. The initialization step of each generator consists of initializing the needed Velocity Templates. Each template represents a certain type of file that

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

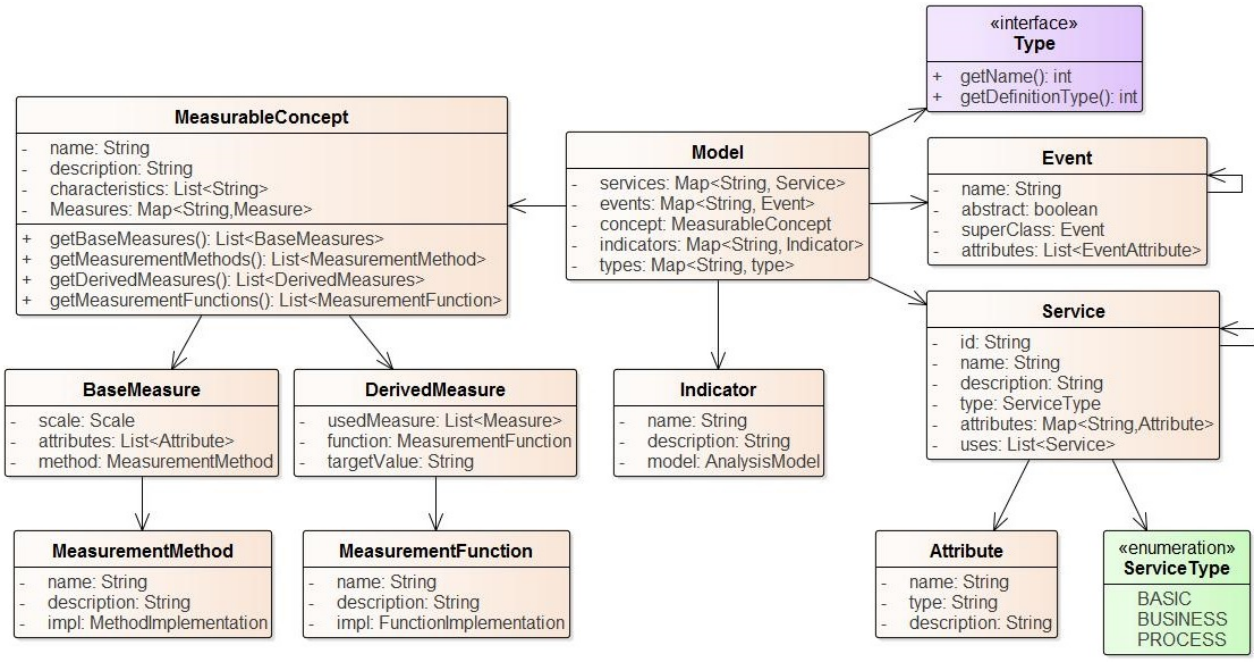


Figure 6. Overview of the in-memory model.

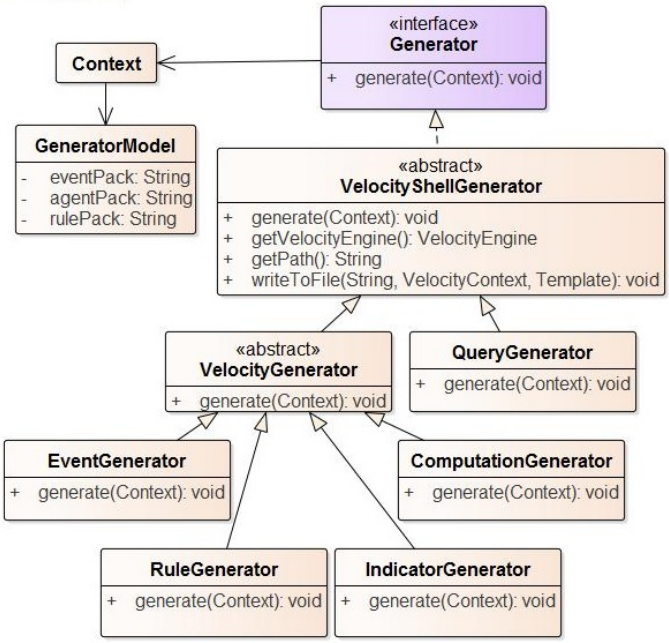


Figure 7. Overview of different generators.

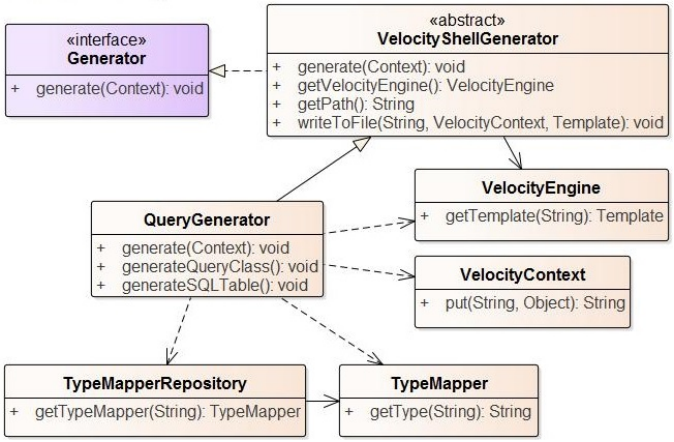


Figure 8. QueryGenerator and certain dependencies.

can be generated. The `QueryGenerator` is derived from `VelocityShellGenerator` and thus the initialization step is very complex to offer more flexibility. A query consists of Java and SQL files, thus several templates are needed. The generation is started through the `generate(Context)` method call. The first step is to determine the needed TypeMappers. As stated before mappers for Java and SQL types are needed, which can be loaded by their names (e.g., 'java', 'sql'). The second step is to generate the files and artefacts by building the needed `VelocityContext` classes and calling the derived method `writeToFile`. Each `VelocityContext` object contains all data (e.g., variable names, data types, etc.) in a simple Map like data structure. Inside each template certain special codes can be used to access Context data, execute loops or use conditional statements to determine what will be rendered.

V. Implemented Concepts and their Artefacts

In the following paragraphs, different concrete parts and their corresponding artefacts are presented. Note, only excerpts are shown and currently not all elements of the abstract part are used. In Fig. 11 an overview of the QoS System and it's different components is given. The Measurement Agents (or QoS Agent) are only design-wise part of the QoS-System. In general they are placed inside the measured system (e.g., an

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

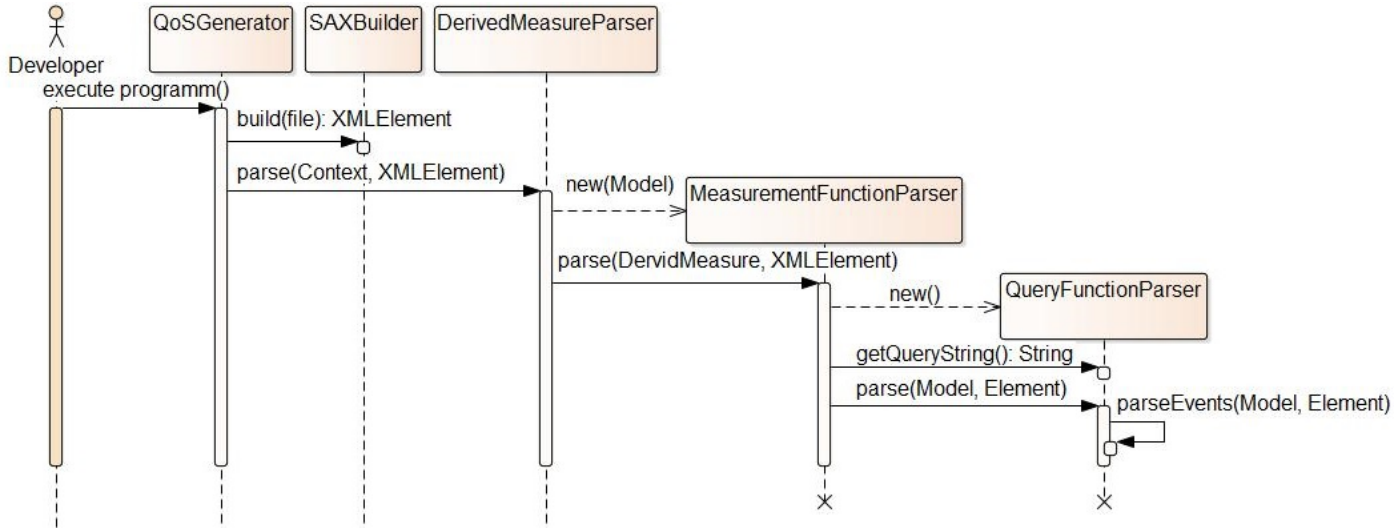


Figure 9. Overview of a parser execution.

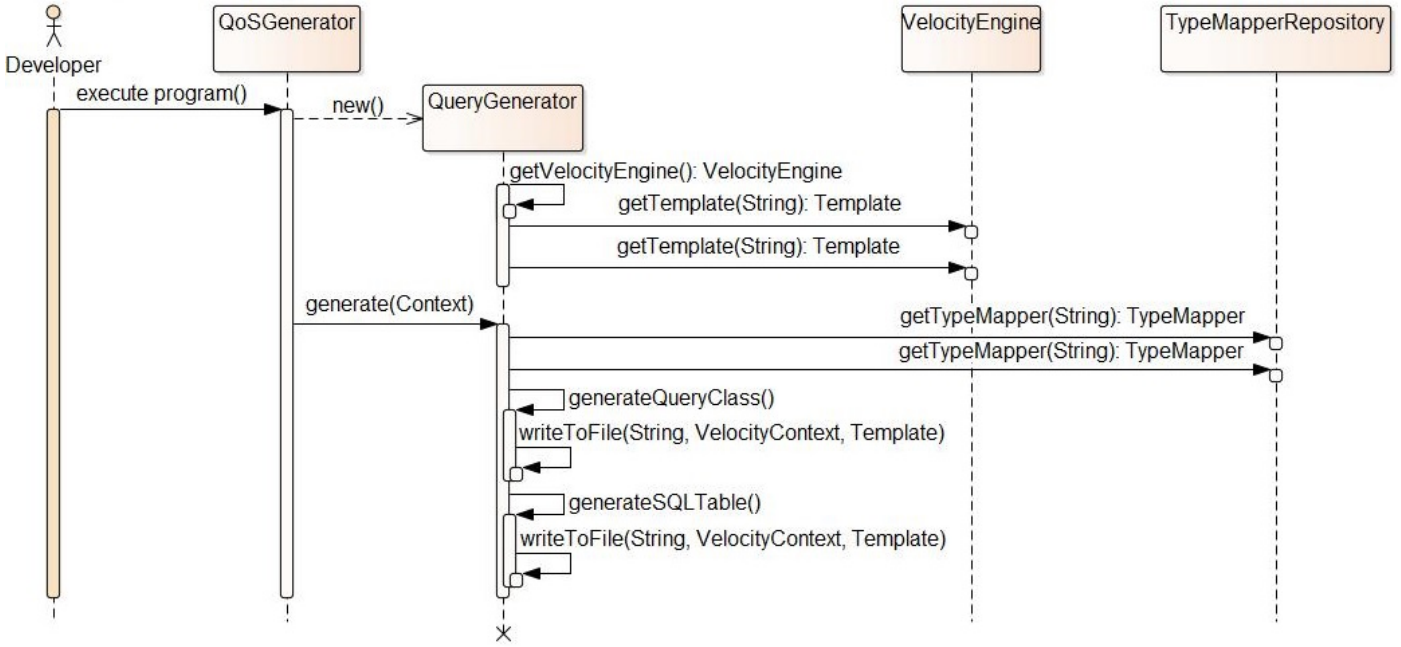


Figure 10. Overview of a generator execution.

ESB, process engine, etc.) and send its results to the QoS Platform. In our current case, this is done via REST but other technologies (e.g., Sockets, CORBA, SOAP, JMS) can be implemented too. The QoS Platform consists of two distinct components.

QoS Measurement is the 'first stage' and contains logic to format or filter incoming events. Furthermore, it contains a CEP engine (JBoss Drools) to further compute and analyze the event stream. QoS Analyzer module is the 'second stage' and contains the main computation of Derived Measures and Indicators. Interfaces to offer these data to downstream systems (e.g., alerting) are also implemented in this module. While the QoS Measurement handles incoming events as fast as possible, the Analyzer module is heavily based on SQL queries and computations which are only executed when needed.

a) Event Model: The `EventModel` is a concrete part inside an IM but part of a specific IM Measure. It is defined aside of these concepts and defines the different events used inside QoS Platform, especially by the QoS Agent and the Measure components. It offers concepts to define events, their `Attributes` and dependencies (inheritance) between them. Currently used is the model shown in Listing 5.

From this model several Java POJOs are generated, which are shown in Fig. 12. An overview of the generated code is shown in Listing 6 and Listing 7. Each Attribute is generated with their mapped type, their name and the needed getters and setters. Furthermore, `abstract` keyword is generated if set

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

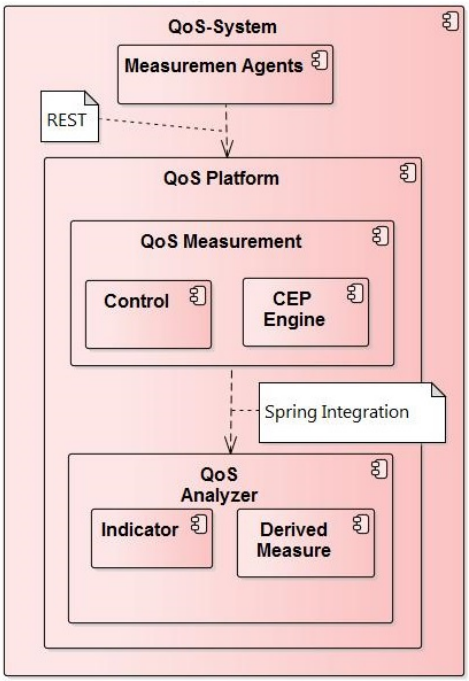


Figure 11. Overview of the QoS System.

```
<EventModel>
 <EventType Name="ServiceBaseEvent" Abstract=
    "true">
  <Attribute Name="id"
     Type="xs:positiveInteger"/>
  <Attribute Name="t_term"
     Type="xs:positiveInteger"/>
 </EventType>
 <EventType Name="ServiceStartEvent"
     Super="ServiceBaseEvent"/>
 <EventType Name="ServiceEndEvent"
     Super="ServiceBaseEvent"/>
 <EventType Name="ServiceDurationEvent">
  <Attribute Name="id"
      Type="xs:positiveInteger"/>
  <Attribute Name="t_inst" Type="xs:long"/>
  <Attribute Name="t_proc" Type="xs:long"/>
 </EventType>
</EventModel>
```

Listing 5. EventModel concrete part of an IM.

to true. If a super class is set, the Java extends clauses are generated too. The `ServiceBaseEvent` is mainly used by the QoS Agent and Measurement component. The ServiceDurationEvent is a complex event created through the CEP rule. While the id attributes are simply integers as defined by the insurance system, the other attributes are of data type long. This is necessary to hold timestamps with the needed precision.

b) Measurement Agent: The concrete part of the Base Measure `t_inst` is given in Listing 2. It defines `Attribute` elements and references the computed QoS Event. The ServiceCallID is parsed from Service Call data. The Time attribute

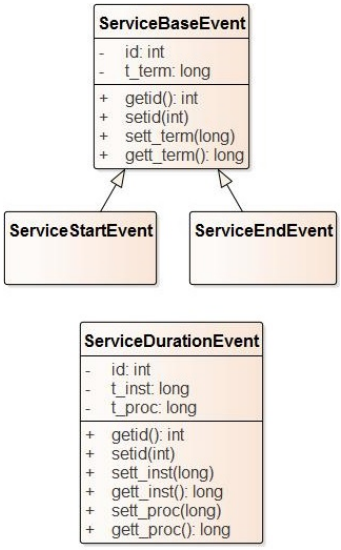


Figure 12. Detailed view of the generated Event classes.

```
public abstract class ServiceBaseEvent  {
 private int id;
 public int getid() {
  return id;
 }
 public void setid(int id) {
  this.id = id;
 }

 private long t_term;
 public long getT_term() {
  return t_term;
 }
 public void setT_term(long t_term) {
  this.t_term = t_term;
 }
}
```

Listing 6. Generated Java POJO ServiceBaseEvent.

will be computed by the Agent itself. Furthermore, a class attribute is given in the Agent element. It is used to structure the generated code and the corresponding artefacts. The specific method name is derived from the `MeasurementMethod` element. While the QoS Agent is designed as part of the QoS System, it is actually placed directly into the SOA as part of the ESB component. The used ESB is a partner specific implementation.

c) Measurement Module: A brief overview of the Measurement module is given in Fig. 13. The QoS Agent send several events to this module. Each event is currently handled and formatted by `EventController` and `EventFormatter` classes. The formatters are needed because Agents only sent raw string-based data. After the formatter constructs actual Event classes, these are given to the Drools CEP Engine. This is done through the `DroolsEndpoint` class. These classes are considered to be part of the QoS Platform core. Thus, they are not generated and only adjusted if needed. EventFormatter are written for each Event that a BaseMeasure uses. In the `postConstruct` method, the CEP engine is

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
public class ServiceDurationEvent  {
 private int id;
 public int getid() {
  return id;
 }
 public void setid(int id) {
  this.id = id;
 }

 private long t_inst;
 public long gett_inst() {
  return t_inst;
 }
 public void sett_inst(long t_inst) {
  this.t_inst = t_inst;
 }

 private long t_proc;
 public long gett_proc() {
  return t_proc;
 }
 public void sett_proc(long t_proc) {
  this.t_proc = t_proc;
 }
}
```

Listing 7. Generated Java POJO ServiceDurationEvent.

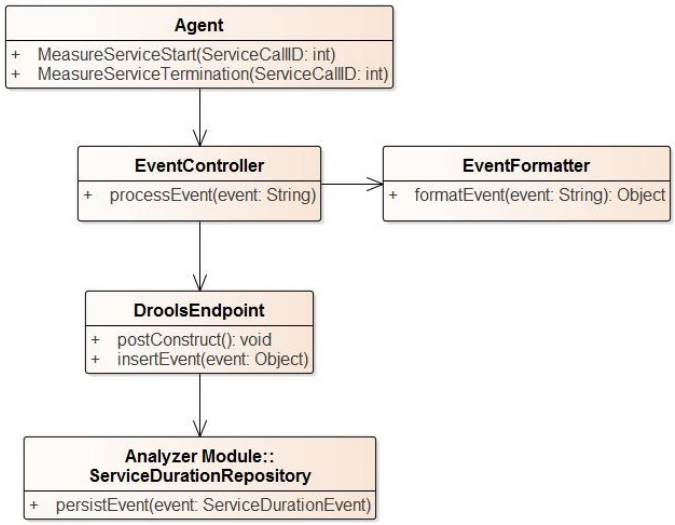


Figure 13. Overview of the measurement module.

further initialized with logging capabilities and a Spring Integration Channel. The `ServiceDurationRepository` of the Analyzer module is listening on this channel and persists every incoming complex event.

The concrete part of the Derived Measure `t_proc` is given in Listing 8. It contains the definition of the CEP rule, which computes the complex event ServiceDurationEvent. Plain element indicates that this code fragment will be placed 'as is' into a rule file. Only certain definitions (e.g., for event classes) will be added. The rule file is loaded on start up by the CEP engine (JBoss Drools) of the QoS Measurement module. The generated rule file is shown in Listing 9. Besides the import definitions, also the package definition is added. This

```
<Rule>
 <Event Name="ServiceDurationEvent"
  Handle="output"/>
 <Plain>
  rule "Service Duration Rule"
  when
   $start : ServiceStartEvent()
   $end : ServiceEndEvent(
    this after[ 0s , 2s ] $start &&
    this.id == $start.id
   )
  then
   channels["analyzer"].send(
    new ServiceDurationEvent(...)
   );
  end
 </Plain>
</Rule>
```

Listing 8. Concrete Part of a Rule.

```
package de.hshannover.ccitm.qos.measurement;

import de.hshannover.ccitm.qos.events.*;

declare ServiceEndEvent
 @role( event )
end
declare ServiceDurationEvent
 @role( event )
end
declare ServiceStartEvent
 @role( event )
end

rule "Service Duration Rule"
when
 $start : ServiceStartEvent()
 $end : ServiceEndEvent( this after[ 0s , 2s ]
     ] $start && this.id == $start.id )
then
 logger.info("Duration rule fired");
 ServiceDurationEvent duration = new
     ServiceDurationEvent();
 duration.setid($start.getid());
 duration.sett_inst($start.getT_term());
 duration.sett_proc($end.getT_term() - $start
     .getT_term());
 channels["analyzer"].send(duration);
end
```

Listing 9. Generated rule file.

definition is based upon the information of the GeneratorModel described earlier. Another important and generated part are the `declare` statements. These define the Event POJOs as CEP events in the JBoss Drools sense. As shown, only the actual needed events of the included rule have declare statements.

d) Analyzer Module: The generated class and rule files for the DerivedMeasures are part of the QoS Platform (and part of the QoS Platform.war). For example, the generated classes

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

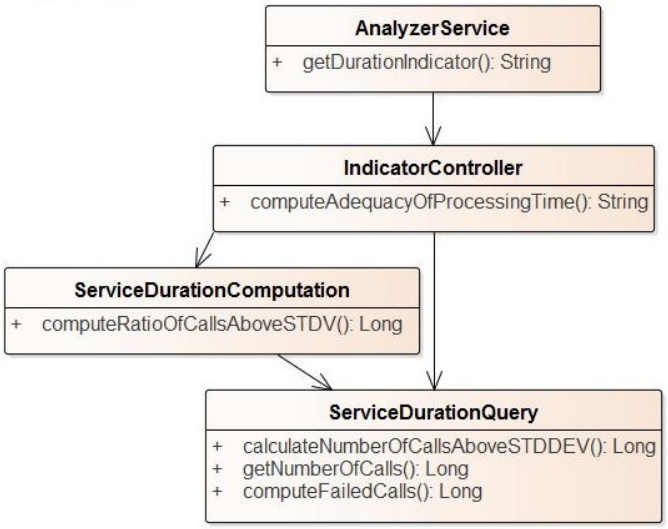


Figure 14. Detailed view into the Analyzer Module.

of the Analyzer module are shown in Fig. 14. Indicator and *-Duration classes are integrated, if needed manually, into the QoS Platform. The AnalyzerService class is considered part of the QoS Platform core and implements the REST interface for downstream systems (e.g., alerting).

The concrete part of the Derived Measure `COUNT_STDEV_CALLS` is given in Listing 3. It contains the SQL query to get the count of all events with a runtime above the doubled standard deviation. The generated class is shown in Listing 10. The `QUERY_STDDEV_EVENTQuery` attribute contains the SQL query given in a Plain element. Again, the class attribute is used to structure the code and artefacts, but the Type attribute is specific for a query and specifies the return type (in this case Long) of the query. The name of method is given in the `MeasurementFunction` element. Also, this class contains the SQL queries for other derived measures. Furthermore, needed imports and Spring code to integrate the jdbcOperations object are generated.

The concrete part of the Indicator `SLoT_proc` is given in Listing 4. Each `IndicatorEntry` element consists of an `Input` where the Indicator condition is defined and a `Result` element, which contains the actual Indicator response. Each of these results has to be a valid HMN type. The generated IndicatorController class is given in Listing 11. The dependencies to other Measure results are given by the `Uses` element. This information is also used for generation and manual modifications. In this case, the ratio of service calls above the standard deviation is computed by the `ServiceDurationComputation` class. The `ServiceDurationQuery` class provides the number of failed service calls.

VI. Measurements

For the evaluation of the described measurement concept, it is stressed with an initial load test. The general 'Information Need' (Fig. 1) is the information about how volatile a software

```
public class ServiceDurationQuery {
 private String QUERY_STDDEV_EVENTQuery =
  "SELECT COUNT(*) FROM serviceduration
 WHERE
  TINTS > TIME_SECS(DATEADD('DAY',-30,NOW()))
 AND TPROC > (SELECT AVG(TPROC)
  + (2 * STDDEV(TPROC))
 FROM serviceduration
 WHERE TINTS >
  TIME_SECS(DATEADD('DAY',-30, NOW())))";
 ...

 private JdbcOperations jdbcOperations;

 public ServiceDurationQuery(DataSource
     dataSource) {
  jdbcOperations =
    new JdbcTemplate(dataSource);
 }

 public Long
     calculateNumberOfCallsAboveSTDDEV() {
  return jdbcOperations.queryForObject(
    QUERY_STDDEV_EVENTQuery, Long.class );
 }

 public Long getNumberOfCals() {
  return jdbcOperations.queryForObject(
   QUERY_EVENT_COUNTQuery, Long.class );
 }

 public Long computeFailedCalls() {
  return jdbcOperations.queryForObject(
   QUERY_FAILED_EVENTQuery, Long.class );
 }
}
```

Listing 10. Generated query class.

system is currently being stressed. Static thresholds cannot fulfill the desired 'Information Need' of the partner companies in the insurance industry. The dynamic approach of measuring the spikes, which exceed the standard-deviation of a measuring period, can provide better answers here. For the evaluation, such spikes are directly provoked. When generating the spikes, two parameters are randomly influenced:

- Intensity: The intensity of the spikes.
- Frequency: The frequency at which the spikes occur.

In the stress test, the two parameters 'Intensity' and 'Frequency' are set. A high intensity means that a spike is generated with a high level of volatility. The intensity describes, how long the response time of a service request is and how 'intensive' the standard deviation is exceeded according to (1). The frequency determines, how often such a spike should occur in the stress test. The stress test therefore generates very volatile measurement events, which must be recorded dynamically by the measuring system. So, a random variation of these two parameters will provoke volatile stress situations with unpredictable intensity and frequency. This allows the measuring system to be tested as strongly and dynamically as possible. Some of the preliminary results measured with the QoS System are shown in Fig. 15. The yellow line shows the standard deviation barrier. In this case, 3 % of all measured

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
public class IndicatorController {
 @Autowired
 ServiceDurationQuery durationQuery;
 @Autowired
 ServiceDurationComputation
     durationComputation;

 public String
     computeAdequacyOfProcessingTime() {
  long devPercentageCount =
      durationComputation.
      computeRatioOfCallsAboveSTDV();
  long badCount = durationQuery.
      computeFailedCalls();

  if(devPercentageCount < 5 &&
   badCount == 0) {
    return "low";
  }
  if(devPercentageCount >= 5 &&
   badCount == 0) {
    return "middle";
  } else {
   return "high";
  }
}}
```

Listing 11. Generated indicator class.

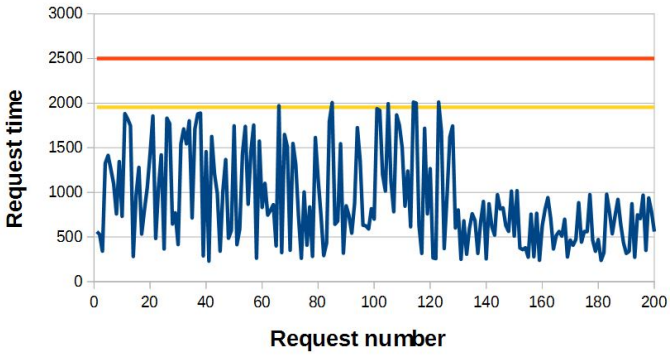


Figure 15. Preliminary Measurement Results.

requests (6 service calls) are violating the barrier, thus the computed indicator would be low. Above 5 %, the indicator would be middle. The red line shows the SLA barrier introduced by service consumers like check24.de. If one request exceeds this barrier, the indicator switches to high. A more thorough test and evaluation based on these loads will be given in our future work. But based on these results, the measurement concept can be used to even measure very volatile stress situations.

VII. Conclusion and Future Work

In this article, we presented an approach for monitoring a distributed SOA environment, which we see as a promising path to take. Our SOA Quality Model is aimed to follow the ISO/IEC-Standard 15939 (cf. [32]), which enables a wide range of use cases. Our Measurement Concept outlines an execution platform for the specific QoS Information Model, which should cause minimal impact on the SOA environment.

The separation of Measurement Agents and QoS-Analyzer on one hand allows lightweight agents and on the other hand a very capable analyzer component. Furthermore, certain parts of our QoS Platform can be replaced or complemented with common tools, e.g., from the microservices eco system. For example, Netflix's Hystrix could be used to implement a BaseMeasure or Prometheus to implement DerivedMeasures. This flexibility in our architecture with the general concept given by our SOA Quality Model offers new opportunities for our partner companies.

Already in previous work [2] [3], we presented our general measurement concept, an initial business process (the 'check 24' Proposal Service insurance use case, a basic business relevant scenario), and our information model and concept. The core contributions of the present article are implementation details of our approach.

Therefore, in Section IV, we dive deeply into design and operation of the QoS Generator. In Section V, implementation details of the generated artifacts – e.g., event model, agent, measurement module, and analyzer module – are described in depth.

Our ongoing work of applying the QoS System to an application scenario relevant to our partner in the insurance industry (the so called 'Check 24 process'), will provide evidence of the practical usability of the created framework. Furthermore, a more thorough evaluation will be the main part of our future work.

To this end, we designed and implemented a simulation environment based on the QoS System and applied to the partner's system architecture. The simulation environment will be feed with real data, i.e., the number of requests per unit of time over the day, to perform measurement and analysis.

It is expected that our monitoring system will help to discover potential bottlenecks in the current system design of our partner's distributed services. Therefore, it will create value in the process of solving these issues.

In future work, we have planned to apply our existing work to the more complex insurance process 'Angebot erstellen' ('create individual proposal') of the VAA [28]. Thus, we will implement a more complex insurance scenario. Moreover, the actual measurement and analysis of the results are an ongoing process, which is yet to be finalized.

We also have plans to apply these results onto cloud-based environments. Furthermore, a deeper subdivision or extraction, from the current coarse granular SOA services into fine-grained microservices, will be investigated by us in future work 'where it makes sense', for e.g., to allow for a better scalability of individual microservices.

References

[1] A. Hausotter, A. Koschel, J. Busch, and M. Zuch, "A Generic Measurement Model for Service-based Systems," in: The 10th International Conferences on Advanced Service Computing (Service Computation), IARIA, Barcelona, Spain, 2018, pp. 12-18.

[2] A. Hausotter, A. Koschel, J. Busch, M. Petzsch, and M. Zuch, "Implementing a Framework for QoS Measurement in SOA – A Uniform Approach Based on a QoS Meta Model," in: IARIA Intl. Journal On Advances in Software, 10(3,4), 2017, pp. 251-262.

[3] A. Hausotter, A. Koschel, J. Busch, M. Petzsch, and M. Zuch, "Agent based Framework for QoS Measurement Applied in SOA," in: The 9th International Conferences on Advanced Service Computing (Service Computation), IARIA, Athens, Greece, 2017, pp. 16-23.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[4] T. Bergemann, A. Hausotter, and A. Koschel, "Keeping Workflow-Enabled Enterprises Flexible: WfMS Abstraction and Advanced Task Management," in: 4th Intl. Conference on Grid and Pervasive Computing Conference (GPC), 2009, pp. 19-26.

[5] C. Gäth, A. Hödicke, S. Marth, J. Siedentopf, A. Hausotter, and A. Koschel, "Always Stay Agile! – Towards Service-oriented Integration of Business Process and Business Rules Management," in: The Sixth International Conferences on Advanced Service Computing (Service Computation), IARIA, Venice, Italy, 2014, pp. 40-43.

[6] A. Hausotter, C. Kleiner, A. Koschel, D. Zhang, and H. Gehrken, "Always Stay Flexible! WfMS-independent Business Process Controlling in SOA," in: IEEE EDOCW 2011: Workshops Proc. of the 15th IEEE Intl. Enterprise Distributed Object Computing Conference, IEEE: Helsinki, Finnland, 2011, pp. 184-193.

[7] A. Koschel and R. Kramer, "Configurable Event Triggered Services for CORBA-based Systems," Proc. 2nd Intl. Enterprise Distributed Object Computing Workshop (EDOC'98), San Diego, U.S.A, 1998, pp. 1-13.

[8] M. Schaaf, I. Astrova, A. Koschel, and S. Gatziu, "The OM4SPACE Activity Service - A semantically well-defined cloud-based event notification middleware," in: IARIA Intl. Journal On Advances in Software, 7(3,4), 2014, pp. 697-709.

[9] B. Schroeder, "On-Line Monitoring: A Tutorial," IEEE Computer, 28(6), pp. 72-80, 1995.

[10] S. Schwiderski, "Monitoring the Behavior of Distributed Systems," PhD thesis, Selwyn College, University of Cambridge, University of Cambridge, Computer Lab, Cambridge, United Kingdom, 1996.

[11] Dynatrace LLC, "Dynatrace Application Monitoring," [Online]. URL: https://www.dynatrace.com/de/products/application-monitoring.html [accessed: 2017-12-06].

[12] Nagios.ORG, "Nagios Core Editions," [Online]. URL: https://www.nagios.org/ [accessed: 2016-12-26].

[13] N. W. Paton (ed.), "Active Rules for Databases," Springer, New York, 1999.

[14] ACT-NET Consortium, "The Active DBMS Manifesto," ACM SIGMOD Record, 25(3), 1996.

[15] M. Garcia-Valls, P. Basanta-Val, M. Marcos, and E. Estévez, "A bi-dimensional QoS model for SOA and real-time middleware," in: Intl. Journal of Computer Systems Science and Engineering, CLR Publishing, 2013, pp. 315-326.

[16] V. Krishnamurthy and C. Babu, "Pattern Based Adaptation for Service Oriented Applications," in: ACM SIGSOFT Softw. Eng. Notes 37, 2012(1), 2012, pp. 1-6.

[17] T. Frotscher, G. Starke (ed.), and S. Tilkov (ed.), "Der Webservices-Architekturstack," in: SOA-Expertenwissen, Heidelberg, dpunkt.verlag, 2007, pp. 489-506.

[18] F. Curbera, R. Khalaf, and N. Mukhi, "Quality of Service in SOA Environments. An Overview and Research Agenda," in: it - Information Technology 50, 2008(2), 2008, pp. 99-107.

[19] S.W. Choi, J.S. Her, and S.D. Kim, "QoS Metrics for Evaluating Services from the Perspective of Service Providers," in: Proc. of the IEEE International Conference on e-Business Engineering, Washington DC, USA : IEEE Computer Society (ICEBE'07), 2007, pp. 622-625.

[20] Z. Balfagih and M.F. Hassan, "Quality Model for Web Services from Multi-stakeholders' Perspective," in: Proceedings of the 2009 International Conference on Information Management and Engineering, Washington DC, USA : IEEE Computer Society (ICIME'09), 2009, pp. 287-291.

[21] G. Wang, A. Chen, C. Wang, C. Fung, and S. Uczekaj, "Integrated Quality of Service (QoS) Management in Service-Oriented Enterprise Architectures," in: Proceedings of the 8th IEEE Intl. Enterprise Distributed Object Computing Conference (EDOC'04), Washington DC, USA, IEEE, 2004, pp. 21-32.

[22] M. Varela, L. Skorin-Kapov, F. Guyard, and M. Fiedler, "Meta-Modeling QoE," PIK-Praxis der Informationsverarbeitung und Kommunikation, 2014, Vol. 37(4), pp. 265-274.

[23] R.W. Maule and W.C. Lewis, "Performance and QoS in Service-Based Systems," Proc. of the 2011 IEEE World Congress on Services, IEEE Computer Society, 2011, pp. 556-563.

[24] B. Wetzstein et al., "Monitoring and Analyzing Influential Factors of Business Process Performance," in: Proc. IEEE Intl. Enterprise Distributed Object Computing Conf. (EDOC'09), 2009, pp. 141-150.

[25] F. Rosenberg, C. Platzer, and S. Dustdar, "Bootstrapping Performance and Dependability Attributes of Web Services," in: Proc. International Conference on Web Services (ICWS'06), 2006, pp. 205-212.

[26] M. Schmid, J. Schaefer, and R. Kroeger, "Ein MDSD-Ansatz zum QoS-Monitoring von Diensten in Serviceorientierten Architekturen," in: PIK Praxis der Informationsverarbeitung und Kommunikation, 31 (2008) 4, 2008, pp. 232-238.

[27] S.M.S. da Cruz, R.M. Costa, M. Manhaes, and J. Zavaleta, "Monitoring SOA-based Applications with Business Provenance," Proc. of the 28th Annual ACM Symposium on Applied Computing (ACM SAC), ACM, 2013, pp. 1927-1932.

[28] GDV (Gesamtverband der Deutschen Versicherungswirtschaft e.V. – General Association o.t. German Insurance Industry), "Die Anwendungsarchitektur der Versicherungswirtschaft: Das Objektorientierte Fachliche Referenzmodell (The application architecture of the German insurance business – The functional object-oriented reference model," VAA Final Edt. Vers. 2.0, 2001, [Online]. URL: http://www.gdv-online.de/vaa/vaafe_html/dokument/ofrm.pdf [accessed: 2017-01-11].

[29] ISO - International Organization for Standardization (ed.), "ISO/IEC 25010:2011 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models," 2011.

[30] M. Azuma, "SQuaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality, " in: Proc. of European Software Control and Metrics (ESCOM), 2001, pp. 337-346.

[31] The Apache Software Foundation, "The Apache Velocity Project," [Online]. URL: https://velocity.apache.org// [accessed: 2018-08-15].

[32] ISO - International Organization for Standardization (ed.), "ISO/IEC 15939:2007 - Systems and software engineering - Measurement process," 2007.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

An Extended Evaluation of Process Log Analysis for BPEL Test Coverage Calculation

Daniel Lübke

Leibniz Universität Hannover
FG Software Engineering
Welfengarten 1, D-30167 Hannover, Germany
Email: daniel.luebke@inf.uni-hannover.de

***Abstract*—With today's requirement of quickly developing digitization solutions, companies often use specialized workflow languages like the BPEL or BPMN 2.0, which orchestrate services along the process flow. Because process models are of critical importance to the functioning of the organization, high quality and reliability of the implementations are mandatory. Therefore, testing becomes an even more important activity in the development process. For judging the quality of developed tests, Test Coverage Metrics can be used. Current approaches to test coverage calculation for BPEL either rely on instrumentation, which is slow, or are limited to vendor-provided unit test frameworks, in which all dependent services are mocked, which limits the applicability of such approaches. Our refined approach relies on analyzing process event logs that are written during process execution. Within this article we analyze the performance characteristics of process log analysis versus the instrumentation-based approach by running an experiment with BPEL processes and their accompanying test suites developed in an industry project. According to our findings, the improved version of process log analysis is significantly faster for all scenarios.**

***Keywords*–Test Coverage; Process Mining; BPEL; Event Log; Experiment; Performance.**

I. Introduction

This article presents improvements, an extended description as well as an improved experimental evaluation of using process log analysis as a method for measuring test coverage of BPEL processes presented earlier [1].

Having a flexible and fast tool for measuring test coverage is important due to the rising importance of digital business process solutions. Partner networks are being connected tighter and the integration between different businesses is often driven by business process needs. For example, offering fully fledged digital services to customers requires a high degree of automation, i.e., the implementation of large parts of business processes in software solutions. Because the failure of customer- or partner-facing processes can have dramatic financial and reputational consequences, the required quality, stability and correctness of process-based software solutions is an important problem in practice – and as such a relevant research topic.

Business processes can be digitized by using special workflows, which are refered to as executable business processes. Standards like BPEL or BPMN 2.0 have been developed to automate business processes in large companies by orchestrating services. These are software artefacts and can contain complex orchestration logic. With the increasing demand for fully digitized solutions, it is likely that more and more business processes are being implemented in these or similar orchestration languages.

Because business processes – and as such their software implementations – are very critical to the functioning and performance of organizations, it is mandatory to perform good quality assurance in order to avoid costly problems in production [2]. Quality Assurance can include static checking of process models (e.g., consistency check of service contracts to executable processes [3]). However, most projects use testing as their main quality assurance activity. Consequently, they require an assessment of the adequacy and quality of the tests. It has been shown by Piwowarski et al. [4] that a) test coverage measurements are deemed beneficial by testers, although b) they are rarely applied because of being difficult to use, and c) that higher coverage values lead to more defects being found. These findings are supported by Horgan et al. [5], who linked data-flow testing metrics to reliability, and Braind et al. [6], who simulated the impact of higher test coverage on quality. Furthermore, Malaiya et al. [7] and Cai & Lyu [8] have developed prediction models that can link test coverage to test effort and software reliability.

Quality Assurance, and thus test coverage measurement, should be an ongoing activity because executable processes will evolve over time [9]. One way for continously measuring test quality is to measure test coverage as part of all testing activities. Test Coverage then serves as measurement of test data adequacy [10].

While approaches applicable for developing unit tests for executable processes have been proposed by academia (e.g., [11], [12]) and developed by vendors for their respective process engines, there is no practical way to efficiently calculate test coverage for tests that are not controlled by a unit testing framework. Also, approaches relying on instrumentation create significant additional overhead by a factor larger than 2.0 compared with the "plain" test case execution times [13]. This is far more than instrumentation approaches for "normal" programming languages, e.g., Java, require.

An approach that better guides quality assurance in software projects, which develop executable processes, is required. This approach shall be applicable in several test scenarios, including unit tests, integration tests and system tests. Ideally, it is easier to set up than existing methods in order to improve acceptance by practitioners [4].

Within this article, we evaluate an approach based on analyzing process event logs, which are automatically written

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

by process engines during process execution regardless of whether testing frameworks are used or not. The evaluation is done experimentally and compares the execution times of two test coverage measurement approaches: process log analysis and process instrumentation. For running the experiment, four BPEL processes from the commerical Terravis project [14] as well as two research processes from Schnelle [15] are used.

This article is structured as follows: First, the process modeling language BPEL is shortly explained in Section II before related work is presented in Section III. The approach for mining test coverage metrics is described in detail in Section IV followed by a short evaluation of its flexibility in Section V. The main part of this article is presented in Section VI, which describes the experiment set-up, the gathered results and their interpretation comparing the performance of our new approach and the existing instrumentation-based approach. Finally, we conclude and give an outlook on possible future work.

II. Background on BPEL

BPEL (short for WS-BPEL; Web Services Business Process Execution Language) is an OASIS standard that defines a modeling language for developing executable business processes by orchestrating Web services.

BPEL Models consist of *Activities*, which are divided into *Basic Activities* and *Structured Activities*. *Basic Activities* carry out actual work, e.g., performing data transformations or calling a service, while *Structured Activities* are controlling the process-flow, e.g., conditional branching, loops, etc.

Important *Basic Activities* include the *invoke* activity (which calls Web services), the *assign* activity (which performs data transformations), and the receive and reply activities (which offer service others to call a process via service interfaces). Important *Structured Activities* are the *if*, *while*, *repeatUntil*, and *forEach* activities, which offer the same control-flow structures like their pendants in general purpose programming languages, and the *flow* activity, which allows process designers to build a graph-based model including parallel execution. For building the graph, BPEL defines *links* that can also carry conditions for modelling conditional branches.

For handling error conditions and scoped messages, BPEL provides different kinds of *Handlers*: *Fault Handlers* are comparable to try/catch constructs: Whenever a SOAP Fault is returned by an invoked service or is thrown within the process, the Process Engine searches for defined *Fault Handlers*. These may trigger *Compensation Handlers*, which can undo already executed operations. For receiving events asynchronously outside the main process flow, *Event Handlers* can be defined. These come in two flavors: *onEvent Handlers* for receiving SOAP messages, and *onAlarm Handlers* for reacting on (possibly reoccurring) times and time intervals.

BPEL does not define a graphical representation like the BPMN 2.0 standard does, but standardizes the XML format, in which it is saved. Vendors have developed their own graphical representations. Within this article we use the notation of the Eclipse BPEL Designer. A process that will be used as an example in this article is shown in Figure 1: A customer places an order (“receiveInput”). A check is made, whether the customer has VIP status or not. In case of a VIP customer, points are credited to the customer’s account (“SavePointsEarned”). In both cases appropriate response message to the customer are prepared (“PrepareReplyFor...”), which is then sent back to the customer (“reply”). For handling failures while storing the earned points, a *Fault Handler* called “catch” is defined, which prepares (“PrepareTicketCreation”) and creates a help desk ticket (“CreateHelpDeskTicket”) so that the points can be manually added later.

BPEL processes are deployed to a *Process Engine*, which has the responsibility for executing process instances and managing all aspects around process versioning, persistence, etc. The amount of data, which is persisted during process execution, is vendor-dependent and can be configured in most engines during the deployment of a process model.

BPEL has been designed to be extensible. Many extensions by both standard committees and vendors have been made. For example, BPEL4People allows to interact not only with services but also with humans during process execution.

III. Related Work

Testing BPEL processes has become subject of many research projects. For example, Li et al. (BPEL4WS Unit Testing Framework [11]), Mayer & Lübke (BPELUnit [12]), and Dong et al. (Petri Net Approach to BPEL Testing [16]) have developed approaches for testing BPEL processes and published their ideas.

The BPELUnit framework was developed by Mayer & Lübke [12] and was later extended by Lübke et al. [13] with test coverage measurement support. First, the coverage metrics needed to be defined, which is not as straightforward as for other programming languages due to BPEL’s different mechanisms for defining the process-flow. Consequently, three coverage metrics were defined: *Activity Coverage*, *Handler Coverage*, and *Link Coverage*.

Coverage Measurement was done by instrumenting the BPEL process: For tracing the execution, the process is changed prior to deployment. Additional service calls are inserted for every activity. The service calls send the current execution position (“markers”) to the test framework. This enables the test framework to know which activities have been executed in the test run. However, the test framework needs to run while the instrumented processes are executed in order to collect the markers, which makes its use limited in practice. Even more, the overhead introduced by many new service calls is considerable: The reported overhead in the original paper is more than 100%, i.e., the test execution times have more than doubled. This stems from the instrumentation mechanism, which requires every execution trace point to be sent out of the process via a service call, which in turn requires XML serialization and involves the network stack. This also makes the BPEL process much larger: The number of basic activities tripples for instrumenting all measurement points for calculating activity coverage alone. One advantage of the approach is that it only slightly depends on the Process Engine being used: The changes to the BPEL process are completely standards-compliant. Only the new service for collecting markers needs to be added to the engine-specific deployment descriptor. One way to migitate the performance problems is to distribute the tests, e.g., as described by Kapfhammer [17].

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

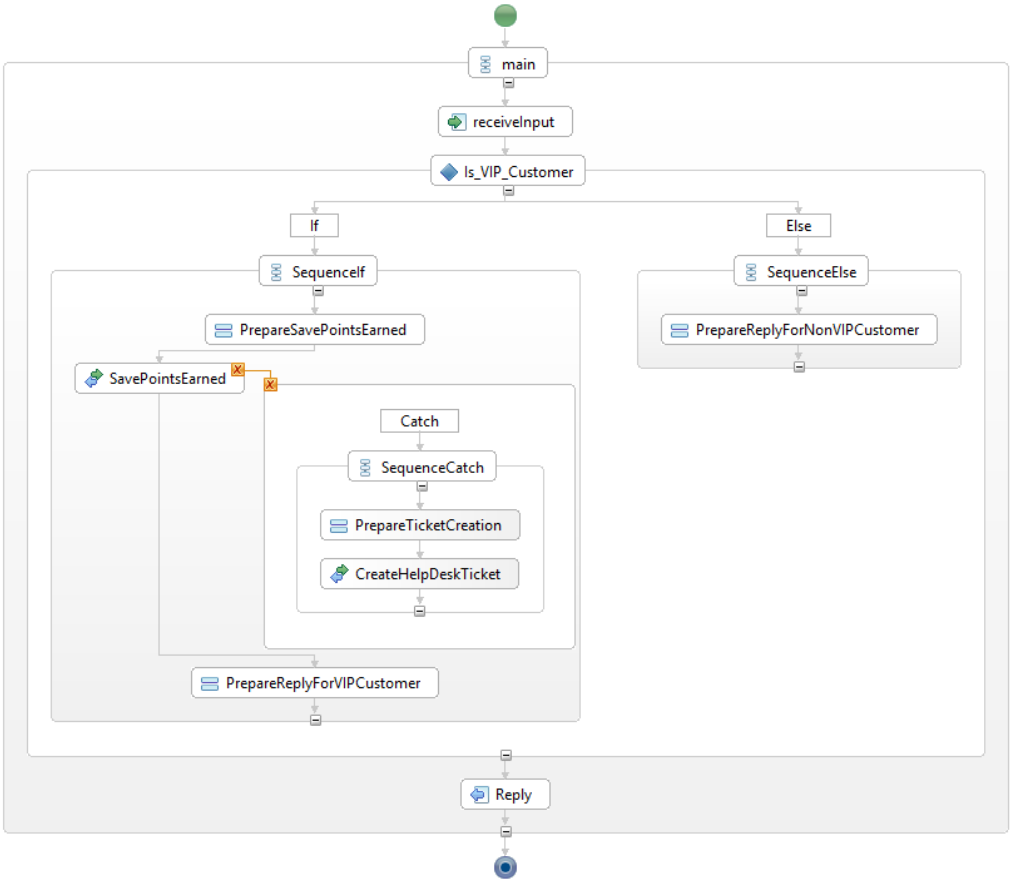


Figure 1. Sample BPEL Process for processing an Order.

Process Engine vendors have also developed their own proprietary solutions for measuring test coverage: Test Cases are developed in the development environment of the process engine and can be executed from there or on a server. All services are mocked and the test frameworks simply inject predefined SOAP messages. Such test frameworks use a striped-down version of the process engine. This results in a mixture between simulation and test: The process engine uses the same logic but not all parts of its code are triggered because some features are disabled. Also, there is no possiblity of calling "real" services instead of mocks. While test coverage calculation is very fast, because the algorithms have access to internal engine data structures, its use is limited to unit test scenarios only. Examples of such vendor-provided test frameworks are Informatica's BUnit [18] and Oracle's BPELTest [19].

Endo et al. [20] defined coverage criteria for their test generation approach. For example, one criterion called All-nodes is that all activities are executed. Other criteria require certain activity types to be executed (e.g., All-nodes for all *invoke* and *reply* activities). The authors use these criteria to guide the test case selection of their generation approach.

On a more general level, test metrics and their publication by services themselves have been researched in the context of Service-Oriented Architectures (SOA) by Miranda et al. [21], Bartolini et al. [22], and Eler et al. [23]. Their approaches are independent of the language used for implementing the services and thus more abstract.

Schnelle & Lübke proposed an approach to generate unit test cases from classification trees, which are designed from a business perspective [24]. Coverage can be specified as the coverage of different properties (=leafs) of that tree. The coverage is not code-based but instead requirements-based.

Other approaches try to generate test cases with good or optimal coverage: Kaschner & Lohmann [25] developed an approach that generates test suites that cover all service interactions. Service Interactions are externally observable behavior and are thus deemed the most important aspect to test by the authors. Ji et al. [26] describe another way: They developed an algorithm that tries to choose the most efficient test cases by analyzing the data-flow of a BPEL process.

Although many coverage metrics have been defined, Weiser et al. [27] have shown that "[e]mpirically comparing structural test coverage metrics reveals that test sets that satisfy one metric are likely to satisfy another metric as well". This means that for practical purposes any of the proposed test coverage metrics will likely behave as well or as badly as another one.

All test approaches available for BPEL claim that they achieve – or at least help to achieve – a good test coverage.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

However, no empirical studies have been made whether this is the case and what typical test coverage values constitute for BPEL processes. In contrast, there is a huge body of knowledge available for general purpose programming languages, as has been shown in a survey for the Java programming language by Yang et al. [28].

In a multiple case study approach Mockus et al. [29] have shown that with increasing test coverage more defects are found prior to production but that costs to improve test coverage increase exponentially while the numbers of found defects only increases linearily. Therefore, Pavlopoulou & Young [30] propose to monitor the production environment for execution of previously untested code sections. By leaving only instrumentation code for these code areas of interest, they could reduce the overhead imposed by coverage measurements.

All in all, there is currently no approach available for BPEL processes that can be used to measure test coverage on code level with acceptable performance and the ability to be used in conjunction with manual tests and integration & system tests.

IV. Test Coverage Mining

This section presents the different steps of our approach that are performed for analyzing process logs in order to calculate test coverage.

A. Metric Calculation Process

For calculating test coverage, we use process mining techniques. Process Mining is concerned with building "a strong relation between a process model and 'reality'captured in the form of an event log" [31, p. 41]. Although process mining techniques are usually used to help the business (e.g., [32], [33], [34]), it is used here to guide the development project: By having the BPEL process model and the event logs of all test cases available from the process engine's database, we are able to *replay* the event logs generated from the tests on top of the BPEL process model. Out of the many possible motivations to do a replay, our goal is to extend our model with frequency information [31, p. 43].

Accordingly, our approach is divided into four sub-steps, which are described in the following sections:

1) Build the BPEL Process Model Syntax Tree from its XML representation (BPEL Analysis),
2) Wait until the whole event log has been written,
3) Fetch the event log from the Process Engine (Data Gathering),
4) Replay the event logs on top of the BPEL Process and calculate coverage metrics (Mining).

B. BPEL Analysis

Within this step, the BPEL XML representation is read and the control-flow graph is being constructed as described by the block-based structured activities. For example, activities contained in *sequence* activity are chained together by control-flow links. The construction of the control-flow graph is the same as for the instrumentation approach to measuring BPEL test coverage [13] and thus takes the same time to build. All BPEL Models are accessible via the process engine's repository and can be extracted as part of the coverage mining. This guarantees that the event logs match the process model versions exactly.

C. Wait for the Event Log

Unfortunately, the used BPEL engine writes the event log asynchronously and delayed during process execution. This means that there is a delay between process completion and the event log being written to the database. The persistence interval can be configured. In the initial version of process log analysis for test coverage calculation [1], a fixed delay before reading the event logs was introduced that was a long as the configured persistence interval. This configuration specific delay is a fixed cost penality before test coverage calculation starts. The improved version of our implementation actually queries the event log as long as the end event for the last process instance has been written. This should reduce the wait time on average by half.

D. Data Gathering

The BPEL processes of the industry project, which we use in our experiment, uses Informatica ActiveVOS [18] as its BPMS. ActiveVOS is a process engine fully compliant with the BPEL 2.0 and BPEL4People standards and stores all data – especially all available process models in all versions, active and completed process instances, and event logs – in a relational database. This allows access to and analysis of the available data that can be mined for calculating test coverage metrics. For different persistence settings ActiveVOS stores different lifecycle events for every BPEL activity, which include *ready to execute*, *executing*, *completed*, *faulting* and *will not execute*. In addition, there are two more event types for links (edges for graph-based modeling; *link evaluated to true* and *link evaluated to false* and the same for loop and branching conditions (*condition evaluated to true* and *condition evaluated to false*). Besides the event type, the event timestamp, the corresponding process instance, and an internal activity or link identifier is logged.

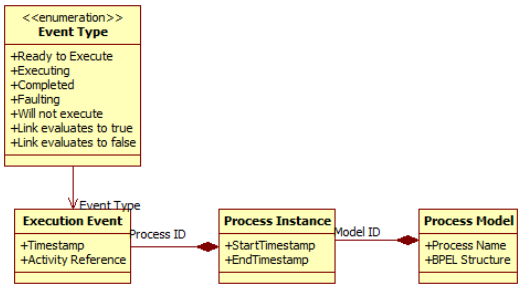


Figure 2. Conceptual Data Model of the Process Engine being used.

This means that all necessary data is available for reconstructing the execution of a process instance and thereby calculating the test coverage metrics: For calculating activity and handler coverage, all *completed*, and *faulting* events need to be fetched for a given test run. For calculating link coverage both *link evaluated* events need to be fetched and for calculating

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

branch coverage, both *condition evaluated* events need to be fetched as well.

All other event types – especially *Will not execute* events can be ignored, which allows to use all engine settings except for "no logging." The underlying conceptual data model, as it is implemented in the ActiveVOS engine, is shown in Figure 2.

E. Replay & Metric Calculation

Test Coverage Metrics – as defined by Lübke et al. [13] – is calculated with the data extracted in the previous steps. At first, all activities, handlers and conditional links in the syntax tree are marked as not executed. In the second step, all events are being applied to the syntax tree and all activities and handlers that have a corresponding *completed* or *faulting* event are marked as being executed. Also, conditional BPEL links for graph-based modeling are marked with the link evaluation events. However, every link can carry two different markers: one if the condition was evaluated to true and another if the condition was evaluated to false. Because links without a condition are excluded from the coverage metric, they are ignored from further analysis.

Similarily, loops are being marked according to the *condition evaluated* events. During this phase, loop activities can be marked as executed twice for calculating the branch coverage in later stages. For *if* or *ifElse* branches and for most loops markers can be set, whenever the condition is evaluated to true; except for the *repeatUntil*, which follows an inverted boolean logic, and the markers are set if the condition is evaluated to false. The only exception is the parallel *forEach* loop, in which the activity identifier contains the number of the currently executed parallel branch. If a counter larger than one is encountered, the *forEach* activity is marked as executed at least twice.

After all events have been replayed on top of the syntax tree, the coverage metrics can be calculated. The easiest test coverage metric to compute is Activity Coverage C_A: The syntax tree is traversed and all basic activities are counted, which are marked (A_m) and which are not marked (A_u) as shown in equation (1).

$$C_A := \frac{|A_m|}{|A_m| + |A_u|} \tag{1}$$

This metric can be filtered by basic activity type. For example, the coverage of all executed *invoke* activities can be calculated as shown in equation (2).

$$C_{invoke} := \frac{|A_m^{Invoke}|}{|A_m^{Invoke}| + |A_u^{Invoke}|} \tag{2}$$

Similarily, Handler Coverage C_H can be calculated by searching the syntax tree for handlers that have been successfully executed as shown in equation (3): The coverage is the proportion of executed handlers in relation to all handlers.

$$C_H := \frac{|H_m|}{|H_m| + |H_u|} \tag{3}$$

With the given Process Engine it was important to not mark event handlers when they are *ready to execute* because this event will be triggered by the Process Engine whenever the context of the handler gets activated and the handler might be triggered and not when the handler really starts executing or is completed. We use the completion events of the first contained basic activity in an event handler. A handler has been executed, if and only if its first basic activity has been completed successfully or unsucessfully.

This metric can again be filtered for different handler types, e.g., fault handlers, as shown in equation (4).

$$C_{faultHandler} := \frac{|H_m^{fault}|}{|H_m^{fault}| + |H_u^{fault}|} \tag{4}$$

Link Coverage C_L as defined in equation (5) determines what fraction of conditional links in *flow* activities has been evaluated to *true* and *false* respectively.

$$C_L := \frac{|L_m^+| + |L_m^-|}{|L_m^+| + |L_m^-| + |L_u^+| + |L_u^-|} \tag{5}$$

The ActiveVOS BPEL engine logs *transition condition evaluated events*, which also contain the evaluation results. This is very different compared to the instrumentation approach, which requires heavy model modifications in order to distinguish between links that have been subject to BPEL's dead path elimination [35] or which have been really evaluated to false. Because of this, link coverage can be easily calculated by traversing the marked syntax tree. The set L is the set of all conditional links, L_m^+ are all conditional links that have been marked as being executed with the condition evaluated to true, L_m^- are all conditional links that have been marked as being executed with the condition being evaluated to false, L_u^+ are all conditional links that are not marked as being executed with the condition being true, and L_u^- are all conditional links that are not marked as being executed with the condition being false.

Branch Coverage C_B metric complements Link Coverage: Branch Coverage includes all edges in the control-flow graph of structured BPEL activities, i.e., *if*, parallel and sequential *forEach*, *while*, and *repeatUntil* activities but does not include the links in the graph-based *flow* activity, which are only covered by link coverage as defined above. The main problem is that this metric needs to count executions of edges that are not necessarily part of the BPEL model: an *if* does not need to have an *else* and the loops have no edges returning to the loop start and can even support parallel execution like the parallel *forEach* loop. Thus, no completion events can be used but instead other events or further analysis of the model are required. In case of the ActiveVOS BPEL engine, all sequential loops are handeled by using *condition evaluation* events similar to the calculation of link coverage. Parallel *forEach* loops can be measured by parsing the activity identifier in the event, which contains an instance number: If the instance counter is larger than 1 the *forEach* "loops" more than once.

Branch Coverage can be calculated according to Equation (6): The number of *forEach* (F), *repeatUntil* (R) and *while* (W) activities, which have been marked as not executed,

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

executed once or executed more than once are divided by all possible markers, which are three markers for the *forEach* and *while* activities, which can have arbitrary loop counts, and two markers for the *repeatUntil* activitiy, which must be looped at least once.

$$C_L := \frac{|F_m^{0,1,*}| + |W_m^{0,1,*}||R_m^{1,*}|}{3 \cdot |F_a| + 3 \cdot |W_a| + 2 \cdot |R_a|} \quad (6)$$

Depending on the BPEL modeler's choice between using BPEL's *flow* activity or the block-structured activities, link or branch coverage is more meaningful.

When conducting another research project, we accidentially found that – in contrast to other programming languages – the branch coverage is not stricter than the activity coverage. If an event handler – and consequently its basic child activities – is not executed, branch coverage can be complete but the basic activity coverage is not. The problems arises due to the original split of control-flow related coverage metrics into branch, link and handler coverage.

Therefore, we define the new coverage metric Conditional Coverage for BPEL processes that unifies all conditional control-flow metrics. The measurement of all values follows the rules as described for the other coverage metrics above.

F. Example

To illustrate the replay of the event log on top of the process model we assume three test cases for the example BPEL process as shown in Figure 3. The first test case tests the VIP Customer, the second one the Non-VIP Customer and the third the VIP Customer with a problem when booking bonus points.

The "completed" events generated by the Process Engine for the first test case are

1) Receiving the start message (*receive* activity "ReceiveInput"),
2) creating a message for storing the points(*assign* activity "PrepareSavePointsEarned"),
3) calling a service to credit points (*invoke* activity "SavePointsEarned"),
4) creating the response message (*assign* activity "PrepareReplyforVIPCustomer"),
5) completing the sequence within the *if* (*sequence* activity "Sequence"),
6) completing the *if* (*if* branch "If and *if* activity "Is_VIP_Customer"),
7) sending the reply (*reply* activity "Reply"), and
8) finally completing the main sequence (*sequence* activity "main").

As can be seen in the traces in Figure 3, the completion events are differently ordered than the definition in the BPEL process model: structured activities like a *sequence* or an *if* are completed after all their child activities have been completed. The replay algorithm needs to take this into account when replaying the event log against the process model.

Taking the event log for the first test case and replaying it on top of the BPEL process model yields the markings as illustrated in the left of Figure 4. Additionally replaying the second and third test case yields the markings as shown on the right hand side of the same figure. The numbers in the markers denote how often the activity has been executed. With these three test cases, all basic activities are covered, i.e., all basic activities have been executed at least once, all branches are covered, i.e., both the "if" and "else" branch have been executed, and all handlers are covered, i.e., the "catch" handler is executed at least once.

V. Comparison to Instrumentation

When we compare our approach to instrumentation (see Figure 5), there are many parts of the calculation that are similar or even the same. Instrumentation would initially load the BPEL process model and construct a syntax tree. However, it would then change the process model by introducing service calls that signal the internal process state to the test framework. During run-time these service calls are equivalent to log events. These events are replayed on the process model in both approaches. Thus, the main differences are that

- instrumentation needs to change the BPEL process model while process mining does not,
- as a consequence instrumentation needs to build the syntax tree prior to the test run and a service receiving all markers must be active during the whole test while process mining can perform all activities after the test run is completed, and
- the events are collected in the instrumentation approach by signaling service calls instead of extracting all event logs with one database query like in our approach. For a test run, the instrumentation approach requires at least as many service calls for signaling the process state as the number of executed basic activities depending on the coverage metrics that shall be calculated.

Due to these conceptual differences, our approach is more flexible than instrumentation because it defers the decision whether to calculate test coverage on a given test run: it is possible that coverage is calculated without preparation after testing has completed and if the event logs are still available.

We expect our log analysis approach also to be overall faster than the instrumentation approach: Making and answering many fine-grained service calls is time-consuming as outlined above. Being able to fetch all events from the Process Engine's event log at once should yield better performance. In addition, our approach does not slow down the execution of the executable processes because they behave as they are implemented and are not changed by an instrumentation process and their run-time behavior is not altered by introducing probes. This means that no additional error sources (e.g., by defects in the instrumentation) or different behavior (e.g., in parallel activities by instrumentation code) can occur. This hypothesis is tested in an experiment described in the next section.

VI. Experiment

In order to evaluate the presented approach, we conducted an experiment that is described in this section.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

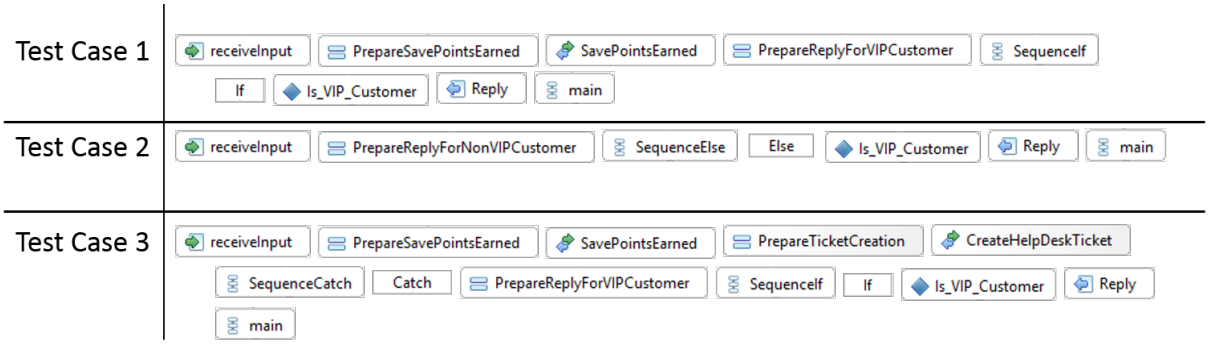


Figure 3. Event Traces for Different Test Cases.

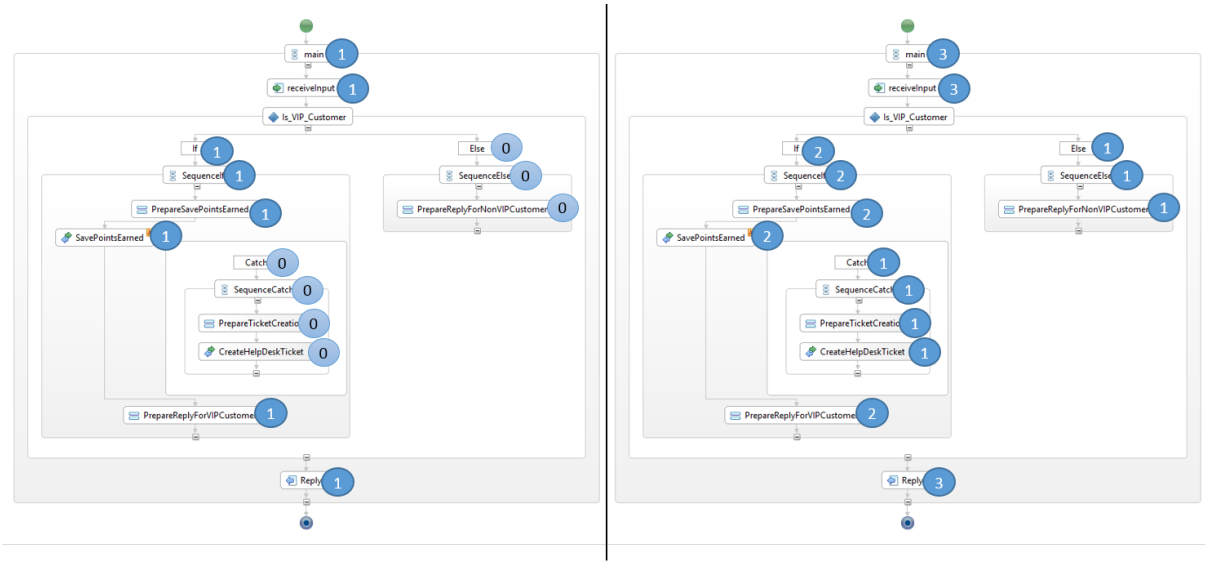


Figure 4. Markers for First Test Case (left) and all Test Cases (right).

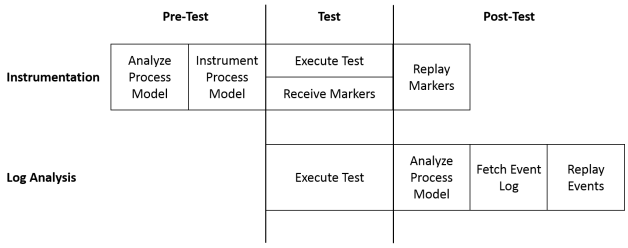


Figure 5. Comparison of Instrumentation and Mining.

A. Experiment Description & Design

For evaluating the performance implications of our approach, we conduct an experiment, in which we want to answer the following research questions:

- RQ1: What is the associated overhead for instrumentation-based coverage calculation?
- RQ2: What is the associated overhead for mining process coverage?
- RQ3: When is the associated overhead for mining process coverage less than for instrumentation-based coverage calculation?
- RQ4: Does the size of the test suite influence the overhead of mining coverage calculation?
- RQ5: Does the size of the test suite influence the overhead of instrumentation-based coverage calculation?

In order to find answers to these questions we define a two factor/two treatments with-in group experiment design: The first independent variable is the coverage method (Instrumentation vs. Log Analysis) and the second is the test suite size. The dependent variable is the execution time of the measured test suites.

As subjects we used 6 BPEL processes, for which tests based on classification trees are available [24]. Classification Trees allow for a generator-based approach for creating test suites. By randomly selecting a subset of test cases, test suites of configurable sizes can be generated. This yields the advantage that test suites of arbitrary sizes can be generated, so that the test suite size can be controlled in the experiment and each process can be tested with different test suites. Four processes have been developed within the industry project

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Terravis, which is an industrial project that develops and runs a process-integration platform between land registers, notaries, banks and other parties across whole Switzerland [14]. Two additional processes, which had classification trees for automated testing, are taken from Schnelle [15]. Latter were originally developed using the Eclipse BPEL Designer and Apache ODE but the vendor-specific configuration was added for ActiveVOS as part of this experiment in order to use the same environment including process engine and test coverage tools.

Process descriptions of all processes that were subjects in this experiment are are shown according to the process classification proposed by Lübke et al. [36] in Table I.

B. Data Collection

1) Environment and Measurement Process: For running the experiment we set up a process engine on a dedicated virtual server together with the required infrastructure, e.g., the tools for measuring test coverage.

The experiment was conducted by executing the following steps for every test suite:

1) Reset database and start BPMS,
2) Instrument the deployment unit,
3) Deploy the instrumented deployment unit,
4) Run the test suite with the marker collector,
5) Deploy the original deployment unit,
6) Run the test suite,
7) Wait for process log and calculate coverage,
8) Shutdown BPMS.

We chose to alternate the deployments of the instrumented and non-instrumented process versions in order to not allow the BPMS to optimize the deployment by reusing the old process definitions.

For every process, we generated random test suites with the sizes $n \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 50, 75, 100\}$ if possible. Some processes had only a smaller number of possible test cases, thereby the experiment could only use test suites with max. 25 and respectively 50 test cases for these processes.

We repeated these test runs 20 times in order to build representative mean values for all time measurements. All in all, 2920 test suite runs were made.

We used a virtual machine with 2 virtual CPUs and 4 GiByte of RAM running on Kubuntu with Informatica ActiveVOS 9.2 and MySQL for all our test executions. The search indexing of both the host and virtual machine operating system were disabled in order to not have load unrelated to the experiment and the computer was taken offline in order to further shield it from unexpected load.

ActiveVOS was configured with all necessary settings for executing all processes. This especially includes custom project-specific extensions and service-wide configured settings. The configuration also specified a 1 second write delay for storing the event log.

2) Sample Implementation: We implemented a tool that performs the previously defined test coverage calculation. The tool connects to the database of the process engine and extracts all relevant information. After the tests have been completed, the tool extracts the events for all newly created process instances. It expects that the tested processes have been configured appropriately to at least store the required events. The coverage measurement tool, which uses log analysis, is available as open source[1].

The implementation is highly dependent on the process engine being used. The available process log data and its format is defined by vendors because it is not specified in any standard. As outlined in the previous section, post-processing of the event log data is required in order to properly resolve the referenced activities.

One additional problem we encountered while developing the sample implementation was BPEL's lack of unique identifiers for activities: The activity labels are not necessarily unique and can be defined by the designer without uniqueness constraints. Due to this, BPMS vendors are forced to build their own ways of identifying activities. We had to reverse-engineer the way the used process engine creates identifiers in the process log. Internally, our implementation uses XPath expressions that evaluate to a single activity by either matching a unique name – if one exists – or the position of the activity in BPEL's XML tree. We wrote a mapper, which rewrites the event log's activity identifiers to valid XPath expressions. This step is highly specific to the process engine being used and requires reverse-engineering the format and construction rules for the proprietary identifiers.

In contrast to our first experiment published previously [1], which used a fixed waiting time, we improved the waiting process by reading the highest process identifier from the database and see whether a process completion or failing event was written for that process instance. Because no other events can follow, this means that the whole event log is available. In order to guard against test cases that fail to complete a process, we added a maximum wait time that equals the write delay of the process log.

However, we also needed to re-implement the instrumentation tool: Because the original BPELUnit tool for measuring test coverage [13] did neither support vendor extensions nor the deployment artefacts of the used process engine, we needed to re-implement the instrumentation tool with full support for these features, which are used by the industry project.

C. Results

The mean execution times of our measurements (calculated in milliseconds) are shown in Table II. T or S indicate the process set (Terravis or Schnelle), 1 to 4 indicate which process from this set, and N, I or L indicate normal execution (N) or the coverage measurement method (I for instrumentation and L for log analysis.)

For all other chosen test suite sizes, log analysis performs faster than instrumentation.

By subtracting the normal execution time of a test suite we derive the absolute overhead (calculated in ms) as shown in Table III. In general, the numbers for log analysis are much lower than for instrumentation and do not increase that much. The highest overhead for log analysis is 4513ms in contrast

[1] http://www.daniel-luebke.de/net.bpelunit.tools.coveragecalculator.zip

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. PROCESS CLASSIFICATION OF SUBJECTS IN THIS EXPERIMENT

	Online Shop (S1)	Credit Approval (S2)	Land Register Notifications (T1)	Depot Check (T2)	Transfer Approval (T3)	Register of Commerce (T4)
Version	-	-	-	-	-	-
Domain	E-Commerce	Banking	Mortgage Transactions			Register of Commerce
Geography	None	None	Switzerland			
Time	2016	2016	2018			
Boundaries	-	-	Cross-Organizational	Within-Dep.	Cross-Organizational	
Relationship	No call	No call	Is being called	No Call	Calls another	Calls another/ Is begin called
Scope	Core	Core	Auxilliary	Auxilliary	Core	Core
Purpose	Execution					
People Involvement	None	None	None	None	None	None
Process Language	BPEL 2.0	BPEL 2.0	BPEL 2.0 plus vendor extensions			
Execution Engine	Apache ODE		Informatica ActiveVOS 9.2			
Model Maturity	Illustrative	Illustrative	Productive			
Basic Activities	19	25	84	46	33	39
Structured Activities	8	12	89	46	34	33
Non-linear Struct.A.	6	14	46	20	9	4

TABLE II. TOTAL MEAN EXECUTION TIME (ms)

#TC	S1-N	S1-I	S1-L	S2-N	S2-I	S2-L	T1-N	T1-I	T1-L	T2-N	T2-I	T2-L	T3-N	T3-I	T3-L	T4-N	T4-I	T4-L
1	1973	3745	2613	1933	3589	2600	4534	7704	5231	2600	5003	3329	3888	6506	4615	4543	8019	5328
2	2244	4958	2941	2181	4473	2808	5190	10123	5927	2975	6675	3731	5421	9759	6196	4825	8535	5533
3	2287	5078	2919	2319	5516	3055	5162	10032	5950	3057	7405	3871	6606	11792	7440	5134	10482	5820
4	2409	5503	3111	2405	5831	3091	5733	11767	6547	3537	10761	4307	8004	14710	8743	5570	13540	6366
5	2471	5645	3107	2389	5890	3050	5499	11863	6427	4035	12886	4968	9405	17883	10132	5562	12774	6317
6	2732	7017	3480	2510	6486	3174	6050	13996	7012	3755	11545	4577	10721	20697	11538	5901	15428	6716
7	2853	7567	3570	3150	10248	3896	6524	15432	7455	4220	14224	5118	12040	23447	12866	5853	15131	6709
8	3156	9411	3795	2706	7668	3373	6827	16618	7828	4288	14442	5176	12967	24095	13740	6452	18699	7294
9	3142	9039	3887	3158	10429	3973	6801	16765	7838	4737	18490	5708	14951	29809	15826	6731	20277	7522
10	3250	9673	3989	3010	9692	3742	7576	18652	8656	5091	19469	6211	15859	30656	16729	7097	21889	7928
25	5164	19569	5914	4725	17369	5522	11187	34880	12874	8060	41445	9368	35033	70271	36183	10623	44380	11731
50	-	-	-	7193	36245	8113	17349	62530	19976	11727	73215	13571	-	-	-	-	-	-
75	-	-	-	7623	37552	8571	23502	90296	27161	15244	105404	17536	-	-	-	-	-	-
100	-	-	-	-	-	-	29746	117730	34259	18966	141603	21926	-	-	-	-	-	-

for up to 122637ms for instrumentation. The overhead is the largest for the second Terravis process (T2) for process log analysis while it is the largest for instrumentation with the first Terravis process (T1).

We calculated the relative overhead for the processes by dividing the absolute overhead by the normal test suite execution time as shown in Table IV. While for larger test suites the relative overhead increases with instrumentation, it decreases for log analysis. Relative overhead of instrumentation ranges between 67.3% and 647.3%, while it ranges bwetween 3.3% and 34.3% for log analysis.

The measurements grouped by coverage calculation method and process for all test suite runs are shown in Figure 6 side by side for comparison: Test suites with more test cases expectedly take longer to execute and log analysis is always faster than instrumentation.

The absolute and relative overhead of both coverage calculation methods are shown in Figure 7. Different colors in both charts indicate different processes. The absolute overhead shows clusters of overhead times that are associated with a test suite. As can be seen the values for both the absolute – and following from that – the relative overhead is always higher for the instrumentation approach.

In order to answer RQ3 we performed a paired, two-sided Wilcoxon hypothesis test with the null hypothesis H_0 being that no difference exists in the test suite execution times when using instrumentation or log analysis.

We calculcated the effect size as the absolute difference in execution time between both methods as well as the p-value for all combinations of test suite size and process (see Table V).

In the next step we analyzed the overhead of test coverage calculation in relation to the whole test suite size in terms of test activities, i.e., the activities in a test suite across all test cases.

Figure 8 shows the relationship between number of test activities and test execution time. The relationship is nearly perfectly linear: The more test activities are executed as part of a test suite the longer test execution takes. However, the slope of the linear relationship depends on the process under test.

In the next step we analyzed the relationship between the number of test activities and the absolute overhead of coverage calculation as shown in Figure 9 (please note that the y-axis scale is different for instrumentation and log analysis.) Both instrumentation and log analysis have a nearly perfectly linear increase of test duration. However, log analysis has a much

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE III. ABSOLUTE MEAN OVERHEAD OF TEST COVERAGE CALCULATION (ms)

#TC	S1-I	S1-L	S2-I	S2-L	T1-I	T1-L	T2-I	T2-L	T3-I	T3-L	T4-I	T4-L
1	1772	640	1656	666	3170	697	2403	728	727	2617	785	3475
2	2714	697	2291	626	4933	737	3700	757	776	4338	708	3710
3	2791	632	3197	736	4870	788	4348	813	834	5186	686	5348
4	3094	702	3427	687	6034	814	7223	769	739	6705	796	7970
5	3174	636	3502	661	6364	928	8851	933	727	8478	755	7211
6	4284	747	3976	664	7945	962	7790	822	817	9976	815	9527
7	4715	717	7098	746	8908	931	10004	898	827	11408	856	9278
8	6255	639	4962	667	9792	1002	10154	888	773	11128	842	12248
9	5898	746	7272	815	9964	1037	13753	971	875	14857	792	13546
10	6424	739	6681	732	11076	1080	14378	1120	870	14797	831	14792
25	14406	750	12644	797	23693	1687	33385	1309	1151	35238	1108	33757
50	-	-	29052	921	45182	2627	61488	1844	-	-	-	-
75	-	-	29929	949	66793	3658	90161	2292	-	-	-	-
100	-	-	-	-	87985	4513	122637	2960	-	-	-	-

TABLE IV. RELATIVE MEAN OVERHEAD OF TEST COVERAGE CALCULATION

#TC	S1-I	S1-L	S2-I	S2-L	T1-I	T1-L	T2-I	T2-L	T3-I	T3-L	T4-I	T4-L
1	0.90	0.32	0.86	0.34	0.70	0.15	0.93	0.28	0.19	0.67	0.17	0.77
2	1.21	0.31	1.06	0.29	0.95	0.14	1.25	0.25	0.14	0.80	0.15	0.77
3	1.23	0.28	1.40	0.32	0.95	0.15	1.42	0.27	0.13	0.79	0.13	1.04
4	1.29	0.29	1.43	0.29	1.05	0.14	2.04	0.22	0.09	0.84	0.14	1.43
5	1.29	0.26	1.47	0.28	1.16	0.17	2.20	0.23	0.08	0.90	0.14	1.30
6	1.57	0.27	1.59	0.27	1.31	0.16	2.08	0.22	0.08	0.93	0.14	1.62
7	1.66	0.25	2.26	0.24	1.37	0.14	2.37	0.21	0.07	0.95	0.15	1.59
8	1.99	0.20	1.84	0.25	1.44	0.15	2.37	0.21	0.06	0.86	0.13	1.90
9	1.88	0.24	2.31	0.26	1.47	0.15	2.91	0.21	0.06	0.99	0.12	2.02
10	1.98	0.23	2.22	0.24	1.46	0.14	2.83	0.22	0.05	0.93	0.12	2.09
25	2.79	0.15	2.71	0.17	2.12	0.15	4.15	0.16	0.03	1.01	0.10	3.19
50	-	-	4.06	0.13	2.61	0.15	5.25	0.16	-	-	-	-
75	-	-	3.92	0.13	2.84	0.16	5.92	0.15	-	-	-	-
100	-	-	-	-	2.96	0.15	6.47	0.16	-	-	-	-

TABLE V. STATISTICAL ANALYSIS OF DIFFERENCES BETWEEN INSTRUMENTATION AND LOG ANALYSIS (EFFECT IN MS, P-VALUES)

#TC	S1-delta	S1-p	S2-delta	S2-p	T1-delta	T1-p	T2-delta	T2-p	T3-delta	T3-p	T4-delta	T4-p
1	1132	1.91×10^{-6}	989	1.91×10^{-6}	2473	1.91×10^{-6}	1674	1.91×10^{-6}	1891	1.91×10^{-6}	2691	9.56×10^{-5}
2	2017	1.91×10^{-6}	1665	1.91×10^{-6}	4196	1.91×10^{-6}	2943	1.91×10^{-6}	3563	9.56×10^{-5}	3002	1.91×10^{-6}
3	2159	1.91×10^{-6}	2461	9.56×10^{-5}	4082	1.91×10^{-6}	3535	1.91×10^{-6}	4352	1.91×10^{-6}	4662	1.91×10^{-6}
4	2392	9.56×10^{-5}	2740	1.91×10^{-6}	5220	9.56×10^{-5}	6454	1.91×10^{-6}	5966	1.91×10^{-6}	7174	1.91×10^{-6}
5	2538	1.91×10^{-6}	2841	1.91×10^{-6}	5436	1.91×10^{-6}	7918	1.91×10^{-6}	7751	1.91×10^{-6}	6456	1.91×10^{-6}
6	3537	1.91×10^{-6}	3312	1.91×10^{-6}	6984	1.91×10^{-6}	6968	1.91×10^{-6}	9160	9.56×10^{-5}	8712	1.91×10^{-6}
7	3998	1.91×10^{-6}	6352	1.91×10^{-6}	7977	1.91×10^{-6}	9106	1.91×10^{-6}	10581	1.91×10^{-6}	8423	1.91×10^{-6}
8	5616	1.91×10^{-6}	4295	9.56×10^{-5}	8790	1.91×10^{-6}	9266	1.91×10^{-6}	10355	1.91×10^{-6}	11405	1.91×10^{-6}
9	5152	9.56×10^{-5}	6456	9.56×10^{-5}	8927	1.91×10^{-6}	12782	1.91×10^{-6}	13983	1.91×10^{-6}	12754	1.91×10^{-6}
10	5685	1.91×10^{-6}	5949	1.91×10^{-6}	9996	1.91×10^{-6}	13258	1.91×10^{-6}	13927	1.91×10^{-6}	13961	1.91×10^{-6}
25	13655	1.91×10^{-6}	11847	1.91×10^{-6}	22006	1.91×10^{-6}	32076	1.91×10^{-6}	34087	9.56×10^{-5}	32649	1.91×10^{-6}
50	-	-	28131	1.91×10^{-6}	42555	1.91×10^{-6}	59645	1.91×10^{-6}	-	-	-	-
75	-	-	28981	3.81×10^{-6}	63135	1.91×10^{-6}	87869	1.91×10^{-6}	-	-	-	-
100	-	-	-	-	83471	1.91×10^{-6}	119677	1.91×10^{-6}	-	-	-	-

lower slope, i.e., the overhead increases much less for every additional test activity than instrumentation does. Again, the linear increase depends on the process. The initial penality for log analysis (i.e., test activity count is 0) can be estimated by the linear fitting to approximately 0.5s, which is the expected value: Due to the 1s write delay of the event log the average waiting time for all events to be written to the database is 0.5s.

The different slopes of instrumentation and log analysis lead to different relative overhead as shown in Figure 10. Because the the overhead of instrumentation increases more than the unmeasured test duration, the relative overhead increases when more test activities are executed. In contrast, log analysis increases slower. This results in a decreasing – or for one process nearly constant – relative overhead. However, the relative overhead does not increase or decrease linearly. A logarithmic regression model provides a good fit.

D. Interpretation

1) RQ1: Overhead of Instrumentation: Our measurements for the overhead of instrumentation is in line with already published metrics [13]: The absolute overhead ranges between 1656ms and 122600ms. However, the overhead increases with larger test suites. Thus, the relative overhead increases from 67% to 647% for large test suites. In practice this overhead is considerably large. For nightly builds even a 200% increase of test time would in many environments be deemed impractical. Also research projects, which execute many test suites, e.g., for evaluating different test generation approaches, are impacted heavily.

2) RQ2: Overhead of Log Analysis: Our measurements for the overhead of log analysis demonstrate that the absolute overhead increases and the relative overhead decreases with more test cases. The maximum absolute overhead of 4.5s for

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

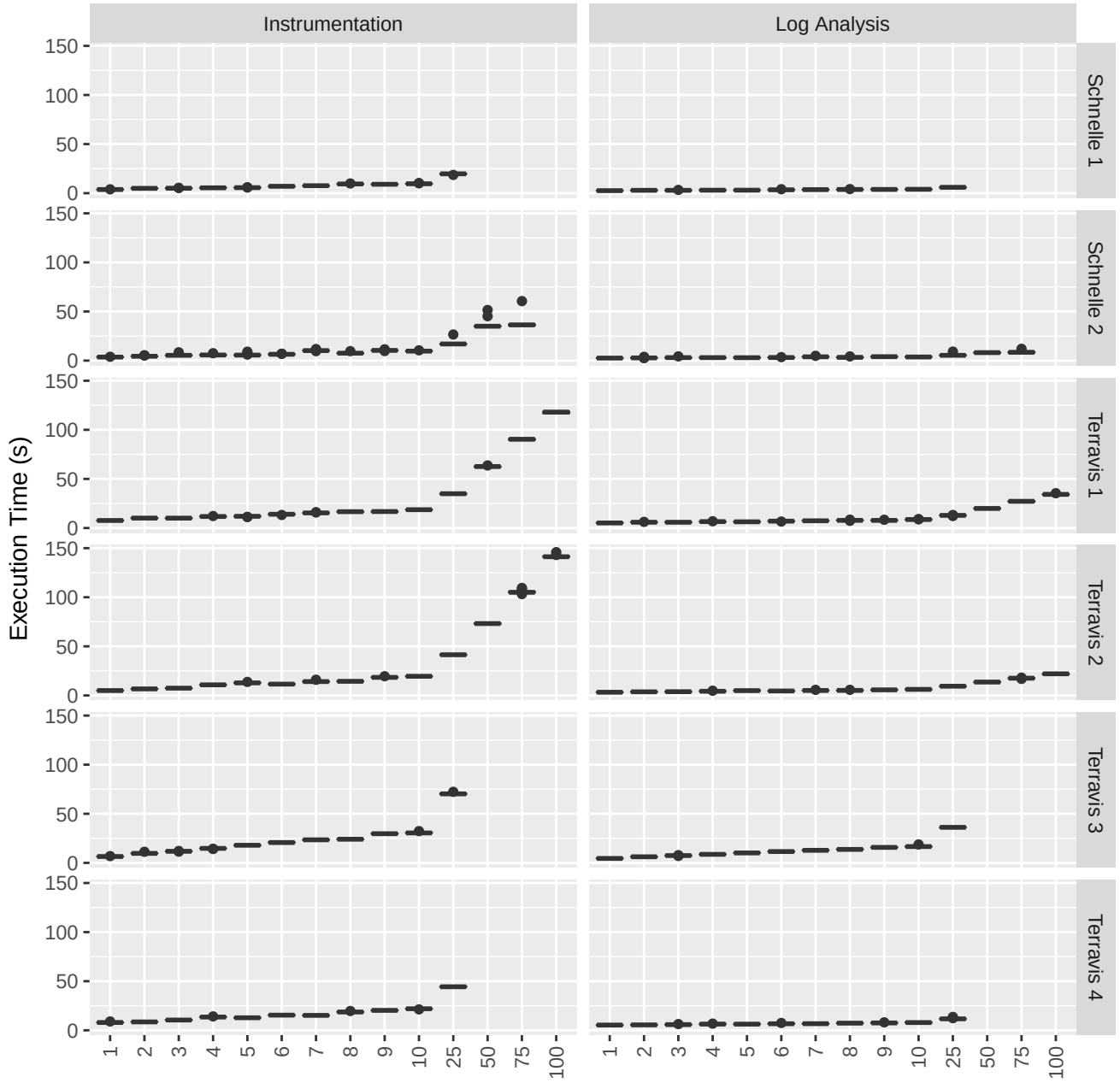


Figure 6. Overall Execution Times (s)

100 test cases the performance penalty is little. This means that measuring approx. 130 test suites of such size would only impose a ten minute overhead (e.g., during nightly builds.) This overhead is much more tolerable in industry projects and research projects, which execute many test suites.

3) RQ3: Relationship between Overhead of Instrumentation and Log Analysis: Our measurements clearly show that log analysis is significantly faster than instrumentation. In the improved version presented in this article, this is also true for trivial test suites, i.e., test suites with only one test case, which were sometimes slower [1] in the unoptimized original version.

While the relative overhead of instrumentation increases with more test cases and reaches 391% (i.e., nearly quintuples the test suite execution time), log analysis imposes 68% overhead in the worst case of a small test suite but decreases to 16% for large test suites. For a further interpretation typical test case sizes in industry are required in order to evaluate typical overhead ranges. Unit test suites for an executable business process in Terravis contain between 1 and 296 test cases. On average a business process is covered by 27.7 test cases. If we take our measurements for 25 test cases

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

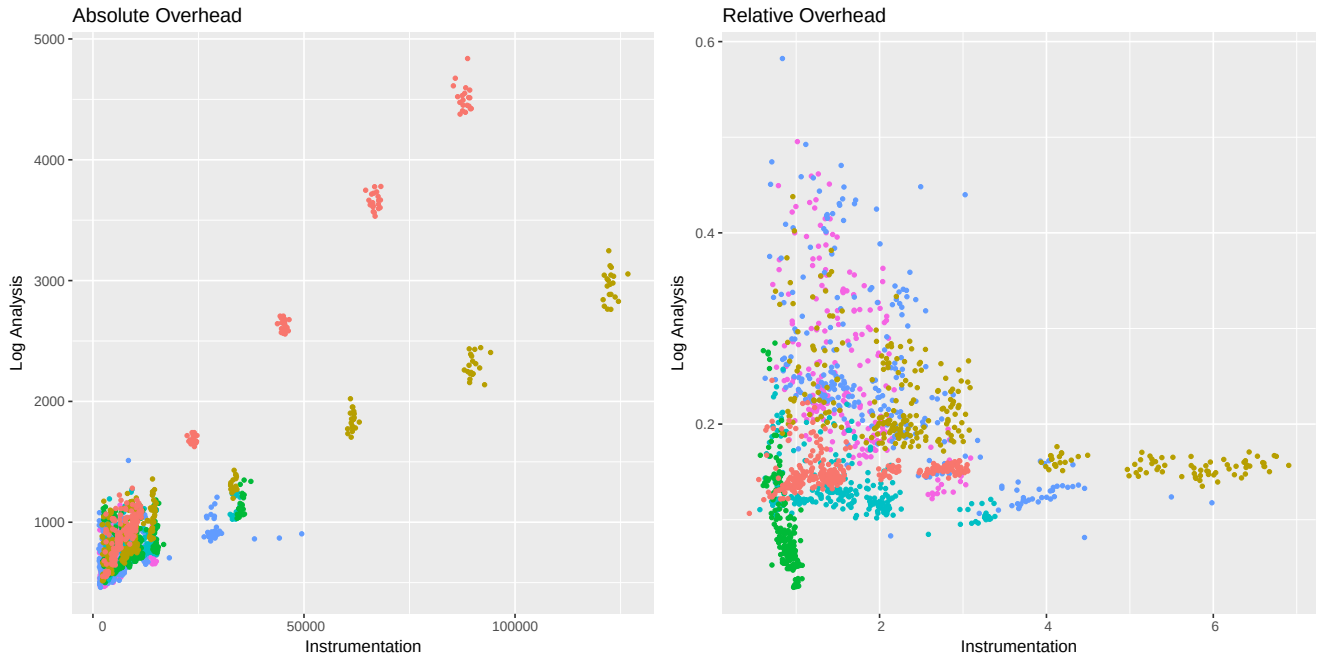


Figure 7. Coverage Measurement Overhead

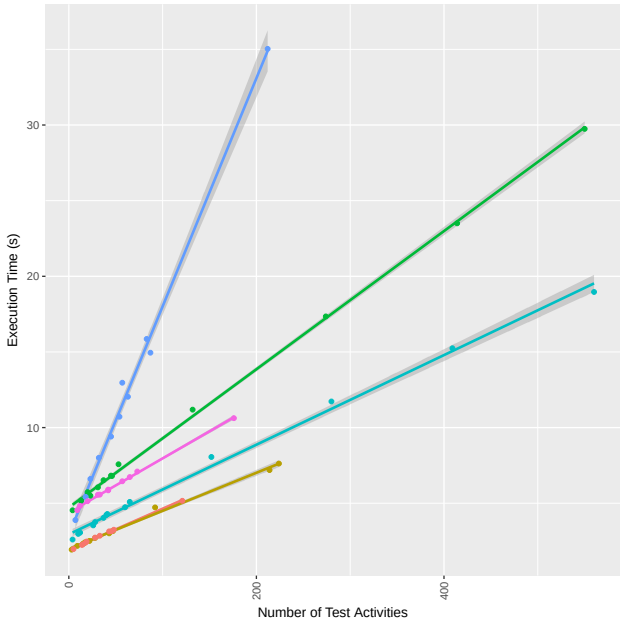


Figure 8. Relationship between Test Suite Execution Time and Number of Test Activities

as a reference the relative overhead is between 24% and 55% for log analysis while it already is between 144% and 268% for instrumentation. This means log analysis has a huge performance benefit when measuring test coverage.

4) RQ4: Influence of Test Suite Size on Instrumentation Overhead: The absolute overhead of an instrumented test run increases linearly with the number of test activities contained in the test suite. However, the linear increase is larger than the linear increase of a normal test run. Therefore, relative overhead of test suite execution time increases non-linearly with an increasing number of test activities. As a result, the instrumentation method does not scale: The larger test suites are getting, the larger the absolute and relative overhead gets. Especially when using test coverage to assess the quality of generated tests, which can easily generate large test suites, the bad scalability will need to be dealt with. For example, test execution needs to be parallalized much sooner than would be otherwise necessary.

5) RQ5: Influence of Test Suite Size on Log Analysis Overhead: Like with instrumentation, the absolute overhead of log analysis increases linearly with the number of test activities. However, the slope is less. The relative increase there gets not-linearly less with more test activities executed as part of a test suite run. Therefore, log analysis can better scale with larger test suites.

E. Threats to Validity

As with every empirical research there are associated threats to validity. While we could increase the number of processes from our initial study [1], our sample mainly consists of processes developed in one project. Thus, the question of generalizability arises.

Since we research technical effects only, the findings should be generalizable to all BPEL processes that execute a minimum threshold number of activities or test cases. The p-value for rejecting the null hypothesis and accepting that log analysis is faster than instrumentation for all test suite sizes is so low that we are confident that replications will find the same results.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

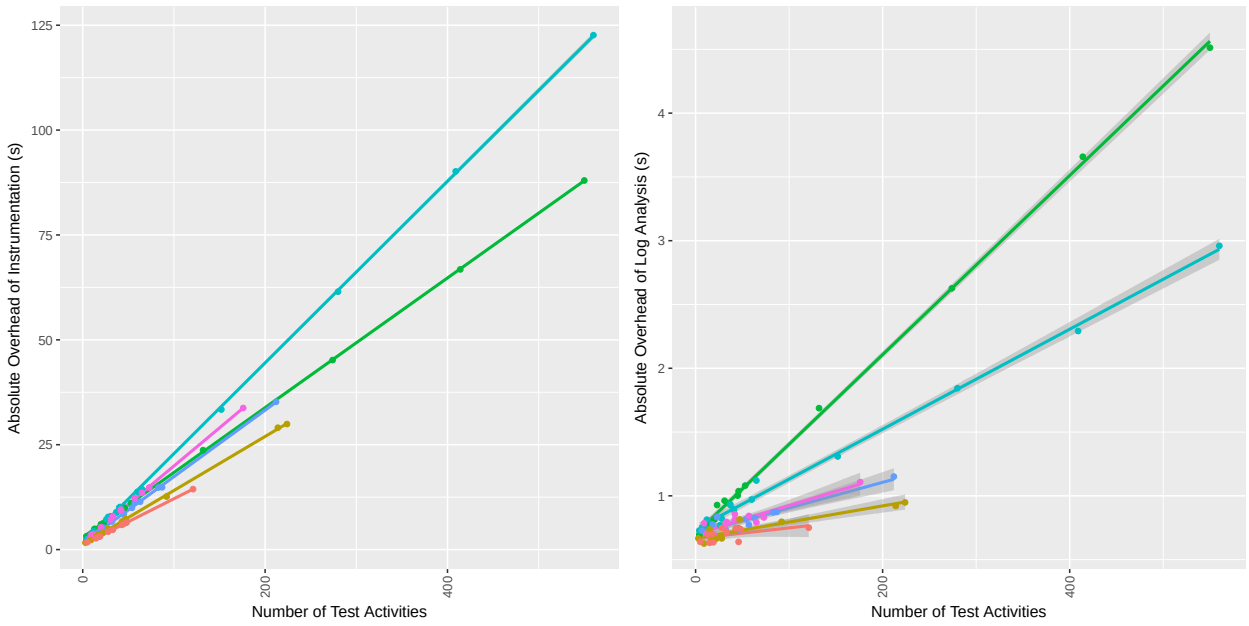


Figure 9. Relationship between Test Suite Execution Time and Absolute Overhead of a) Instrumentation and b) Log Analysis

However, our sample is also constrained to one process engine. This poses the threat of non-portability to other process engines. Therefore, we analyzed Apache ODE, which is an open-source BPEL engine, and found that it emits all necessary events as well [37].

As long as the process engine stores all relevant events that are required for calculating the test coverage metrics, the log analysis can be implemented for such a process engine. To our knowledge, all BPEL engines are able to write event logs that contain the required event types. For every newly supported BPEL engine, a new interpreter of these events needs to be developed. The analysis and replay components can be reused. However, as part of our study we also found that this is also true for instrumentation tools despite the claim that this approach is portable: While BPEL is standardized, its extensions and especially the deployment artefacts are not as we encountered when we tried to measure industry projects.

The presented numbers are clearly only applicable to automated unit tests. While we think it is safe to generalize the absolute overhead to other test scenarios, we expect that the relative numbers to be much smaller: Manual tests take longer for executing the same number of processes, because user interactions require time, which makes the process duration longer. Thus, we do not think that the relative overhead can be generalized to other test types. Even automated integration and system tests are slower because real services usually respond much slower than mocked services that reply a predefined message. For example, automated acceptance tests written for Behavior-Driven Development [38] in the same project take up to 2 minutes to complete per test case [39].

Another threat is the presence of configuration options that heavily impact performance: In the case of the used BPMS - Informatica ActiveVOS - the configurable write delay of the event log can impose longer waiting times. Therefore, environments with a higher configured delay will experience worse log analysis performance because the event log is not immediately available after unit tests are completed. When configured extremely enough, this can lead to a worse performance of log analysis compared to instrumentation.

VII. Conclusion & Future Work

Within this article we evaluated a new approach to mine process event logs – which are usually already written when using a process execution engine – to calculate test coverage metrics of BPEL processes. We demonstrated that the new approach utilizing log analysis is significantly faster than the instrumentation approach. With the enhanced waiting strategy for the event log, this is even true for small test suites, which was not the case in the original version.

Furthermore, the log analysis approach can be used in more scenarios than the instrumentation approach: Because all activities for mining test coverage are performed after the tests are run, it does not matter how the tests are run and when they were run. In contrast, coverage calculation needs a marker collection service running the whole time, which in practice is only feasible during unit tests. Mining the process logs is completely independent of any test automation and can be used for automatic unit tests, automatic integration tests but also manual integration and system tests. The only drawback is, however, that the Process Engine needs to be configured to write the event log for all measured processes.

Although we have implemented test coverage mining for BPEL processes, the approach can be applied to other executable process languages as well: Process engine architectures are the same, e.g., BPMN 2.0 as the successor to BPEL defines other activities and is completely graph based. However, process engines executing BPMN 2.0 are also logging events for executed activities, which can be replayed on top of BPMN 2.0

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

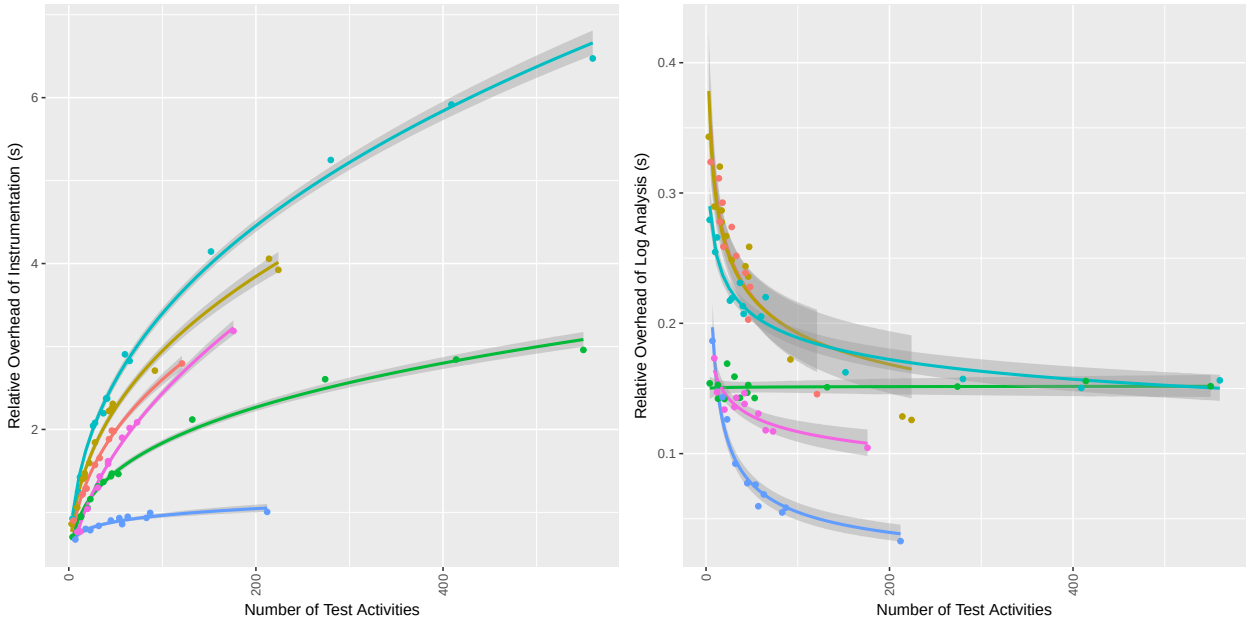


Figure 10. Relationship between Test Suite Execution Time and Relative Overhead of a) Instrumentation and b) Log Analysis

process models. Writing the process mining algorithm should be even simpler, because BPMN 2.0 defines process-wide unique identifiers for activities that are hopefully contained in the event log making reverse-engineering of vendor-specific identifiers obsolete.

Our research implementation is available as open source and is free to use for both researchers and practitioners. Being able to quickly and easily calculate test coverage for many test types allows further research into executable process test methods, e.g., experiments on the influence of different testing approaches on test coverage.

Acknowledgment

This research was not funded by any institution or research grant. The author is neither affiliated with Informatica nor involved in the development of ActiveVOS in any way. The author is part of the Terravis development team.

References

[1] D. Lübke, "Calculating Test Coverage for BPEL Processes With Process Log Analysis," in BUSTECH 2018, The Eighth International Conference on Business Intelligence and Technology, 2018, pp. 1–7.

[2] D. Lübke, "Unit Testing BPEL Compositions," in Test and Analysis of Service-Oriented Systems, L. Baresi and E. D. Nitto, Eds. Springer, 2007, ch. Unit Testing BPEL Compositions, pp. 149–171.

[3] E. Cambronero, J. C. Okika, and A. P. Ravn, "Consistency Checking of Web Service Contracts," Int'l Journal Advances in Systems and Measurements, vol. 1, no. 1, 2008, pp. 29–39.

[4] P. Piwowarski, M. Ohba, and J. Caruso, "Coverage Measurement Experience During Function Test," in Proceedings of the 15th International Conference on Software Engineering, ser. ICSE '93. Los Alamitos, CA, USA: IEEE Computer Society Press, 1993, pp. 287–301.

[5] J. R. Horgan, S. London, and M. R. Lyu, "Achieving software quality with testing coverage measures," Computer, vol. 27, no. 9, Sept 1994, pp. 60–69.

[6] L. C. Briand, Y. Labiche, and Y. Wang, "Using simulation to empirically investigate test coverage criteria based on statechart," in Proceedings. 26th International Conference on Software Engineering, May 2004, pp. 86–95.

[7] Y. K. Malaiya, M. N. Li, J. M. Bieman, and R. Karcich, "Software reliability growth with test coverage," IEEE Transactions on Reliability, vol. 51, no. 4, Dec 2002, pp. 420–426.

[8] X. Cai and M. R. Lyu, "Software Reliability Modeling with Test Coverage: Experimentation and Measurement with A Fault-Tolerant Software Project," in The 18th IEEE International Symposium on Software Reliability (ISSRE '07), Nov 2007, pp. 17–26.

[9] D. Lübke, "Using Metric Time Lines for Identifying Architecture Shortcomings in Process Execution Architectures," in Software Architecture and Metrics (SAM), 2015 IEEE/ACM 2nd International Workshop on. IEEE, 2015, pp. 55–58.

[10] H. Zhu, P. A. V. Hall, and J. H. R. May, "Software Unit Test Coverage and Adequacy," ACM Comput. Surv., vol. 29, no. 4, Dec. 1997, pp. 366–427. [Online]. Available: http://doi.acm.org/10.1145/267580.267590

[11] Z. Li, W. Sun, Z. B. Jiang, and X. Zhang, "BPEL4WS Unit Testing: Framework and Implementation," in ICWS '05: Proceedings of the IEEE International Conference on Web Services (ICWS'05). Washington, DC, USA: IEEE Computer Society, 2005, pp. 103–110.

[12] P. Mayer and D. Lübke, "Towards a BPEL unit testing framework," in TAV-WEB '06: Proceedings of the 2006 workshop on Testing, analysis, and verification of web services and applications. New York, NY, USA: ACM Press, 2006, pp. 33–42.

[13] D. Lübke, L. Singer, and A. Salnikow, "Calculating BPEL Test Coverage through Instrumentation," in Workshop on Automated Software Testing (AST 2009), ICSE 2009, 2009, pp. 115–122.

[14] W. Berli, D. Lübke, and W. Möckli, "Terravis – Large Scale Business Process Integration between Public and Private Partners," in Lecture Notes in Informatics (LNI), Proceedings INFORMATIK 2014, E. Plödereder, L. Grunske, E. Schneider, and D. Ull, Eds., vol. P-232, Gesellschaft für Informatik e.V. Gesellschaft für Informatik e.V., 2014, pp. 1075–1090.

[15] T. Schnelle, "Generierung von BPELUnit-Testsuites aus Klassifikationsbäumen," Master's thesis, Leibniz Universität Hannover, Fachgebiet Software Engineering, 2016.

[16] W. l. Dong, H. Yu, and Y. b. Zhang, "Testing BPEL-based Web Service Composition Using High-level Petri Nets," in 2006 10th IEEE

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

International Enterprise Distributed Object Computing Conference (EDOC'06), Oct 2006, pp. 441–444.

[17] G. M. Kapfhammer, "Automatically and Transparently Distributing the Execution of Regression Test Suites," in Proceedings of the 18th International Conference on Testing Computer Software, June 2001.

[18] Informatica. BPEL Unit Testing. [Online]. Available: http://infocenter.activevos.com/infocenter/ActiveVOS/v92/index.jsp?topic=/com.activee.bpep.doc/html/UG21.html (2016)

[19] Oracle. Oracle BPEL Process Manager Developer's Guide: Testing BPEL Processes. [Online]. Available: https://docs.oracle.com/cd/E11036_01/integrate.1013/b28981/testsuite.htm (2007)

[20] A. T. Endo, A. da Silva Simao, S. d. R. S. de Souza, and P. S. L. de Souza, "Web services composition testing: a strategy based on structural testing of parallel programs," in Testing: Academic & Industrial Conference-Practice and Research Techniques (taic part 2008). IEEE, 2008, pp. 3–12.

[21] B. Miranda, "A Proposal for Revisiting Coverage Testing Metrics," in Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, ser. ASE '14. New York, NY, USA: ACM, 2014, pp. 899–902. [Online]. Available: http://doi.acm.org/10.1145/2642937.2653471

[22] C. Bartolini, A. Bertolino, S. Elbaum, and E. Marchetti, "Whitening SOA Testing," in Proceedings of the the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, ser. ESEC/FSE '09. New York, NY, USA: ACM, 2009, pp. 161–170. [Online]. Available: http://doi.acm.org/10.1145/1595696.1595721

[23] M. M. Eler, A. Bertolino, and P. C. Masiero, "More testable service compositions by test metadata," in Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE), Dec 2011, pp. 204–213.

[24] T. Schnelle and D. Lübke, "Towards the Generation of Test Cases for Executable Business Processes from Classification Trees," in Proceedings of the 9th Central European Workshop on Services and their Composition (ZEUS) 2017, 2017, pp. 15–22.

[25] K. Kaschner and N. Lohmann, "Automatic test case generation for interacting services," in International Conference on Service-Oriented Computing. Springer, 2008, pp. 66–78.

[26] S. Ji, B. Li, and P. Zhang, "Test Case Selection for Data Flow Based Regression Testing of BPEL Composite Services," in Services Computing (SCC), 2016 IEEE International Conference on. IEEE, 2016, pp. 547–554.

[27] M. D. Weiser, J. D. Gannon, and P. R. McMullin, "Comparison of Structural Test Coverage Metrics," IEEE Software, vol. 2, no. 2, Mar 1985, pp. 80–85, copyright - Copyright IEEE Computer Society Mar/Apr 1985; Last updated - 2014-05-17; CODEN - IESOEG. [Online]. Available: https://search.proquest.com/docview/215840674?accountid=14486

[28] Q. Yang, J. J. Li, and D. M. Weiss, "A survey of coverage-based testing tools," The Computer Journal, vol. 52, no. 5, 2009, pp. 589–597.

[29] A. Mockus, N. Nagappan, and T. T. Dinh-Trong, "Test coverage and post-verification defects: A multiple case study," in 2009 3rd International Symposium on Empirical Software Engineering and Measurement, Oct 2009, pp. 291–301.

[30] C. Pavlopoulou and M. Young, "Residual Test Coverage Monitoring," in Proceedings of the 21st International Conference on Software Engineering, ser. ICSE '99. New York, NY, USA: ACM, 1999, pp. 277–284. [Online]. Available: http://doi.acm.org/10.1145/302405.302637

[31] W. van der Aalst, Process Mining – Data Science in Action. Springer, 2016.

[32] S. Schönig, M. Seitz, C. Piesche, M. Zeising, and S. Jablonski, "Process observation as support for evolutionary process engineering," International Journal on Advances in Systems and Measurements Volume 5, Number 3 & 4, 2012, 2012.

[33] M. Jäntti, A. Cater-Steel, and A. Shrestha, "Towards an improved it service desk system and processes: a case study," International Journal on Advances in Systems and Measurements, vol. 5, no. 3 & 4, 2012, pp. 203–215.

[34] E. Bruballa Vilas, Á. Wong, D. I. Rexachs del Rosario, E. Luque, and F. Epelde Gonzalo, "Evaluation of response capacity to patient attention demand in an Emergency Department," International journal on advances in systems and measurements, vol. 10, no. 1&2, 2017, pp. 11–22.

[35] C. Ouyang, E. Verbeek, W. M. Van Der Aalst, S. Breutel, M. Dumas, and A. H. Ter Hofstede, "Formal semantics and analysis of control flow in WS-BPEL," Science of computer programming, vol. 67, no. 2-3, 2007, pp. 162–198.

[36] D. Lübke, A. Ivanchikj, and C. Pautasso, "A Template for Sharing Empirical Business Process Metrics," in Business Process Management Forum - BPM Forum 2017, 2017.

[37] Apache Software Foundation, "ODE Execution Events," 2018. [Online]. Available: http://ode.apache.org/ode-execution-events.html, lastaccessed2018-08-07

[38] D. North, "Introducing BDD," 2006, http://dannorth.net/introducing-bdd. [Online]. Available: http://dannorth.net/introducing-bdd

[39] D. Lübke and T. van Lessen, "Modeling Test Cases in BPMN for Behavior-Driven Development," IEEE Software, vol. Sep/Oct 2016, Sep/Oct 2016, pp. 17–23.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The Financial Credit Risk Assessment Model: Three Perspectives

Eric Mantelaers
Optimizing Knowledge-Intensive Business Processes
Zuyd University of Applied Sciences
Sittard, the Netherlands
eric.mantelaers@zuyd.nl

Martijn Zoet
Optimizing Knowledge-Intensive Business Processes
Zuyd University of Applied Sciences
Sittard, the Netherlands
martijn.zoet@zuyd.nl

***Abstract*—Within recent years, Financial Credit Risk Assessment (FCRA) has become an increasingly important issue within the financial industry. Therefore, the search for features that can predict the credit risk of an organization has increased. Using multiple statistical techniques, a variance of features has been proposed. Applying a structured literature review, 258 papers have been selected. From the selected papers, 835 features have been identified. The features have been analyzed with respect to the type of feature, the information sources needed and the type of organization that applies the features. Based on the results of the analysis, the features have been plotted in the FCRA Model. The results show that most features focus on hard information from a transactional source, based on official information with a high latency. In this paper, we re-address and -present our earlier work [1]. We extended the previous research with more detailed descriptions of the related literature, findings, and results, which provides a grounded basis from which further research on FCRA can be conducted.**

Keywords-Financial Credit Risk Assessment; Business Failure Prediction; Credit Risk Features; DMN Requirements Diagrams (DRD).

I. INTRODUCTION

Within the field of the Financial Credit Risk Assessment (FCRA) there are two main areas of interest. The first main interest, credit rating (or scoring), is used to solve the problem to label companies as bad/good credit or bankrupt/healthy. Credit rating is used not only internally for screening borrowers, pricing loans and managing credit risk thereafter, but also externally for calibrating regulatory capital requirements [2]. The second main interest, bankruptcy (failure) prediction (or business failure prediction or going concern assessment) is intended to predict the probability that the company may belong to a high-risk group or may become bankrupt during the following year(s). Both of them are strongly related and solved in a similar way, namely as a binary classification task. In this paper, both categories of problems are collectively called FCRA, which is a business decision-making problem that is relevant for creditors, auditors, senior management, bankers and other stakeholders.

FCRA is a domain which has been studied for many decades. According to Balcaen and Ooghe [3], there are four main areas with reference to FCRA: (1) Classical paradigm (arbitrary definition of failure, non-stationarity and data instability, sampling selectivity), (2) Neglect of the time dimension of failure (use of one single observation, fixed score output/concept of resemblance/descriptive nature, failure not seen as a process), (3) Application focus (variable selection, selection of modelling method), (4) Other problems (use of a linear classification rule, use of annual account information, neglect of multidimensional nature of failure). The literature on FCRA and business failure dates back to the 1930's [27]. Watson and Everett [4] described five categories to define failure: 1) ceasing to exist (discontinuance for any reason), 2) closing or a change in ownership, 3) filing for bankruptcy, 4) closing to limit losses and 5) failing to reach financial goals. When the FCRA is negative, it is called business failure, which is a general term and, according to a widespread definition, it is the situation that a firm cannot pay lenders, preferred stock shareholders, suppliers, etc., or a bill is overdrawn, or the firm is bankrupt according to the law [5]. There is extensive literature in which this topic has been researched from the perspective of auditors or bankers. On the other hand, rare literature can be found about related literature from an information and decision perspective. The features (variables) which are relevant in the field of FCRA will be analyzed in this paper. In this paper the focus will be on the auditor's, bankers and crediting rating firms, hence forward the term financial industry will be used to describe all three. A combination will be made between the financial industry and an information and decision perspective.

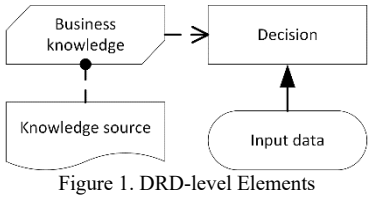


Figure 1. DRD-level Elements

To do so, the DRD model will be used. The reason DMN (Decision Model and Notation) is used, is because it is currently the standard to model decisions. In September 2015, the Object Management Group (OMG) [6] released a new standard for modelling decisions and underlying business logic, DMN The DMN standard is based on two levels; the Decision Requirements Diagram (DRD) level and the Decision Logic Level (DLL). The DRD level consists of four concepts that are used to capture essential information with regards to decisions: 1) the decision, 2) business knowledge, which represents the collection of business logic required to execute the decision, 3) input data, and 4) a knowledge source, which enforces how the decision should be taken by influencing the underlying business logic. The contents of the DLL are represented by the business knowledge container in the DRD level.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The remainder of this paper is organized as follows. Section II contains a description of relevant literature regarding features, feature selection, and techniques with reference to FCRA, from a combined perspective of both the financial industry and information and decision analysts, followed by the research method in Section III. In Section IV, our data collection and analysis will be reported. Subsequently, in Section V, a presentation of the results derived from the applied data analysis techniques will be given. The conclusion (Section VI) closes the article.

II. Literature Review

Feature selection is a critical step in FCRA, which refers to the process that reduces the feature space and selects an optimum subset of relevant features. Three possible methods can be distinguished: 1) human, 2) statistical and 3) hybrid. In the human approach, an auditor decides which features are important and how they relate to each other. The model in 'the head' of the auditor is rebuild into the system. For the statistical approach several alternative methodologies are applied for the feature selection. Tsai [7] compares five well-known feature selection methods used in bankruptcy prediction, which are: 1) *t*-test, 2) correlation matrix, 3) stepwise regression, 4) principle component analysis (PCA) and 5) factor analysis. The hybrid approach applies both the human and statistical manner.

Statistical techniques:
1. Linear discriminant analysis (LDA)
2. Multivariate discriminate analysis (MDA)
3. Quadratic discriminant analysis (QDA)
4. Logistic regression (LR)
5. Factor analysis (FA)

Intelligent techniques:
1. Neural networks (NN)
2. Decision trees (DT)
3. Rough sets
4. Case-based reasoning (CBR)
5. Support vector machines (SVM)
6. Data envelopments analysis
7. Soft computing (hybrid intelligent systems)
8. Operational research techniques
9. Other intelligent techniques

Figure 2. Statistical and Intelligent Techniques

To apply the selected features from the features selection to take the FCRA-decision, different methods are applied. Broadly, these methods are dived into two broad categories: statistical and intelligent techniques [8] [9]. They exist out of multiple sub-categories, see Figure 2. For a detailed description of the techniques we refer to Ravi Kumar and Ravi [8].

Based on literature studied, we developed a model that exists out of three axes that determine the type of features applied. To ground our theory, we first present the end model: the FCRA Model.

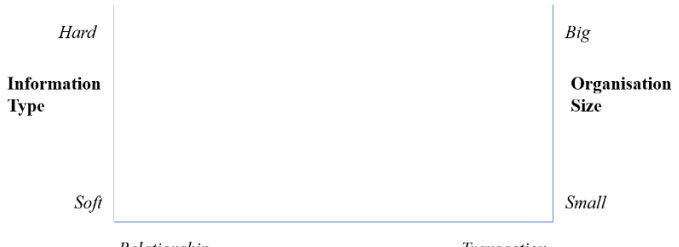


Figure 3. Financial Credit Risk Assessment Model

The first axe describes the type of data that organizations retrieve to make a judgement about the financial credit risk. In the papers of Berger [21][25], the same distinction is made in an information type perspective: hard versus soft (or quantitative versus qualitative) data. Different related names are used in this field:

Quantitative features	Qualitative features
Hard information	Soft information
Financial information	Non-financial information
Accounting information	Non-accounting information

The second axe describes the manner in which this information is retrieved. For example, two manners in which information can be collected are: 1) through face to face contact between a loan officer and the organization's owner and 2) through a form on a website or any other digital manner. The third axe describes the organization size, varying from small to big. The loan decision model of small banks is known to differ from the loan decision model of large banks [12]. According to Berger [10] small organizations (organization size), make use of soft information (information type), based on the relationship with their clients (information source). Bollen et al. [13] recognize four categories of business failures: 1) Tadpole (company failed because it was a basically unhealthy company, 2) Drowned frog (the company is over-ambitious or may show signs of extreme growth, 3) Boiled frog (companies in this category may be failing as a result of external conditions (e.g., disasters), bad economic conditions or fundamental changes in the business environment to which the company has failed to respond adequately, 4) Bullfrog (the companies in this category have drawn a relatively large portion of public attention, because they are often related to fraudulent activities of managers or employees).

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A. Information Type

Hard

According to Petersen [14] hard information is almost always recorded as numbers and is comparable. The durability of information is potentially greater when it is hard. The collection method of hard information is mostly not personal. Hard information is mostly standardized and easy to document and transfer to others [15]. Nemoto et al. [16] also recognize the verifiability which normally is higher in case of hard information. Decision processes which depend upon hard information are easier to automate. Knowing what information you are looking for, and why it is valuable, is essential if information collection and possibly decision making, based on the information is to be delegated. Most features are based on data from the financial statements. Financial statements are, in most organizations, created once or twice a year. Therefore, the data needed to calculate the features is available once or twice a year. This causes an information opacity problem, thereby reducing the effectiveness of the features. Other organizations that also assess the financial credit risk of an organization are banks, credit assessors, etc. Both previously also had to trust numbers that are published once a year. Since this time period is too long for both parties, they searched for solutions to address this problem. According to Berger and Udell [17] hard information may include, as examples, financial ratios calculated from audited financial statements; credit scores assembled from data on the payment histories of the small and medium sized entities (SME) end its owner provided by credit bureaus; or information about accounts receivable from transparent, low-risk obligors that may pledged as collateral by the SME or sold to the financial institution.

Soft

Soft information is mostly relationship-based and not easily quantified [18]. The replacement of soft with hard information inevitably results in a loss of information. The early studies for FCRA were univariate (a specific statistical method applied) studies which had important implications for future model development. These laid the groundwork for multivariate studies. Ravi Kumar and Ravi [8] identify statistical and intelligent techniques to solve the bankruptcy prediction problem. For each type of technique, they describe the way they work. Chen, Ribeiro and Chen [9] summarize the traditional statistical models and state-of-the-art intelligent methods. Auditors can utilize data mining techniques to analyze external (soft) data (e.g., census data, social media, news articles) in their assessments of client business risk, fraud risk, internal controls, going concern [19]. Lu et al. [20] explain the possibilities of data mining (text mining) based on soft information on websites and in financial reports to predict bankruptcy.

Altman et al. [21] describe the value of qualitative (soft) information in SME risk management. They find that qualitative data relating to such variables as legal action by creditors to recover unpaid debts, company filing histories, comprehensive audit report/opinion data and firm specific characteristics, make a significant contribution to increasing the default prediction power of risk models, built specifically for SMEs. Lenders must invest in the production of 'soft information' to supplement the financial data used in these models [22]. Dainelli et al. [23] give a summarization of determinants of SME credit worthiness under Basel rules. As their model does not include qualitative information, future research could aim to set out the qualitative determinants in the rating judgment. Petersen [14] concludes that technology is changing the way we communicate. One of these changes is a greater reliance on hard relative to soft information. Despite this, very little research has been published on the concept of activities used by lenders to gather soft information [24]. Suter [24] studied the collection of soft information by small community banks. He built a conceptual framework existing of four factors to reduce the asymmetric information: 1) Knowledge of business, 2) Knowledge of industry, 3) Knowledge of local market, and 4) Value of the social contract. Angilella and Mazzù [25] structured the non-financial criteria hierarchically on the basis of the risk areas, specific to an innovative firm: development, technological, market, and production. The risk areas considered are: Technological risk, Market risk, Production risk, Innovation indicators, Financial criteria.

Performance

To measure performance, there are several metrics [9]. One of the most important measures is accuracy. In terms of performance, an accuracy rate between 81 and 90% reflects a realistic average performance based on the results of the analyzed studies [26]. The top five bankruptcy models with an accuracy level of more than 80 per cent are [27]: 1) Altman [28], 2) Edmister [29], 3) Deakin [30], 4) Springate, [28] and 5) Fulmer [29]. All of these only use hard features. Chen et al. [31] find that the use of soft information significantly improves the power of default prediction models.

The same conclusion is realized by Ju and Sohn [32] who proposed to update the credit scoring model based on new features like management, technology, marketability, and business and profitability. Kosmidis and Stavropoulos [33] even got one step further in their conclusion, as they state that factors such as economic cycle phase, cash flow information and the detection of fraudulent financial reporting can evidently enhance the predictive power of existing models. Altman, Sabato and Wilson [21] reach the same conclusion as they state: *"that qualitative data relating to such variables as legal action by creditors to recover unpaid debts, company filing histories, comprehensive audit report/opinion data and firm specific characters make a significant contribution to increasing the default prediction power of risk models built specifically for SMEs."*. This leads us to the first conclusion that the financial industry should not only rely on hard features, which have a time delay, but also on soft information to assess the financial credit risk; see bottom left side in Figure 3. Relationship lending is based on soft information and is best suited for entities that are more opaque; and transactions-based lending is best suited for SMEs that are more transparent [34].

To realize proper research in this area, the researchers have to go beyond the already cumulative features and look at the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

base data. E.g., no longer apply the cumulative feature: current assets but instead build features on the base information such as debtors' information.

B. Information Source

In addition to the type of information available, the data source and its fluidity are also factors. In financial literature, this phenomenon is called "the hardening of soft information" [45]. The concept "the hardening of soft information" states that because personal contact with financial institutions has decreased, therefore they rely more and more on hard quantitative information. Since more banks, credit organizations, and accountants rely on non-personal contacts, this statement is gaining importance.

Thereby underlying the fact that the traditional features are the most useful features to analyze the financial credit risk. The main reason they state to support their claim is the adoption rate of technology.

However, a counter claim can be made that through the adoption of technology, soft information can be more easily collected. For example, through firehose access to social media websites. However, this will depend on the type of soft or hard information one wants to retrieve because not all soft information can be retrieved through social websites, some still might need to be retrieved face to face. Therefore, the bottom part of our model, see Figure 3, indicates the lending technologies, being the manner in which the information is retrieved. A lending technology is *"a set of screening and underwriting policies and procedures, a loan contract structure, and monitoring strategies and mechanisms"* [11]. Examples of lending technologies they apply are: leasing, commercial real estate lending, residential real estate lending, motor vehicle lending, and equipment lending, asset-based lending, financial statement lending, small business credit scoring, relationship lending and judgment lending.

C. Organisation Size

In FCRA literature, from a banking perspective, a distinction is made between the manner in which small and big banks assess the risk. Small banks apply more of a relationship perspective to assess the risk, while big banks apply the analysis of transactions to determine the risk. Although this specific distinction cannot be found in accountancy and lending (firms) literature, the hypothesis is that the same basic rules apply. Therefore, the right axe of the FCRA Model contains the size of the firms, assessing the risk; see Figure 3.

Loans to small businesses have traditionally been based on intimate relationships between borrower firms and lenders, because many of these firms are much more informationally opaque than large firms. Thus, lenders primarily rely on soft information, gathering through long-lasting transaction relationships. For banks it is difficult to obtain detailed information from small firms since the financial reports of small firms are mainly for tax purposes [35].

III. Research Method

The goal of this research is to identify and classify features that have been applied to assess Financial Credit Risk. In addition to the goal of the research, also, the maturity of the research field is a factor in determining the appropriate research method and technique. Based on the number of publications and identified features, the maturity of the FCRA research field can be classified as mature. Mature research fields should A) focus on further external validity and generalizability of the phenomena studied, or B) focus on a different perspective on the constructs and relationships between identified constructs [36]. Current studies have focused on two elements: 1) selecting the best features to predict bankruptcy, while other studies have focused on 2) comparing the efficiency and effectiveness of the different features identified. However, current analysis focuses on two viewpoints: 1) a high abstraction level and 2) a high latency perspective.

Summarized, to accomplish our research goal, a research approach is needed in which the current features are explored, compared and mapped to the FCRA Model.

To accomplish this goal, a research approach is needed that can 1) identify features for FCRA, 2) identify similarities and dissimilarities between features, and 3) map the features to the FCRA Model. The first two goals are realized by applying a structured literature research and the use of a comparison table. The last goal is realized by coding the features identified, based on a priori coding scheme.

IV. Data Collection and Analysis

As stated in the previous section, the goal of this research is to 1) identify features for FCRA, 2) identify similarities and dissimilarities between features for FCRA, and 3) map the features to the FCRA Model.

The selection of the papers has been conducted via the link-tracing methodology [37], more specifically via snowball sampling. The snowballing was applied to take advantage of the social networks of identified respondents to provide a researcher with an ever-expanding set of potential contacts [38]. Snowballing is an effective and efficient form of contact tracing for use in diversity of research methods and designs, and apparently well suited for a number of research purposes [39] - [40]. For both the hard features and soft features two different snowball samplings have been conducted. For the hard features this resulted in 238 papers that were included in the coding. With respect to the soft features this resulted in 20 papers to be selected for coding. For a study to be selected for coding, the study must explicitly address hard and/or soft features for FCRA (see Table II for details). The unit of analysis for coding is a single feature, implying that one study can contribute multiple units of analysis. For example, Alam et al. (2000) contributed five features: 1) "*Net loan losses / Total assets less Total loans"*, 2) "*Net loan losses / Total loans"*, 3) "*Net loan losses plus Provision for loan losses / Net income"*, 4) "*Loans past due 90 days plus Nonaccrual loans / Total assets"* and, 5) "*Net income / Total assets".* This resulted in the identification of 700 hard features and 135 soft

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

features. Each of the hard and soft features have been added to a comparison table, see Table I [2][41][42][43].

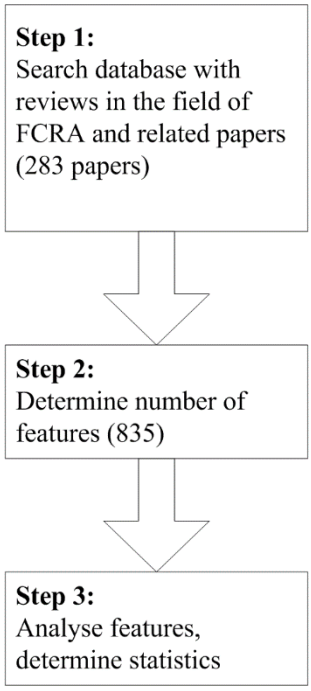


Figure 4. Feature Selection

Data analysis was conducted in one cycle of coding with the use of a priori coding scheme. The reason an a priori coding scheme was applied, is because the concepts that needed to be coded were known upfront, based on the previously defined FCRA Model.

To code the selected items, the following questions are asked: 1) *is the feature a hard or soft feature?* and 2) *is the feature a relational or transactional feature?* For example, the feature *"net income/total assets"* is a hard feature from a transactional perspective. A hard feature because the ratio can be calculated and transactional, because the figures can be derived from a system. An example of a hard / relational feature is *"the number of times the annual financial statements are deposited too late"*. A hard feature because the number can be calculated and relationship because it's a proxy of a soft feature, for example of management quality.

"The quality of management" is a soft feature from a relational perspective. A soft feature because it cannot be calculated directly and the (qualitative) information has to gathered via personal contacts.

TABLE I. SNAPSHOT COMPARISON TABLE

	Grunert (2005)	Altman (2010)	Cornee (2014)	Tronnberg (2014)
Management Quality	1		1	1
Country Court Decisions		1		
Audit Accounts		1		
Late Filling Days		1		
Sector				
Industry Risk				

V. RESULTS

In this section, the results of the data collection are presented. As described in the previous section, first features from existing studies have been collected, added to a comparison table and coded. Therefore, three separate results can be identified: 1) descriptive statistics for hard features, 2) descriptive statistics for soft features and, 3) the mapping of the hard and soft features to the FCRA Model.

A. Results from an information type perspective

As stated in this section, most features are based on data from the financial statements. Financial statements are, in most organizations, created once or twice a year. Therefore, the data needed to calculate the features, is available once or twice a year. This causes an information opacity problem thereby reducing the effectiveness of the features. Other organizations that also assess the financial credit risk of an organization are banks, credit assessors, etc. Both previously also had to trust numbers that are published once a year. Since this time period is too long for both parties, they searched for solutions to address this problem.

Camaco-Miñano et al. [44] show that sector, size, number of shareholdings, ROA, and liquidity can explain the bankruptcy process outcome and also predict the process for still-healthy firms. Three of five features exist of qualitative information.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A1. Descriptive statistics for hard features

The extraction of the features resulted in the registration of 700 features from 238 papers. From this sample, the top ten features were identified and selected; see Table II.

TABLE II. TOP TEN FEATURES

Feature 01: Net income/total assets	85 (papers)
Feature 02: current ratio	74
Feature 03: EBIT/total assets (*)	65
Feature 04: retained earnings/total assets (*)	62
Feature 05: working capital/total assets (*)	60
Feature 06: sales/total assets (*)	46
Feature 07: quick ratio	41
Feature 08: current assets/total assets	39
Feature 09: total debt/total assets	39
Feature 10: cash/total assets	32

Analysis of the hard features show that each of the top ten applied features are features that are applied in the financial statements of the organization. In addition, each feature lies a connection between the three main components of the financial statements namely: the cash flow statement, profit and loss statement, and the balance sheet. They do so by comparing the liquidity (features 02, 05, 07, 08, 09 and 10), the solvency (features 03 and 04) and the profitability (features 01, 03 and 06). Where the liquidity is primarily related to cash flow; the solvency is related to the balance sheet; the profitability is primarily related to the profit and loss statement. Of course, there is a main interrelationship between all these three main components of the financial statements.

Additionally, results show that our findings are in line with statements made in previous research, namely that the Altman model for bankruptcy prediction [28] is the most applied one. This is indicated by the fact that 4 features in the top 10 (indicated by an asterisk) are part of the Altman-Z score. And the fifth feature by Altman (Market Value of Equity/Total Liabilities) ranks thirteenth.

A2. Descriptive statistics for soft features

The extraction of the features resulted in the registration of 135 features from 20 papers. Likewise, to the hard features a top ten can be derived. However, in contrast to the hard features this top ten would exist out of features that are only mentioned four, three, or two times. One feature is listed four times, namely *"management quality"*. Four features are listed three times, namely: *"county court judgements"*, *"decision to check audited accounts"*, *"decision to issue cash flow statements"* and, *"late filing days"*. Followed by twelve features mentioned only twice. The remaining 118 features all are mentioned once. Therefore, creating a top ten didn't seem useful. In addition, the soft features have been additional coded to create a categorization (see Figure 5). The categorization has three main differentiations: 1) internal, 2) external and 3) social contract. 'Internal' is defined as qualitative (soft) information about the client; for example, about the client's management and its innovative power. 'External' can be seen as the environment that affects and interacts with the client. There are three main sub-categories: business (e.g., the number of visits with customer vendors & suppliers or visits with customers about business status), industry (e.g., the number of reviews of trade journals from customer's industries or the number of memberships in trade associations relating to customer's industries) and economy (e.g., the number of attendances at local chamber events or number of memberships in civic and community organizations). The last differentiation is 'social contract', which is defined as qualitative (soft) information about the lending relationship. 'Social contract' is further divided in two categories: quality of the credit relationship and value of the social contract.

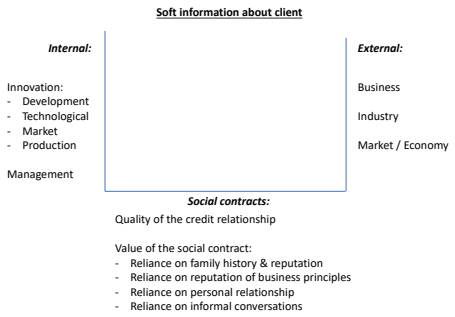


Figure 5. Categorization Soft Features

A3. Descriptive statistics for techniques

Based on the 128 papers [8] the frequency of the techniques applied to take the FCRA-decision have collected. In total 84 techniques have been identified. Seven techniques occur more than five times. Out of these seven techniques four occur more than 10 times, see Table III. The remaining 77 techniques occur up to 4, 3, 2 or 1 times.

TABLE III. TOP 7 TECHNIQUES

Technique	#
BPNN (Back propagation trained Neural Network)	28
DA (Discriminant Analysis)	18
LOGIT	18
LDA (Linear Discriminant Analysis)	10
Rough Set	08
GA (Genetic Algorithm)	05
Probit	05

A4. Descriptive Statistics for performance

Overall can be stated that research on feature identification does not clearly report on the (overall) performance of the features identified. Off the researched paper, only 16 report extensively on the performance of the applied techniques. We argue that further research should report on the performance of the identified and tested features. To measure the performance Chen [9] identified 17 measurements which can

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

be applied, for example: 1) accuracy, 2) root mean squared error, 3) true positive, and 4) true negative. For the detailed description of these 17 measurement we refer to Chen [45].

B. Results from a DMN perspective

Analyzing the top ten features from a DMN perspective shows four results. The first result: decision versus data input show that each feature is treated like a decision. The feature is derived from one or more conditions. For example, the first feature is derived out of two conditions: net income and total assets to which a mathematical formula is applied, in this specific case, net income divided by total assets. Each feature in the 10 retrieves the applied conditions from one data source, namely, the financial statements (the cashflow statement, the profit and loss account and/or the balance sheet).

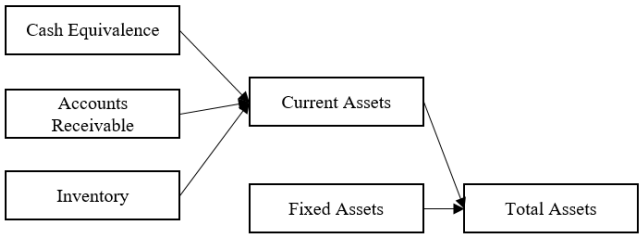


Figure 6. DRD-level Elements (Hard Features)

From the perspective of the financial statements, the conditions applied, e.g., net income, actually are data input since all are listed there. However, when analyzing one step deeper, each data input on the balance sheet or the profit and loss account is actually a decision. For example, total assets, is calculated as current assets plus fixed assets; see Figure 6. When analyzing all of the quantitative features selected, all features are derived from the cashflow statement, the profit and loss account and/or the balance sheet. A potential explanation of this phenomenon can be that the financial industry only looks at formal documents and formal statements. However, this raises the question if these combined features contain specific sub-decisions or specific input data elements that make them suitable for analysis. According to the researchers, this would be a subject to further investigate.

In addition, the features only apply information from the current financial statements. Formally, the cashflow statement, the profit and loss account and the balance sheet have to be created once a year. Most companies create this information more times a year, voluntarily or obligatory. Also, not comparing information from early years, thereby indicating that the patterns have no additional information value. By analyzing the deeper layers underneath the features described previously, the hypothesis is that a better and quicker FCRA can be performed.

C. Results from an information source perspective

The third perspective from which factors can be classified is the information source perspective. The concept *"the hardening of information"* states that because personal contact with the bank has decreased the banks rely more and more on hard quantitative information. However, if the model on which they base these conclusions is further dissected, two axes can be distinguished: A) the type of data and B) the manner in which the data is retrieved. The first axe describes the type of data that organizations retrieve to make a judgement about the financial credit risk. In the papers of Berger [21][25], the same distinction is made in an information type perspective: hard versus soft data. The second axe described the manner in which this information is retrieved. For example, two manners in which information can be collected are: 1) through face to face contact between a loan officers and the organization's owner and 2) through a form on a website or any other digital manner. Since more banks, credit organizations, and accountants rely on the second, the statement of "the hardening of information" is that only quantitative data is used. Thereby underlying the fact that the traditional features are the most useful features to analyze going concern assessment. The main reason they state to support their claim is the adoption rate of technology.

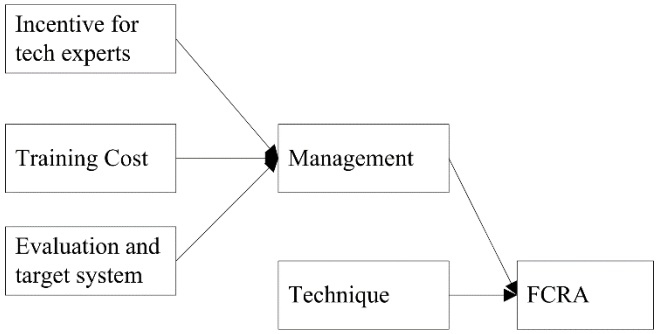


Figure 7. DRD-level Elements (Soft Features)

However, a counter claim can be made that through the adoption of technology soft information can be more easily collected. For example, through firehose access to social media websites. However, this will depend on the type of soft or hard information one wants to retrieve because not all soft information can be retrieved through social websites, some still might need to be retrieved face to face. Therefore, the bottom part of our model, see Figure 3, indicates the manner in which the information is retrieved.

D. Results from an organization perspective

In FCRA literature, from a banking perspective, a distinction is made between the manner in which small and big banks assess the risk. Small banks apply more of a relationship perspective to assess the risk while big banks apply the analysis of transactions to determine the risk. Although this specific distinction cannot be found in accountancy and lending (firms) literature, the hypothesis is that the same basic rules apply. Therefore, the right axe of the FCRA Model contains the size of the firms assessing the risk; see Figure 3.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

VI. Conclusion and Future Work

In this paper, we aimed at finding an answer to the following research question: *"how to categorize financial credit risk features such that an integrative relationship is established with the information type applied and information sources used?"* To accomplish this goal, we conducted a literature study to identify features that have been designed and applied in previous research followed by coding the features based on an a priori coding scheme. The literature resulted in a total of 258 selected papers. From the selected papers, a total of 835 features were selected. Based on the a priori coding scheme, the features were mapped according to the following dimensions: A) the type of features applied, B) the information source applied and, C) the type of organization that applies the features. The results show that most features focus on hard information from a transactional source from official information with a high latency. In addition, the results show that most features still relate to the traditional Altman-Z score.

All the results have been mapped on the FCRA Model, which is based on Wand and Weber [46], see Figure 3. The insights derived from this study provides a better understanding of the level on which the features are applied and where they score in the FCRA Model. This will enable further exploration and identification of features that have a low latency but still have a proper predictive power. From a practical perspective, our study provides an overview of features that can currently be applied, and which further exploration should be considered.

While we provide an integrative overview of features for FCRA, our study is not without limitations. The first limitation concerns the sampling and sample size. The sample group of features is drawn from the identified paper without considering the effectiveness of the features selected. The main reason for this choice is the fact that not all papers report on the effectiveness of the features applied. While we believe that for the purpose of this study this causes no problems, further refinement of the features selected is recommended. Additionally, our results should be further validated in practice.

References

[1] E. Mantelaers and M. Zoet, "A New Explorative Model to Assess the Financial Credit Risk Assessment," eKNOW 2018.

[2] S. Cornee and A. Szafarz, "Vive la Difference: Social Banks and Reciprocity in the Credit Market," *J. Bus. Ethics*, vol. 125, no. July, pp. 361–380, 2014.

[3] S. Balcaen and H. Ooghe, "35 years of studies on business failure: An overview of the classic statistical methodologies and their related problems," *British Accounting Review*, vol. 38, no. 1. pp. 63–93, 2006.

[4] J. Watson and J. E. Everett, "Do Small Businesses Have High Failure Rates?," *J. Small Bus. Manag.*, vol. 34, no. 4, pp. 45–62, 1996.

[5] B. S. Ahn, S. S. Cho, and C. Y. Kim, "The integrated methodology of rough set theory and artificial neural network for business failure prediction," *Expert Syst. Appl.*, vol. 18, no. 2, pp. 65–74, 2000.

[6] T. Derriks, "A Business Process & Rules Management Maturity Model for the Dutch governmental sector," no. March, 2012.

[7] C.-F. Tsai, "Feature selection in bankruptcy prediction," *Knowledge-Based Syst.*, vol. 22, no. 2, pp. 120–127, 2009.

[8] P. Ravi Kumar and V. Ravi, "Bankruptcy prediction in banks and firms via statistical and intelligent techniques - A review," *Eur. J. Oper. Res.*, vol. 180, no. 1, pp. 1–28, 2007.

[9] N. Chen, B. Ribeiro, and A. Chen, "Financial credit risk assessment: a recent review," *Artif. Intell. Rev.*, vol. 45, no. 1, pp. 1–23, 2016.

[10] A. N. Berger and L. K. Black, "Bank size, lending technologies, and small business finance," *J. Bank. Financ.*, 2011.

[11] A. N. Berger *et al.*, "Small Business Lending by Banks: Lending Technologies and the Effects of Banking Industry Consolidation and Technological Change."

[12] N. J. Kim and R. Z. Elias, "Financial statements, attestation level and lending decision by small banks," *Southwest Bus. Econ. J.*, pp. 63–72, 2008.

[13] L. Bollen, R. Meuwissen, G. M.- Faces, and undefined 2008, "Classifying major European business failures," *FST*.

[14] M. a Petersen, "Information: Hard and Soft," *Northwest. Univ. Doc. Trav.*, no. July, p. 20, 2004.

[15] M. Kano, H. Uchida, G. F. Udell, and W. Watanabe, "Information verifiability, bank organization, bank competition and bank-borrower relationships," *J. Bank. Financ.*, vol. 35, no. 4, pp. 935–954, 2011.

[16] W. W. T. Nemoto., Y. Ogura., "The Decision-Making Mechanism of Regional Financial Institutions and the Utilization of Soft Information *," *Policy Res. Institute, Minist. Financ. Japan*, vol. 9, no. January, pp. 87–116, 2013.

[17] A. N. Berger and G. F. Udell, "A More Complete Conceptual Framework for SME Finance."

[18] D. A. Carter, J. E. Mcnulty, and J. A. Verbrugge, "Do Small Banks have an Advantage in Lending? An Examination of Risk-Adjusted Yields on Business Loans at Large and Small Banks," *J. Financ. Serv. Res.*, vol. 25, pp. 2–3, 2004.

[19] H. Brown-Liburd and M. A. Vasarhelyi, "Big Data and Audit Evidence," *J. Emerg. Technol. Account.*, 2015.

[20] Y.-H. Lu, Y.-C. Lin, and Y.-L. Lin, "資料探勘技術在繼續經營疑慮意見診斷模型之應用 (Going-Concern Opinion: The Application of Data Mining Technologies)." 27-Jul-2016.

[21] E. I. Altman, G. Sabato, and N. Wilson, "The value of non-financial information in small and medium-sized enterprise risk management," *J. Credit Risk*, vol. 6, no. 2, pp. 1–30, 2010.

[22] F. Mccann and T. Mcindoe-Calder, "Firm size, credit scoring accuracy and banks' production of soft information," 2015.

[23] F. Dainelli, F. Giunta, and F. Cipollini, "Determinants of SME credit worthiness under Basel rules: the value of credit history information," *PSL Q. Rev.*, vol. 66, no. 264, pp. 21–47, 2013.

[24] J. P. Suter and I.). F. S. of B. Anderson University (Anderson, *A Quantitative Study of the Collection of Soft Information by Small Community Banks: Building Sustainable Competitive Advantage*. Anderson University

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Falls School of Business, 2007.

[25] S. Angilella and S. Mazzù, "The financing of innovative SMEs: A multicriteria credit rating model," *Eur. J. Oper. Res.*, vol. 244, no. 2, pp. 540–554, 2015.

[26] E. Kirkos, "Assessing methodologies for intelligent bankruptcy prediction," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 83–123, 2012.

[27] M. Aruldoss, M. L. Travis, and V. P. Venkatesan, "A reference model for business intelligence to predict bankruptcy," *J. Enterp. Inf. Manag.*, vol. 28, no. 2, pp. 186–217, 2015.

[28] E. I. Altman, "The Prediction of Corporate Bankruptcy: A Discriminant Analysis," *J. Finance*, vol. 23, no. 1, pp. 193–194, 1968.

[29] R. O. Edmister, "An empirical test of financial ratio analysis for small business failure prediction," *J. Financ. Quant. Anal.*, vol. 7, no. 2, pp. 1477–1493, 1972.

[30] E. B. Deakin, "A Discriminant Analysis of Predictors of Business Failure," *J. Account. Res.*, vol. 10, no. 1, pp. 167–179, 1972.

[31] Y. Chen, R. J. Huang, J. Tsai, and L. Y. Tzeng, "Soft Information and Small Business Lending," *J. Financ. Serv. Res.*, vol. 47, no. 1, pp. 115–133, 2013.

[32] Y. H. Ju and S. Y. Sohn, "Updating a credit-scoring model based on new attributes without realization of actual data," *Eur. J. Oper. Res.*, vol. 234, no. 1, pp. 119–126, 2014.

[33] K. Kosmidis and A. Stavropoulos, "Corporate failure diagnosis in SMEs," *Int. J. Account. Inf. Manag.*, vol. 22, no. 1, pp. 49–67, 2014.

[34] G. F. Udell, "What's in a relationship? The case of commercial lending," *Bus. Horiz.*, vol. 51, no. 2, pp. 93–103, 2008.

[35] S. Bhattacharya and A. V. Thakor, "Contemporary banking theory," *J. Financ. Intermediation*, vol. 3, no. 1, pp. 2–50, 1993.

[36] A. C. Edmondson and S. E. Mcmanus, "Methodological fit in management field research," *Acad. Manag. Rev.*, vol. 32, no. 4, pp. 1155–1179, 2007.

[37] M. Spreen, "Rare Populations, Hidden Populations, and Link-Tracing Designs: What and Why?," *Bull. Méthodologie Sociol.*, vol. 36, no. 1, pp. 34–58, 1992.

[38] S. K. Thompson, "Adaptive sampling in behavioral surveys," *NIDA Res.Monogr*, vol. 167, no. 1046–9516 (Linking), pp. 296–319, 1997.

[39] K. Hjelm, P. Nyberg, Å. Isacsson, and J. Apelqvist, "Beliefs about health and illness essential for self-care practice: A comparison of migrant Yugoslavian and Swedish diabetic females," *J. Adv. Nurs.*, vol. 30, no. 5, pp. 1147–1159, 1999.

[40] J. H. Patrick, R. A. Pruchno, and M. S. Rose, "Recruiting Research Participants: A Comparison of the Costs and Effectiveness of Five Recruitment Strategies," *Gerontologist*, vol. 38, no. 3, pp. 295–302, 1998.

[41] J. Grunert, L. Norden, and M. Weber, "The role of non-financial factors in internal credit ratings," *J. Bank. Financ.*, vol. 29, no. 2, pp. 509–531, 2005.

[42] E. I. Altman, G. Sabato, and N. Wilson, "The value of non-financial information in small and medium-sized enterprise risk management," *J. Credit Risk*, vol. 6, no. 2, pp. 1–33, 2010.

[43] C.-C. Trönnberg and S. Hemlin, "Lending decision making in banks: A critical incident study of loan officers," *Eur. Manag. J.*, vol. 32, pp. 362–372, 2014.

[44] M. del M. Camacho-Miñano, M. J. Segovia-Vargas, and D. Pascual-Ezama, "Which Characteristics Predict the Survival of Insolvent Firms? An SME Reorganization Prediction Model," *J. Small Bus. Manag.*, vol. 53, no. 2, pp. 340–354, 2015.

[45] N. Chen, B. Ribeiro, and A. Chen, "Financial credit risk assessment: a recent review," *Artif. Intell. Rev.*, 2016.

[46] Y. Wand and R. Weber, "An Ontological Model of an Information System," *IEEE Trans. Softw. Eng.*, vol. 16, no. 11, 1990.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

On the Variability Dimensions of Normalized Systems Applications: Experiences from Four Case Studies

Peter De Bruyn, Herwig Mannaert and Philip Huysmans

Department of Management Information Systems
Faculty of Applied Economics
University of Antwerp, Belgium
Email: {peter.debruyn,herwig.mannaert,philip.huysmans}@uantwerp.be

***Abstract*—Normalized Systems Theory aims to create software systems exhibiting a proven degree of evolvability. While its theorems have been formally proven and several applications have been used in practice, no real overview of the typical types or dimensions along which such Normalized Systems software applications can evolve is present. Therefore, this paper presents several cases in which its different variability dimensions are illustrated. Based on these cases, a more general overview of four variability dimensions for Normalized Systems software applications is proposed: changes regarding the application model, expanders, craftings and technological options.**

Keywords–Evolvability; Normalized Systems; Variability dimensions; Case Study

I. INTRODUCTION

This paper extends a previous paper which was originally presented at the EMPAT track on evolvable modularity patterns at the PATTERNS conference 2018 [1].

The evolvability of information systems (IS) is considered as an important attribute determining the survival chances of organizations, although it has not yet received much attention within the IS research area [2]. Normalized Systems Theory (NST) was proposed as one theory to provide an ex-ante proven approach to build evolvable software by leveraging concepts from systems theory and statistical thermodynamics [3]–[5]. The theory prescribes a set of theorems which are necessary conditions to obtain evolvable software and proposes a set of patterns to generate significant parts of software systems which can obey to these theorems. While it has been suggested that software created in this way exhibits evolvability, the main dimensions of evolvability or variability facilitated by the theory have nevertheless not yet been thoroughly discussed. Additionally, while some NST cases have been documented in extant literature [6]–[10], the overall number of cases is still fairly limited and their analysis has never been focused on the different dimensions of evolvability which were possibly present. This paper attempts to tackle both mentioned gaps by first discussing the case of a new (i.e., not previously documented) NST software application, which was built and used for the management of process evaluations of master dissertations at the faculty of the authors. Based on our experiences with this case, we will document variations that occurred along several dimensions: the model (business entities) of the application, the craftings (customizations on top of the generated code), the technology used and the version of the code generators themselves. Next to this new case, we will reinterpret several previous cases which were documented earlier in other work [7]–[10]. So while the cases themselves are not new, our perspective and way of analyzing the cases is. It is also the discussion of these additional (previously reported) cases in this evolvability dimensions context which is the main addition of this paper when compared to our initial contribution at PATTERNS 2018 [1]. In this way, we are able to identify and discuss the different dimensions along which variations in an NST application can arise and provide illustrations for each of them from different cases and examples. Consequently, these dimensions are also important indications with respect to the main areas in which an NST application can evolve throughout time.

The remainder of this paper is structured as follows. In Section II, we briefly present NST as the theoretical basis on which the considered software applications were built. Section III provides some general context regarding the newly reported educational case as well as its analysis in terms of evolvability dimensions. Section IV focuses on some of the earlier presented cases but analyzes them from a different angle than before, i.e., also in terms of evolvability dimensions. We offer a discussion in Section V and our conclusion in Section VI.

II. NORMALIZED SYSTEMS THEORY

The case applications we will present and analyze in the following sections, are based on NST. This theory has been previously formulated with the aim of creating software applications exhibiting a proven amount of evolvability [3]–[5]. More specifically, the goal is to eliminate the generally experienced phenomenon in which software systems become more difficult to maintain and adapt as they become bigger and evolve throughout time [11].

NST is theoretically founded on the concept of *stability* from systems theory. Here, stability is considered as an essential property of systems. Stability means that a bounded input should result in a bounded output, even if an unlimited time period is considered. In the context of information systems, this implies that a bounded set of changes should only result in a bounded impact to the information system, even in cases where an unlimited time period and growth of the system is taken into account (i.e., considering an unlimited systems evolution). Put differently, it is demanded that the impact of changes to an information system should not be dependent on the size of the system to which they are applied, but only on the size and property of the changes to be performed. Changes dependent on the size of the system are called *combinatorial*

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

effects. It has been formally proven that any violation of any of the following *theorems* will result in combinatorial effects (thereby hampering evolvability) [3]–[5]:

- *Separation of Concerns*, stating that each concern (i.e., each change driver) needs to be separated from other concerns in its own construct;
- *Action Version Transparency*, stating that an action entity should be able to be updated without impacting the action entities it is called by;
- *Data Version Transparency*, stating that a data entity should be updateable without impacting the action entities it is called by;
- *Separation of States*, stating that all actions in a workflow should be separated by state (i.e., being called in a stateful way).

The application of the theorems in practice has shown to result in very fine-grained modular structures within a software application. Such structure are, in general, difficult to achieve by manual programming. Therefore, NST proposes five *elements* (action, data, workflow, connector and trigger) that serve as design patterns [4], [5]:

- *data element*: a set of software constructs encapsulating a data construct (including a set of convenience methods, such as get- and set-methods, and providing remote access and persistence), allowing data storage and usage within an NST application;
- *action element*: a set of software constructs encapsulating an action construct (providing remote access, logging and access control), allowing the execution of (units of) processing functionality within an NST application;
- *workflow element*: a set of software constructs allowing the execution of a sequence of action elements (on a specific data element) within an NST application;
- *connector element*: a set of software constructs enabling the interaction of an NST application with external systems and users in a stateful way;
- *trigger element*: a set of software constructs enabling the triggering of action elements within an NST application, based on error and non-error states.

Based on these elements, NST software is generated in a relatively straightforward way through the use of the *NST expansion mechanism*. First, a model of the considered universe of discussion is defined in terms of a set of data, action and workflow elements. Next, NST expanders generate parameterized copies of the general element design patterns into boiler plate source code. Several layers can be discerned in this code: a shared layer (not containing any reference to external technologies), data layer (taking care of data services), logic layer (taking care of business logic and transactions), remote or proxy layer (taking care of remote access), control layer (taking care of the routing of incoming requests to the appropriate method in the appropriate class in the proxy layer) and view layer (taking care of presenting the view to be rendered by the user interface, such as a web browser). This generated code can, if preferred, be complemented with *craftings* (custom code) to add non-standard functionality that is not provided by the expanders themselves at well specified places (anchors) within the boiler plate code. The boiler plate code together with the optional craftings are then compiled (built) so that the application can be deployed.

III. Educational case

In this section, we will first introduce the educational case in-depth. Next, we analyze the case both in general and along several potential evolvability dimensions.

A. Case introduction

The new case we present in this paper is situated within an educational context and concerned with the master thesis evaluations at the Faculty of Applied Economics of the University of Antwerp. At the university, master students writing their dissertation are not only evaluated with regard to the end result (i.e., the thesis itself) but also (for a minor part) with regard to the process they go through in order to arrive at that end result (e.g., their communication and reporting skills, problem-solving attitude, etcetera during the project). This "*process evaluation*" is built around a set of specific evaluation criteria for students of this faculty, based upon the pedagogic vision of the faculty. More specifically, depending on the trajectory a student is following, the thesis advisor(s) need(s) to assess a student two or three times on four skill dimensions (each comprising a set of specific skills to be rated from insufficient up to very good) during the completion of his or her master thesis.

In this context, the *procesEval* application, based on NST, was created around 2013. Up to that moment, the process evaluation was either performed on paper or had to be registered via a customized part of the university's online learning and course management system. While the paper based evaluation was considered as generating administrative overhead (the results had to be manually copied into the university's database systems by the administration) and providing little overview for the thesis advisors (e.g., when performing the second process evaluation they could not easily consult the first process evaluation in order to make a more objective comparison), the electronic variant in the online learning and course management system was considered cumbersome from a usability perspective (e.g., users complaining about the amount of clicks required to perform "simple" actions or experiencing difficulties in order to find the information they are looking for).

The faculty management decided to develop an NST application to manage the process evaluations. This choice was made for several reasons. First, the expertise on how to build NST applications was present within the faculty itself as the theory (and the adjoining code expanders) was the output of research projects of faculty members. Second, as the software system would be developed by members of the faculty itself as well, the developers were highly knowledgeable about the inner working of the faculty (administration) and the associated (functional) requirements. And third, evolvability and maintainability were considered to be import quality aspects of the software system to be developed as the process evaluation was anticipated to remain an important part of the student evaluations for several years to come (but could be subject to some further fine-tuning or redirection in the future). Given the situation of the project as sketched above, it was

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

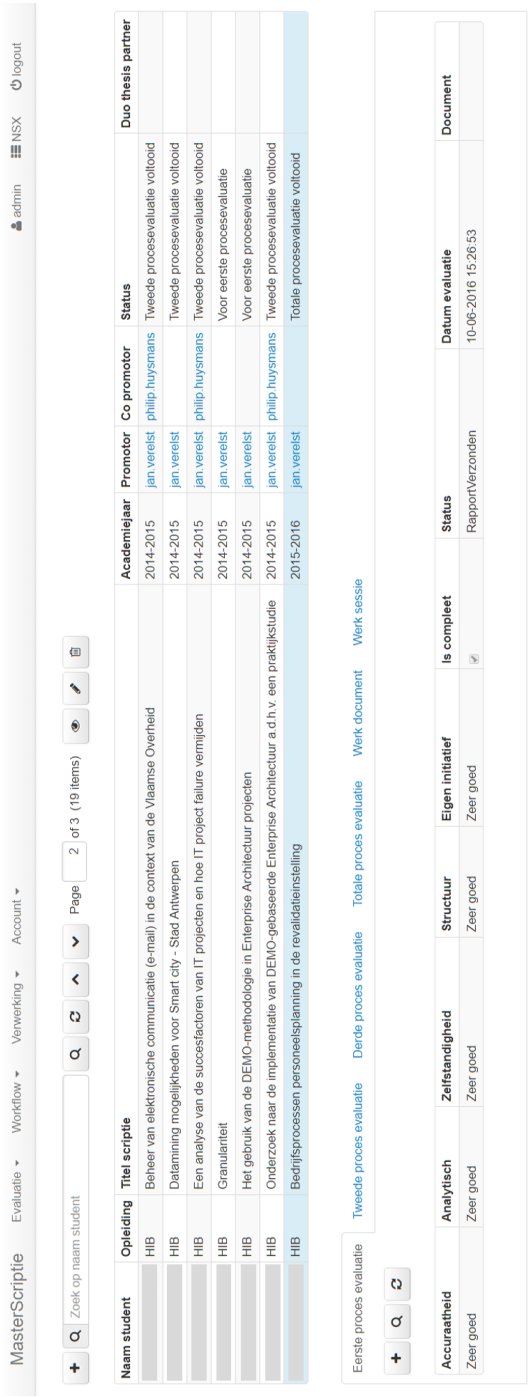


Figure 1. A general screenshot of the procesEval application.

expected that the application could be developed in a rather short development trajectory without too many hurdles (i.e., no significant risk related to the technology was present and the application domain was well known and understood).

The application was developed in the beginning of 2013. In the academic years 2013–2014 and 2014–2015, a first pilot test with a set of key users (technological savvy and proactive faculty members) was conducted. In the academic years 2015–2016 and 2016–2017, the set of test users was gradually enlarged up to the level at which all thesis supervisors could use the procesEval application if they wanted, but could still use the paper version if preferred. As of the academic year 2017–2018, all faculty members were expected to use the NST procesEval application for the administration of the master thesis process evaluations. Apart from minor (usability) adjustments, the project has been completed without major problems. Currently, on a yearly basis, about 45 faculty members manage the process evaluation of roughly 500 students via the procesEval application.

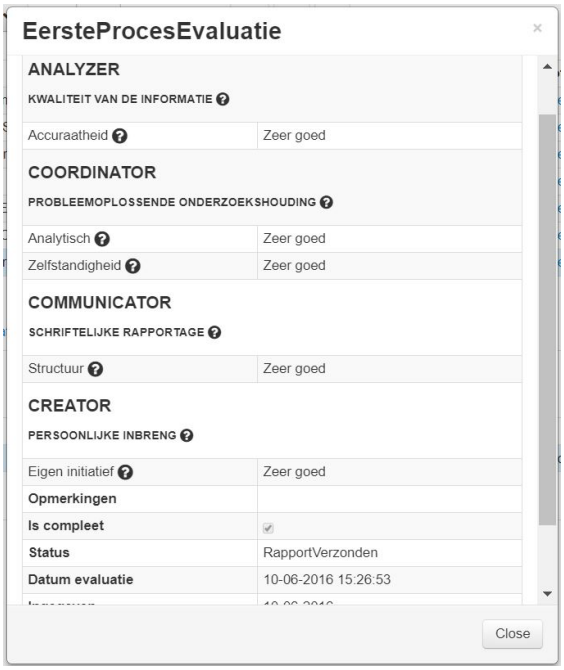


Figure 2. A screenshot of a specific process evaluation within the procesEval application.

In Figure 1, a screenshot of the procesEval application is shown (the names of the students are blurred out to assure anonymity, the names of the labels are Dutch as this is the administrative language of the organization). Here, one can notice that a supervisor can get an overview of all the students he or she is supervising in the current academic year. By selecting a particular student, a set of tabs appears below the first table providing further details regarding his/her (earlier) evaluations or working sessions (e.g., meetings) and documents (e.g., preliminary thesis version). Figure 2 shows a screenshot of one particular process evaluation. The procesEval application therefore manages all process evaluations (typically 2-3) of all master dissertations (as of 2017–2018) of multiple academic years. Based on the provided information, the application automatically generates overview reports of the evaluations

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and sends emails to students and supervisors with information regarding their evaluations, as well as reminders (e.g., when a particular process evaluation is due).

B. Case analysis

We will first provide a general overview of our case analysis, and then zoom into a set of relevant variability dimensions that could be discerned at the level of the case.

1) General overview: An NST application typically consists of a set of base components (which are reused in several or even most applications), as well as one or multiple non-base components (typically specific for the considered application). The base components used within the procesEval application consisted of 29 data elements, 7 task elements and 1 flow element. The non-base component used within the procesEval application consisted of 14 data elements, 8 task elements and 4 flow elements. As a consequence, relatively speaking, the NST application was still rather small: it comprised about 63 NST elements.

2) Model variations: By using the NST approach, the procesEval application could be extended and adapted at the level of the model (i.e., the definition of the different element instances for the considered application domain). For instance, additional elements could be added: next to the registration of three possible process evaluations for each student, some working documents and information regarding working sessions (e.g., what was agreed upon by the student and his supervisor during a meeting) could be added to the model. After re-generating the application based on this updated model, this functionality becomes available in the new version of the application. Similarly, existing (i.e., earlier created) components could be added to the model. For example, a notification component was added to the procesEval application as that component contained the functionality to automatically trigger emails and could be leveraged to enable the automatic report delivery (of the process evaluations to the students, supervisors and administration). While the model could be changed in terms of data elements and components, this also holds for all types of other possible changes within the model. More specifically, the following types of adaptations can be performed to create different variations of the application:

- the addition, update or deletion of a component (i.e., a set of data, task and flow elements);
- the addition, update or deletion of a data element definition (its fields with its types and field options, finders, data element options, child elements);
- the addition or deletion of a task element definition (the specific implementation of a task is a crafting, see below);
- the addition, update or deletion of a flow element definition and its accompanying default state transitions.

It should be remarked that the determination and evolutions of such model is completely technology-agnostic (i.e., it does not require any specification in programming language specific terminology). For instance, the specification of the model (in terms of elements and their properties) is currently stored in an XML file, not containing any references to the (background) technology of the current reference implementation (i.e., Java). Based on this model, boiler plate source code for each of the layers can be created.

3) Crafting variations: Once the model is converted (expanded) into boiler plate source code, additional code (so-called "craftings", which are custom made for an application) could be added between predefined anchors (insertions) or in additional classes (extensions). This way, non-standard functionality can be incorporated within the application as well. In total, the procesEval application contained 22 classes with insertions and 29 additional classes (extension). For instance, specific coding had to be added to make sure that a supervisor logged into the application can only view those master dissertations which he/she is supervising in the concerning year (i.e., dissertations supported by other supervisors or those of the previous year should not be visible). For this purpose, a few lines of code were added in the MasterThesisFinderBean class determining the fetching of the results viewable for a particular user. These FinderBean classes are expanded as part of the data layer: enforcing the filter of master dissertations at the level of the data layer ensures that no data from other users can be retrieved by the currently logged in user. Consequently, this crafting only impacts the data layer, while the remaining layers have no impact resulting from this change: they perform their functionality handling the (filtered) data offered by the MasterThesisFinderBean.

Additionally, a set of screentips was added to assist the user when filling-in the process evaluation (e.g., summarizing the meaning of each of the evaluation criteria in case of a mouse-over). The expanded NST code base supports this functionality by providing a helpInfo Knockout binding. Specific screentips can be added by including a crafting using this Knockout binding, and referring to a certain key. At run-time, the specific values for the required keys can be added in instances of HelpInfo data elements. This enables the configuration of the screentips even when the application has already been deployed. Note that only the view layer is customized for this functionality. This makes sense, since it is purely a useability concern, not impacting actual business logic. However, it is dependent on the specific technology used in the view layer (i.e., Knockout), and should be reprogrammed when a different technology is used.

Next, as mentioned before, the procesEval application also needed to create and send reports summarizing the content of the process evaluations. The definition of these reports (i.e., the items to be included and the corresponding layout) is considered to be a separate functionality, and should therefore be contained in a task element. The expanders provide all boilerplate code needed to execute this task in the NST application, and only the specific report generating functionality needs to be added as a crafting. The actual implementation of the execution of a task element is clearly separated, allowing versions and variations of the task implementation to co-exist. Currently, reports are generated using Jasper Reports. This requires the addition of a Jasper template file to the code base, and some code to fill the parameters to be inserted into this template. The additional processing logic is completely contained in the logic layer.

These craftings were added in a gradual and iterative way to the application: each time a particular additional functionality was added or improved, a new version of the overall application could be built and deployed. Furthermore, it can be remarked that each of these craftings were situated at another layer (i.e., data, view and logic).

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

4) Infrastructural technology variations: The procesEval application could be generated by using various different underlying infrastructural technologies. For instance, whereas a prototype of the application is typically demonstrated by using an HSQL database, most production systems are deployed while using a PostgreSQL database. Nevertheless, one can choose for SQLServer and MySQL databases as well. Further, the procesEval can be built by using different build automation frameworks (i.e., Ant and Maven). And finally, the procesEval could also be generated by using different controlling (Cocoon, Struts2, or combination Struts2-Knockout) and styling frameworks (plain style or using Bootstrap). In practice, the Struts2-Knockout and Bootstrap were used in the production environment. Changing the choice of a particular infrastructural technology in the procesEval only impacts those layers depending on the purpose of the technology (e.g., the database selection impacts the data layer, whereas the GUI framework selection impacts the view layer).

5) Expander version variations: The expanders (i.e., the programming logic used to convert the model into boiler plate source code according to the infrastructural technologies chosen) evolves throughout time as well. This way, when considering the current procesEval project duration (2013–present), 8 different production versions were deployed while using the same model and craftings (as the expanders provide backwards version compatibility). In each of these production versions, the new or improved possibilities of the expanders could be used. For instance, in one particular version of the expanders, information regarding a Date field did no longer have to be entered manually but could be selected by using a more advanced date picker. And, more relevant in the context of the procesEval, another particular version of the expanders allowed the automatic creation of summarizing graphs on certain fields. For example, it would now be possible to inspect the number of master dissertations who did not yet receive a first process evaluation versus those who did in a visual way. In order to use the date picker, no changes in the model or the craftings are required. In order to use the status graphs, only one additional specification in the model (i.e., an option indicating that a graph for a particular field should be created) needs to be added. Clearly, the precise set of layers that is impacted due to an expander update depends on the type of modifications performed in that particular version update (logic related, view related, etcetera).

IV. Reinterpretation of existing cases

In this section, we will look at some previously documented and analyzed cases of NST applications. While the initial publications were not specifically directed towards the illustration of the variability dimensions present within these systems, we will now aim to see to which extent we can find indications of such variability dimensions in them. This should allow us to verify whether the variability dimensions identified in the educational case study above also appear in other cases, thereby increasing the validity of our study.

A. Budget management application

A first case which we published in earlier work concerns a budgeting application for a local Belgian government [7]. The administration of the concerned local government organization was required to track its allocated budgets meticulously and in a very fine-grained way (including the division of budgets in subbudgets, their reservation, changes to the budgets, etcetera). The goal of the application was to provide the functionality for users to have a clear overview and tracking of budgets and subbudgets, budget assignments, changes on them, and so on. While the organization originally had the possibility to perform these activities via Microsoft Excel by using pivot tables, the long term goal was to integrate the application performing these analyses with other functionalities such as project management, budget reporting and simulations. During the analysis of this case, we noticed that the development of this replacing application was not trivial [7]. Indeed, many people are used of working with the popular spreadsheet program Excel, which offers many flexible and versatile analysis options in a user friendly way. In order to be able to meet the high standards of the end users, it was therefore decided to approach the application development in a very iterative and gradual way. In a first stage, attention was almost exclusively devoted to the development of the functionalities related to budget management and the usability of that part for end users. It was only later on that the project started to focus on the realization of a larger application which also incorporated some of the additional functionalities as mentioned above. Therefore, both the optimization of the universe of discussion within the budgeting functionality, as well as the initial (exclusively budgeting oriented) and later phases (focusing on the other functionalities as well) can be seen as different versions of the model throughout time (each time be expanded into working prototypes or working applications). Therefore, this case clearly illustrated the relevance of *model variations*.

During the case, craftings had to be added at various places as well. First, some code was required to provide additional graphical features. That is, a more advanced user interface with more sophisticated screens was needed (compared to those that were by default provided by the code expanders at that point in time). Such more advanced (composed) screens would allow users to inspect budget specifications over various levels concurrently (year, department, article) or from different angles/perspectives (departments, types of activities), thereby replicating behavior somewhat similar as the previously used pivot tables. Next to that, several customizations were present for specific calculations (logical operations). These calculations were very context specific for the organization and domain at hand and were directed towards issues such as the on-the-fly calculation of the currently available budget based on all previous budgets, the verification that budget calls were not exceeding the available budget, etcetera. As these craftings were refined over time, they illustrate the relevance of the *crafting variations* within this case. However, probably even more interesting, while the logic related craftings were very specific for this application that needed to be developed, the graphical extensions (i.e., the composite screens displaying multiple data elements having a one-to-many relationship on one screen) were considered to be useful for other (current and future) NST applications. Stated otherwise, these craftings were regarded as being somewhat generic. As a consequence, over time, some of these graphical extensions have been included in the code expanders. As soon as this happened, the more advanced screens became available for other (already existing or newly developed) NST applications. Therefore, this case clearly illustrated the relevance of *expander variations*.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

B. Infrastructure monitoring application

A second case which has been published earlier involved an application for an organization providing hardware and software for the monitoring of infrastructure (e.g., checking the correct functioning) such as power supplies, airconditioning and so on [8]. Whereas the report of the previous case (discussed in Section IV-A) provided a general overview of the application as a whole, the current case was reported in a temporal way, i.e., four phases were discussed in which the application evolved from its original status to its current status (at the time of publication). Additionally, the case was somewhat atypical for an NST application as it involved one of the first applications (re)developed according to this approach. In particular, the following four phases were distinguished in the concerning case:

- *phase 1*: Initially, the application was designed in a rather monolithic way without explicit attention to modularization while using a Microsoft Access database and a Visual Basic application. This was a version of the application without any use of the NST approach;
- *phase 2*: The application was redeveloped and designed in another technology stack using Java 2 Enterprise Edition (J2EE) with Enterprise JavaBeans 2.1 and the Cocoon framework. With NST not yet formulated and the element expansion mechanism not yet developed, the software system was mainly developed manually but taking into account industry best practices and the (implicit) heuristic knowledge which would later on result into the NST theorems. This resulted in a recurrent structure throughout the application similar to the later on developed NST elements;
- *phase 3*: In the following version, a significant part of the application was defined by using descriptor files describing certain recurring constructs (such as the need to persist a certain type of data) for which the code was then generated by one of the first versions of the pattern expanders. Next to updating the application to a new version of its code base, some additional functionalities (e.g., regarding FAQs and asset management) were added by generating additional data elements for them. Some custom code (e.g., for authorization requirements and user interfaces) was added in separate files;
- *phase 4*: In a final documented phase of the application, a switch to other controls and protocols for the infrastructure was made. At the same time, a newer version of the NST expanders was used in which custom code could be added between specifically located anchors within the code that could be harvested and injected during regeneration later on.

Based on the description of the phases as provided above, we can not only distinguish the relevance of the evolvability dimensions described earlier in the context of the instructure monitoring application, but also gather some information on their historical occurrence throughout time. As in phase 1, a non-NST approach was adopted, no explicit evolvability dimensions were present. The case report mentioned difficulties in order to adapt the application, including duplications and lack of flexibility. In phases 2 and 3, the patterns or elements were introduced (first in a somewhat implicit way, later on in a more explicit way allowing for automatic code generation) and improved for (largely) the same functional requirements. Therefore, the *expander* variability dimension was introduced at this point. In phase 3, due to the addition of some functionality based on the same patterns, the *model* variability dimension was illustrated as well. Finally, in phase 4, the harvesting mechanism allowed the easier migration of custom code from one application version to another. Therefore, this last phase also illustrates the occurrence of the *crafting* variability dimension in this context. One might remark that the last variability dimension, regarding the technological or *infrastructural options* does not seem to covered by the case at hand. While it is true that during the actual use of the NST expanders no significant technological or infrastructural changes have been performed, the transition of phase 1 to phase 2 was partly motivated by the fact that the technologies used for the creation of the initial software application were not adequate to work with in a distributed and multi-user environment. Therefore, at least the relevance of this variability dimension could certainly be argued for in the context of this case as well.

C. Integration applications

Finally, a set of four enterprise application integration cases was presented in [8]. For the purpose of this paper, with our focus on variability dimensions, one case is particularly interesting as it illustrates the infrastructural technology variations possible within this context and this variability dimension was somewhat less prominently present in the cases discussed in Sections IV-A and IV-B. The case was conducted within a multinational human resources consulting firm for which web-based access to 180 data entities needed to be provided from a legacy application (which was using a PL/SQL Oracle database). At the time the case was carried out, the NST expanders did not provide support for the PL/SQL Oracle database as required by the case organization. Therefore, the NST expanders had to be adapted for this possibility. Once this operation was performed, the generated NST application was able to connect with the database of the case organization. Next to that, all previously existing default functionality typically present in an NST application (and not impacted by the required changes in the logic and data layers) were available as well (e.g., the out-of-the-box CRUD screens for all the data elements within the application). And, as this project required an adaptation of the expanders, the possibility to link to PL/SQL databases became as of then available for all other current or future NST applications. Therefore, this case clearly illustrated the relevance of *infrastructural technology variations*.

V. Discussion

Based on the above discussion of NST and its cases, we will discuss two broad areas in this section. First, in Section V-A, we will analyze the offered variability dimensions in the cases in a somewhat more general way. While we illustrated the possible variability dimensions using only one in-depth and three smaller cases, we anticipate that the proposed categorization can be generalized to a large extent as it also aligns with the general "degrees of freedom" available during

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

the development and maintenance of an NST application. Next, in Section V-B, we discuss some general implications that the existence of these kind of evolvability dimensions has for the management of NST projects and the role of the analyst in particular.

A. Application level evolvability dimensions

Based on our analysis as presented above, we identify four variability dimensions, as visualized in Figure 3.

First, as represented at the top of the figure, the modeler should select the *model* he or she wants to expand. Such a model is technology agnostic (i.e., defined without any reference to a particular technology that should be used) and represented by a blue puzzle (i.e., each puzzle piece represents a defined element, with the columns corresponding to data, task, flow, trigger and connector elements). Such a model can have multiple versions throughout time (e.g., being updated or complemented) or concurrently (e.g., choosing between a more extensive or summarized version). As a consequence, the figure contains multiple blue puzzles that are put behind each other and the chosen model represents a variability dimension (represented by the green bidirectional arrow).

Second, the *expanders* (represented by the trapezoid in the figure) generate (boiler plate) source code by taking the specifications in the chosen model as its *arguments*. For instance, for a data element Person, a set of java classes PersonBean, PersonLocal, PersonRemote, PersonDetails, etcetera will be generated. This code can be called boiler plate code as it provides a set of standard functionalities for each of the elements within the model. Nevertheless, one could argue that this set of standard functionalities is already quite decent as it contains the possibilities to provide standard finders, master-detail (waterfall) screens, certain display options, document upload/download functionality, child relations, etcetera. The expanders themselves evolve throughout time. Typically, in each new version, a set of bugs of the previous version are solved and additional features (e.g., creation of a status graph) are provided. It should be remarked that, given the fact that the application model is completely technology agnostic and can be used as argument for any version of the expanders, these bug fixes and additional features become available for all versions of all application models (only a re-expansion or "rejuvenation" is required). As a consequence, the figure contains multiple trapezoids that are put behind each other and the expander version represents a variability dimension (represented by the green bidirectional arrow).

Third, in the middle left of the figure, a set of *infrastructural options* are displayed by means of different rectangular blocks. These consist of global options (e.g., determining the build automation framework), presentation settings (determining the graphical user framework), business logic settings (determining the database used) and technical infrastructure (e.g., determining the background technology). For each of these infrastructural options, the modeler can choose out of a set of possibilities (e.g., different user interface frameworks for which the associated code can be generated), which will be used by the expanders as their *parameters*. That is, given a chosen application model version and expander version, different variants of boiler plate code can be generated, depending on the choices regarding the infrastructural options. As a consequence, the figure contains multiple infrastructural option sets (blocks) that are put behind each other and the infrastructural options represent a variability dimension (represented by the green bidirectional arrow).

Fourth, *craftings* ("custom code") can be applied to the generated source code. These craftings are represented in the lower left of the figure by means of red clouds as they enrich (are put upon) the earlier generated boiler plate code and can be harvested into a separate repository before regenerating the software application (after which they can be applied again). This includes extensions (e.g., additional classes added to the generated code base) as well as insertions (i.e., additional lines of code added between the foreseen anchors within the code). Craftings can have multiple versions throughout time (e.g., being updated or complemented) or concurrently (e.g., choosing between a more advanced or simplified version). These craftings should contain as little technology specific statements within their source code as possible (apart from the chosen background technology). Indeed, craftings referring to (for instance) a specific GUI framework will only be reusable as long as this particular GUI framework is selected during the generation of the application. In contrast, craftings performing certain validations but not containing any EJB specific statements will be able to be reused when applying other versions or choices regarding such framework. Craftings not dependent on the technology framework of a specific layer can be included in the "common" directory structure, whereas technology-dependent craftings need to reside in the directory structure specified for that technology (e.g., EJB for the logic layer, JPA for the data layer, Struts2 for the control layer). As a consequence, the figure contains multiple crafting planes that are put behind each other and the chosen set of craftings represents a variability dimension (represented by the green bidirectional arrow).

In summary, each part in Figure 3 with green bidirectional arrows is a variability dimension in an NST context. It is clear that talking about *the* "version" of an NST application (as is traditionally done for software systems) in such context becomes rather pointless. Indeed, the eventual software application (the grey puzzle at the bottom of the figure) is the result of a specific version of an application model, expander version, infrastructural options and set of craftings. Put differently, with M, E, I and C referring to the number of available application model versions, the number of expander versions, the number of infrastructural option combinations and crafting sets respectively, the total set of possible versions V of a particular NST application becomes equal to:

$$V = M \times E \times I \times C$$

Whereas the specific values of M and C are different for every single application, the values of E and I are dependent on the current state of the expanders. Remark that the number of infrastructural option combinations (I) is equally a product:

$$I = G \times P \times B \times T$$

Where G represents the number of available global option settings, P the number of available presentation settings, B the number of available business logic settings and T the number of available technical infrastructure settings. This general idea in terms of combinatorics corresponds to the overall goal of NST: enabling evolvability and variability by *leveraging the law of exponential variation gains* by means

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

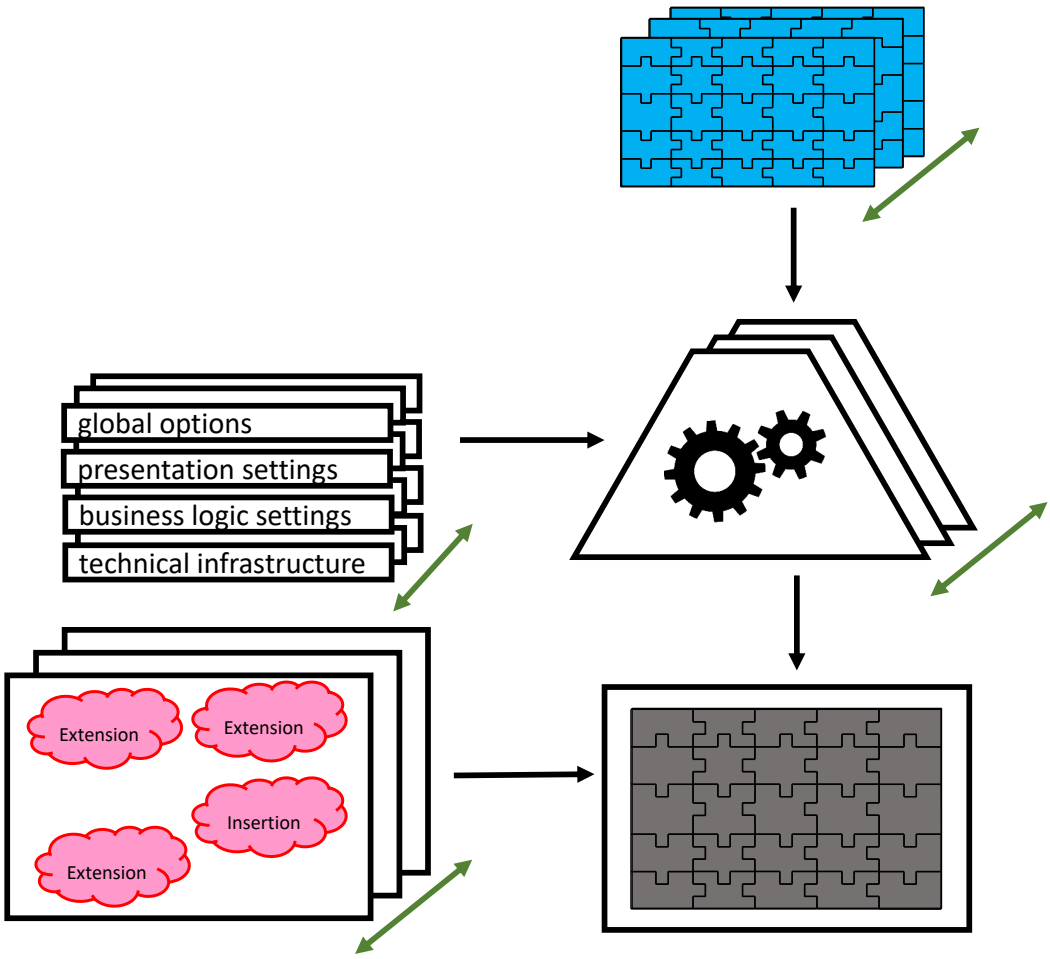


Figure 3. A graphical representation of four variability dimensions within a Normalized Systems application.

of the thorough decoupling of concerns and the facilitation of their recombination potential [5].

B. *Project management*

Performing software development projects in an environment where the chosen approach enables the variability dimensions as discussed in this paper, has some implications on how the project management in such context can be executed and, in particular, on the responsibilities of the analyst.

First, as NST is based upon the idea of realizing evolvable and adaptable applications, it seems logical that most NST projects are conducted in an iterative or agile way. More specifically, due to a visual modeling tool and supporting web application —allowing the definition of an NST model (specified in terms of data, action, flow, trigger and workflow elements), as well as the expansion and deployment of NST applications— analysts are able to create evolutionary prototypes. After some first examination of the universe of discussion, the analyst can thus make his interpretation of the main functional requirements which can immediately be incorporated in a working prototype and shown to, for instance, future end users. Based on their feedback, the analyst can then adapt his model, after which a new version of the prototype can be regenerated and demonstrated. Next to iterations related to the optimization of the model (still focusing on the same universe of discussion), the analyst can also enlarge the model by extending the universe of discussion covered by the application. That is, in initial stages one can opt to model (and further develop with craftings) only a part of an application and, later on, to extend the application towards other areas of the organization. This way of working is clearly related to the *model* variability dimension: the analyst only defines a model and the other dimensions (craftings, technologies and expanders) are at that point irrelevant and can be specified later on.

Once (a part of) the model has been defined, all basic functionality offered by the expanders is present within the generated application. In case additional functionality is required (i.e., not provided by the expanders), this can be added by developers within the provided anchors in the generated code or by additional files (e.g., classes). This code can be added and developed independently from the model (e.g., analysts can keep on working on the extension of the model while developers start adding additional crafting code) as the craftings can be harvested and reinjected from one model to the other (as long as the craftings do not become incompatible with the newly defined model, which could for instance be the case when programming the implementation of a custom finder method using an attribute that would be deleted later on). The analyst can inject these craftings into

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

his prototype as well, thereby validating whether the added code fits the customer requirements and crosscheck them with end users if required. Analysts can clearly see how specified customizations map onto crafting code. This can be helpful in assessing their complexity (e.g., by checking the size of code required) and maintaining the current state of the project (which customizations have been completed and which have not). This part of the project management is clearly related to the *crafting* variability dimension.

As the analyst verifies and inspects the customizations made by the developers, he might notice that some functionalities are not only relevant for the specific case at hand and might in fact prove their usefuless in other and future applications as well. At that point, certain functionalities within the craftings can be incorporated or "generalized" into the expanders. As each of the functionalities needs to adhere to the theorems of NST, needs to be (almost) free of bugs, etcetera this typically only happens after the functionality has been thoroughly tested in the context of multiple projects in such a way that sufficient experience in this matter has been gained. As of then, the functionality is removed from the application's crafting part and becomes available for all other applications as well. Therefore, this part of the project management is clearly related to the *expander* variability dimension.

Finally, the expanders can be used to employ various (combinations of) technology infrastructure. This is typically independent from the model (as it is mostly also not a responsibility of the analyst to take committing decisions on this area) and expander version. Clearly, the craftings need to be written in a certain technology or language which introduces a dependency and might imply adaptations to craftings in case a certain technological infrastructure option is chosen. For instance, when a certain GUI framework is chosen and craftings are added in this part, a future change of GUI framework might imply changes to the previous craftings (which need to be rewritten in the newly chosen GUI framework). In other cases however, technological infrastructure decisions can vary freely from code within the craftings, such as in cases where the chosen technology is irrelevant to the craftings (e.g., craftings in the view layer will not be impacted by changing the selected database) or when the craftings are all encapsulated from the changing technology (e.g., plain non-EJB specific Java code within anchors in a class using EJB annotations). Therefore, this part of the project management is clearly related to the *technology infrastructure* variability dimension.

VI. Conclusion

This paper presented one in-depth case study of an NST software application in an educational context and analyzed the different dimensions in which it could evolve. Additionally, three previously documented cases were reinterpreted using the same point of view. Based on this, four general variability dimensions were proposed.

This paper is believed to make several contributions. From a theoretical side, inductive reasoning based on our cases allowed the formulation and illustration of four variability dimensions, which might be the (or at least a subset of the) orthogonal dimensions along which a typical NST application can evolve. At the same time, these variability dimensions clarify that the concept of an overall application "version" is not applicable for NST applications as a specifically deployed application is the result of a combination of choices for each of the variability dimensions. For practitioners, this paper contributes to the set of case studies available on NST (as well as a reinterpretation from some previously documented cases), which might provide them with a better insight regarding the application potential of the theory in practice.

Next to these contributions, it is clear that this paper is also subject to a set of limitations. That is, we proposed the set of variability dimensions based on a limited set of case studies. Although the size, complexity and industry of the cases were different, their modest amount still limits the generalizability of our findings. Therefore, future research should be directed towards the analysis of additional cases, including information systems being even larger and more complex. These additional cases might confirm, and possibly extend, the variability dimensions proposed in this paper.

References

[1] P. De Bruyn, H. Mannaert, and P. Huysmans, "On the variability dimensions of normalized systems applications: Experiences from an educational case study," in Proceedings of the Tenth International Conference on Pervasive Patterns and Applications (PATTERNS) 2018, 2018, pp. 45–50.

[2] R. Agarwal and A. Tiwana, "Editorial—evolvable systems: Through the looking glass of IS," Information Systems Research, vol. 26, no. 3, 2015, pp. 473–479.

[3] H. Mannaert, J. Verelst, and K. Ven, "The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability," Science of Computer Programming, vol. 76, no. 12, 2011, pp. 1210–1222, special Issue on Software Evolution, Adaptability and Variability.

[4] ——, "Towards evolvable software architectures based on systems theoretic stability," Software: Practice and Experience, vol. 42, no. 1, 2012, pp. 89–116.

[5] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design. Koppa, 2016.

[6] M. Op't Land, M. Krouwel, E. Van Dipten, and J. Verelst, "Exploring normalized systems potential for dutch mods agility: A proof of concept on flexibility, time-to-market, productivity and quality," in Proceedings of the 3rd Practice-driven Research on Enterprise Transformation (PRET) working conference, Luxemburg, Luxemburg, September 2011, pp. 110–121.

[7] G. Oorts, P. Huysmans, P. De Bruyn, H. Mannaert, J. Verelst, and A. Oost, "Building evolvable software using normalized systems theory: a case study," in Proceedings of the 47th annual Hawaii international conference on system sciences (HICSS), Waikoloa, Hawaii, USA, 2014, pp. 4760–4769.

[8] P. Huysmans, P. De Bruyn, G. Oorts, J. Verelst, D. van der Linden, and H. Mannaert, "Analyzing the evolvability of modular structures: a longitudinal normalized systems case study," in Proceedings of the Tenth International Conference on Software Engineering Advances (ICSEA), Barcelona, Spain, November 2015, pp. 319–325.

[9] P. Huysmans, J. Verelst, H. Mannaert, and A. Oost, "Integrating information systems using normalized systems theory: four case studies," in Proceedings of the 17th IEEE Conference on Business Informatics (CBI), Lisbon, Portugal, July 2015, pp. 173–180.

[10] P. De Bruyn, P. Huysmans, and J. Verelst, "Tailoring an analysis approach for developing evolvable software systems : experiences from three case studies," in Proceedings of the 18th IEEE Conference on Business Informatics (CBI), Paris, France, August-September 2016, pp. 208–217.

[11] M. Lehman, "Programs, life cycles, and laws of software evolution," in Proceedings of the IEEE, vol. 68, 1980, pp. 1060–1076.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Integrating Portable Micro-CHP into a Smart Grid

Richard Pump Arne Koschel Volker Ahlers
Department of Computer Science
University of Applied Sciences and Arts
Hannover, Germany
Email: {richard.pump, arne.koschel, volker.ahlers}@hs-hannover.de

***Abstract*—We present an architecture to integrate a portable micro combined heat-and-power (pmCHP) unit into a smart energy grid. The pmCHP is a gateway technology to bridge conventional vehicles battery electric vehicles, increasing range and comfort. Furthermore, pmCHP are to be used in the house within a connected smart energy grid. A software system is required to drive the pmCHP operation within building and vehicle. The System needs to be highly adaptable to accommodate the high amount of changes a novel device will undergo as it is introduced into a real world scenario. To find the best architecture, we design three different architectures using different architectural styles and evaluate them based on five categories of software quality. We conclude that a Service-Oriented Architecture (SOA) using microservices provides a higher quality solution than a layered or Event-Driven Complex-Event-Processing (ED-CEP) approach. Future work will include implementation and simulation-driven evaluation.**

Keywords–Smart Grid; pmCHP; microservices; service-orientation; layered architecture; CEP; scenario-based evaluation

I. Introduction

This article is based on our previous paper [1] and expands upon it. A section showing the exact requirements for the development of the architectures was added as well as further information about the developed architectures giving more insight into the designs. Also, the evaluation was expanded, adding information about the application of the scenarios to the architectures, documenting the changes.

The current energy distribution system of Europe is in change, former tree-like distribution networks are replaced with small autonomous microgrids, which imitate a peer-to-peer network [2]. With ever growing demands for electrical energy old, tree-like structures do not scale well enough to further justify their construction. Peak loads demand over-building for capacities that are only used during a very small amount of time. With a peer-to-peer network, distributed energy generation can be introduced to the grid much easier, which allows every participant of the grid to generate and consume energy opening a new market for small scale energy trading. The change to distributed generation also alleviates the issue of overbuilding to compensate peak loads and allows more efficient expansion of the electrical grid to cover more extensive loads. This however requires a high amount of coordination between the different participants of the grids, resulting in ubiquitous automation of the grid.

In the same vein the automotive industry is currently changing. To combat global warming, cars have to meet strict emission goals to be allowed to operate in some countries. With current cars being optimized to the boundaries of engineering, alternative sources of propulsion are under review, resulting in an emergence of electrically driven cars, called battery-electric vehicles (BEV). However, the development of affordable BEVs with comparable range to conventional combustion-driven vehicles is slow, current BEVs often reach only half the maximum range of conventional cars. Additionally, BEVs only reach their maximum range without passenger climatization, since the energy needed for air conditioning and heating is drawn from the same battery as propulsion. Using a BEV therefore requires careful attention to the available driving range, since recharging station are comparatively scarce and a recharge often takes a lot of time. The fear of being stranded somewhere along the way when using a BEV is a major impediment to customer acceptance.

To alleviate issues in BEV and to bridge the gap between BEV and ICE-cars, the *University of Applied Sciences and Arts Hannover* is currently developing the portable micro-combined-heat-and-power unit (pmCHP). The pmCHP is a small scale version of conventional CHP-units, weighing less than 40 kilograms, and can be carried by a single person. Utilizing the co-generation of heat and electricity to achieve a very high efficiency, the pmCHP is designed to be used in a BEV or to be connected to a smart energy grid in a house. In the BEV the pmCHP generates heat (which can be converted to coldness) and electrical power, increasing the range of the car directly and allowing the conditioning of the passenger compartment [3]. It is comparable to a conventional range-extender, a small combustion engine, which generates electrical power to increase the range of a BEV, but is more efficient and provides conditioning. When the pmCHP is not needed in the BEV, it can easily be unplugged and carried to another point of use, for example in a house. In the house it supports the local heating and energy production, covering peak loads and recharging the local storages. This however requires an integration of the local smart energy grid to identify moments of peak load or other scenarios, in which to use the pmCHP efficiently. For example, the generation of electricity has to be coordinated with other, less flexible producers, like solar cells or small scale wind energy.

In this article we will present a smart grid integration of the pmCHP, starting with defining the requirements of the software, selecting the best architecture and showing the interoperability with the established smart grid standards. To start off, Section II will give an overview of the relatively small amount of related works. Requirements will be explained and listed in Section III and Section IV, before resulting architectures are presented in Section V. After the presentations of the architectures, we will show our process of architecture

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

comparison in Section VI, which is based on a scenario driven approach. The article ends with an evaluation in Section VII and our conclusion in Section VIII.

II. Related Works

Regarding software architectures for the smart grid, mostly interactions are standardized. For example, the standards 61968/61970, designed by the International Electrotechnical Commission (IEC) describe a global domain model of the smart grid with predefined interfaces and messages. The standards however do not describe a predefined internal software architecture.

In [4], Reinprecht et al. describe the IEC Common Information Model (CIM) architecture, which is a layered architecture that ensures standard-compliant implementation over the different levels of the architecture. The authors describe multiple SOA-based designs, which were created for the Smart Grid Interoperability test. A comparison or evaluation of the architectures is not mentioned.

Appelrath et al. [5] show a reference architecture for smart grid software. It describes general interfaces for abstract devices, a real device might be composed of multiple abstract ones. However, neither a concrete implementation nor an evaluation of alternatives is presented.

An architecture to operate a pmCHP testbed is presented in [6]. There is no connection to the smart grid, although microservices are used to provide high architectural flexibility.

To compare different architectural designs, Kazman et al. [7] present a scenario-driven comparison method that provides the general process used in this work.

The most important quality-aspects of smart grid software are proposed by the NIST in [8]. Since the Smart Grid is critical infrastructure, one of the most desired qualities is the availability of the devices. These qualities are considered when comparing the different designs in Section VI.

All considered, there is no concrete work on how to integrate a pmCHP into a smart grid, let alone an evaluation of suitable software architectures for this purpose, known to the authors of this article.

III. Requirements

Building an architecture requires a clear set of goals, often referred to as requirements. The requirements define a clear goal of what the software has to achieve to be successfully developed and provide a baseline, against which a software can be evaluated. For the pmCHP the requirements are split into three categories, based on the planned operations for the device. There are general requirements describing its use in all scenarios, requirements for the usage in the car and requirements for the usage in the building.

A. General Requirements

The general requirements describe the essential processes in operating the pmCHP, regardless of the location of usage. Overall, they contain safety relevant issues, user experience, and general operating procedures.

First, the pmCHP has to choose the operational strategy, which decides how the pmCHP is going to be driven. The pmCHP can be used in three different modes; electricity-driven, heat-driven, and combined heat- and electricity-driven. The electrically-driven mode controls the production of the pmCHP depending on the needed electricity, heat is seen as byproduct and will not be produced if no electricity is needed. In heat-driven mode, the pmCHP uses the heating requirements as the control value, the combined operation mode just produces depending on whichever energy is needed at the time. Each of the modes has different advantages, depending on the situation the pmCHP is employed in. For example, operating the pmCHP in heat-driven mode in a house secures government grants on electricity produced through the usage of the power-heat-co-generation, due to its high efficiency. However, choosing the wrong operational strategy will result in a bad user experience, in the BEV running in heat-driven mode will not result in higher ranges, as long as the passengers do not require air conditioning.

After choosing the general strategy for the pmCHP operation, operational plans have to be made. Ideally, the pmCHP is operated for long uninterrupted periods of time, with the combustion engine running in its optimal operating point. This decreases wear on the moving parts and also guarantees high efficiency with low fuel consumption, allowing the pmCHP to operate at the minimum costs to the user. For the operational plans, external inputs have to be considered, depending on the location of usage.

With the pmCHP up and running within optimal parameters, the device has to be monitored. Engine temperature, rpm, fuel flow, air flow, cooling flow, etc. have to be watched carefully to diagnose errors early and allow safe shutdowns. To accommodate this, the software has to handle many different sensor inputs and correlate the sensors into an operational state.

The state monitoring is needed for two different purposes, first and most important is the emergency shutdown. If the state of the pmCHP is critical, for example if the engine is leaking fuel and has caught fire, the pmCHP has to be shutdown safely, disconnecting it from other devices and stopping its operation. The shutdown has to be fast enough to prevent further damage and has precedence over every other procedure in the software. Current operational parameters, like operational plan and strategy as well as sensor readings, have to be logged for examination.

Also, errors and problems have to be reported to connected devices and the user of the pmCHP. In the connected smart grid or the car, other electronic control units exist and need to know the state of the network, i.e., if devices like the pmCHP are performing as the control unit expects them to. The user needs the information about the device state to make decisions on maintenance.

Lastly the pmCHP-software should give the user an overview over the current operational state and planning to increase transparency of the device.

The general functional requirements can be summarized as follows:

GR01	Smart choice of the operational strategy.
GR02	Creation of operational planning depending on external inputs and operational constraints.
GR03	Monitoring of device state.
GR04	Emergency shutdown on critical or dangerous state.
GR05	Reporting of errors or malfunctions to other devices and the user.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

GR06 Displaying the current operational state and planning to the user.

B. Usage in BEV

After considering general requirements, we will present the requirements for the usage of pmCHP in a BEV. The usage in the car is dominated by the use of the pmCHP as a range extender, the requirements revolve around functions to increase range and comfort. An important point for the usage in the BEV is the lack of other electrical generators in the car. The generation of electrical power hence takes precedence over efficient usage of the pmCHP resources.

To facilitate operation in the car, the pmCHP has to monitor different aspects of the cars state. Most importantly, the pmCHP has to monitor the charge of the internal battery and the expected range depending on current consumption, planned route and expected consumption. If electrical energy is likely needed to complete the planned route, the pmCHP has to act and charge the battery before it depletes completely. Ideally, the pmCHP starts operation at the beginning of the drive to charge the battery and condition the passengers, so it can shut down when thermal requirements are met.

The primary function of the pmCHP is the generation of electrical power, if needed to continue driving. A different requirement is the operation as emergency range extender, when the battery is about to be depleted. Thermal energies produced during emergency mode are however considered as waste, therefore the pmCHP loses its efficiency advantage over conventional range extenders.

As already mentioned, the pmCHP shall condition the passenger compartment to increase passenger comfort during rides. The pmCHP-Software has to monitor the current temperature of the passenger compartment as well as the desired temperature. Furthermore, the pmCHP shall provide heat or cold (depending on the difference between desired and current temperature).

A more exotic requirement is the conditioning of the battery used for driving. Studies have shown that thermal conditioning of the battery can increase the range of an BEV considerably. While this would introduce additional thermal loads to be satisfied by the battery in a normal BEV, a pmCHP-equipped BEV can utilize the pmCHP to cover the thermal loads and profit from efficient co-generation of heat and power. To properly condition the car battery, the pmCHP has to monitor the batteries temperature and regulate it to the optimal temperature.

Summarizing these requirements results in the following list:

BR01 Monitoring of current range and expected driving distance.

BR02 Generation of electrical energy, if it is needed to continue driving.

BR03 Monitoring of the current passenger compartment temperature and desired passenger compartment temperature.

BR04 Conditioning the passenger compartment, if the battery can be charged.

BR05 Monitoring of the car battery temperature.

BR06 Conditioning of the car battery, if it will increase range or the battery can be charged.

C. Usage in the house

The usage of the pmCHP in the house is dominated by its connection to their devices. Integrated into a smart grid, the focus of operation changes to one of cooperation and coordination. Since there are a lot of alternative generators available in the house or the smart grid, the operation of the pmCHP focuses on optimal usage of co-generation effects. Connecting a pmCHP to a potentially global network with a lot of other clients opens the pmCHP to attacks, so security considerations come into play.

The first functional requirement is the proper communication of the pmCHP with other smart grid devices, via the expected smart grid standards like IEC61850. To facilitate coordination, communication is needed. By using the IEC61850 standard interoperability with other smart grid devices is ensured. Since no real modern smart grid is currently in use in Germany or the EU, strict adherence to the predefined standard is needed.

Another requirement is the monitoring of the attached heat storage. In contrast to the car, which has a battery, the house usually has no system for electrical storage (devices like the Tesla Powerwall exist, but are currently not in widespread use). However, most houses have some sort of thermal storage for heating and hot water, which can be used by the pmCHP to store thermal energy. This decouples the time of use from the time of generation and allows more flexibility for pmCHP operation.

The most important functional requirement is the efficient operational planning. The pmCHP-software needs to create efficient operational plans, which use the pmCHP to its maximum potential. To create the operational plans, the software needs to consider current and historical weather data, other energy producing devices within the smart grid, the operational specifics of the pmCHP (long, continuous periods of medium load), and the current price of electricity. The weather data shall be used to predict future loads for proactive generation. Other smart grid devices have to be considered during planning, since there might be no need for the pmCHP to run. Interestingly, the current price of energy also factors into the planning, since the pmCHP consumes fuel to generate heat and electricity. If buying electricity from another source is cheaper than running the pmCHP, it might be more sensible to not run the pmCHP.

In line with the monitoring of the current price of electricity, currently the German government gives grants on sales of electrical energy, if it was produced using the co-generation of heat and power. The pmCHP shall also tap into this stream of revenue by producing and selling electrical energy, if it is not needed in the users house, the heat can be stored and used later, and the profit from the energy sale exceeds the price of fuel.

Since the pmCHP is connected to a network when used in a house, additional functions should be provided by the software. First, the software shall provide the ability for remote maintenance, like software updates, error reporting and state monitoring. This allows the manufacturer to continuously improve upon the pmCHP software and, most importantly, fix security and safety issues in the software quickly.

In a similar fashion, the pmCHP-software needs to provide the ability of emergency control to the electrical grid operator.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Mainly a safety requirement, this allows for remote control of the pmCHP in case of emergencies, like blackouts or times of overproduction, which cannot be handled automatically.

Also, three requirements concerning security are imposed on the software. With hacker attacks on energy grids already being reality, protective measures have to be considered early in development of smart grid devices. The most important requirements for security are the protection of the hardware from malicious control, preventing physical damage to the pmCHP and its surroundings. For example, a hacker with free control over the pmCHP-control could run the pmCHP at a very high load, possibly causing a fire. With software protecting the pmCHP hardware and an update mechanism to change the software, the software itself has to be protected against unauthorized changes. Lastly the information of the user, like times of usage in the car have to be protected from attackers to prevent social or physical damage.

The requirements are summarized as follows:

HR01 Communication with the other smart grid devices.
HR02 Monitoring of the attached heat storage.
HR03 Smart creation of operational plans.
HR04 Sale of electrical energy, if the heat can be stored and profit can be made.
HR05 Provide remote maintenance support.
HR06 Provide emergency remote control to the grid operator.
HR07 Protection of physical components from attacks.
HR08 Protection of the software from unauthorized changes.
HR09 Protection of the users information.

IV. Requirements to software quality

After considering the functional requirements, quality requirements have to be looked at. Quality requirements do not define the softwares behavior like functional requirements, but instead describe the goodness of the software, which often is not directly measurable. Since the device and the projected usage environments are currently in development only a rough definition of quality requirements will be given.

The most important quality requirement is availability, which consists of robustness against errors in the system, time of restoration of service after failure and probability of failure. With the goal of improving the daily life of the user, failure of the software is very problematic as missing energy production can have large impacts to the users daily life. The direct interface to physical components makes failure very dangerous, therefore the availability is of very high importance, with short times of recovery after a failure and long times between failures.

Other very important quality requirements are safety and security, consisting of the security of the software (i.e., resistance to malicious use) and safety due to functional correctness. With pmCHP being part of the smart grid, they automatically become part of critical infrastructure, which needs to be protected extensively against attacks. Also, the direct cyber-physical-interface presented by the software requires high security and safety due to the large impact of malicious use or incorrect functioning of the software.

Also, rather important is the maintainability of the software, consisting of the ease of change and extension as well as conceptual integrity and testability. Since the software will be designed in a very early state of development of the pmCHP and the smart grid, the underlying requirements are not final and might change during the future. A good maintainability allows for easy, rapid change of the softwares components, which is critical in unclear scenarios. Adherence to the established smart grid standards also improves maintainability through ease of integration. According to the German 'Normungsroadmap 2.0' [9] three standards are primarily used in the Smart Grid.

- IEC 61850: Substation automation and protection
- IEC 61968/61970: Application level energy management system interfaces, CIM (Common Information Model)
- IEC 62351: Information security for power system control operations

We will concentrate on the IEC 61850.

However, performance is not as important as the other quality requirements, with the exception of the emergency shutdown and other hardware-related parts of the software. While parts of the software 'close' to the hardware are confronted with hard real-time requirements, those parts are considered to be very small. Other parts like the operational planning are not subject to hard deadlines, a late operational plan can always be executed a bit later.

Usability of the software is not very important, since the software is planned to be very autonomous, with little amounts of user interaction.

To summarize the following importances are derived for different quality aspects:

- Availability
  - High error tolerance.
  - High meant time between failures.
  - Low time of recovery after failure.
- Security/Safety
  - Protection against software manipulation.
  - Protection against abuse of the softwares functions.
  - High safety through functional correctness.
- Maintainability
  - High extendability.
  - High conceptual integrity.
  - High testability.
  - High standard compliance (esp. IEC 61850)
- Performance
  - Hard real-time requirements to the emergency shutdown / other close to hardware functions.
  - Soft real-time requirements elsewhere.
- Usability
  - Low usability requirements for the software.

V. Architectures

Based upon the aforementioned requirements, architectures can be developed. Instead of directly choosing an architectural style, we decided to first build a rough sketch and compare three different architectures based upon the sketch. By first designing the principal components of the software, we were

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

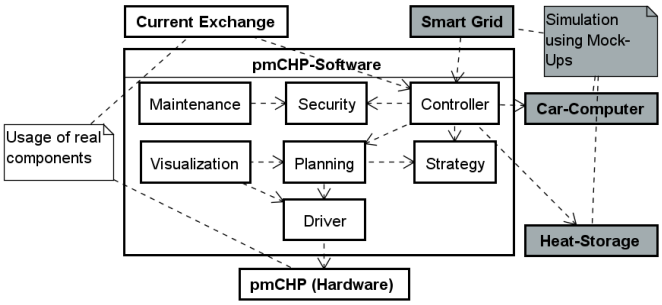


Figure 1. Rough sketch of the pmCHP-Software

able to keep conceptual integrity and continuity between the architectures, which allows proper comparison.

Figure 1 shows the principal components of the pmCHP software. Generally, the energy production will be handled on the basis of energy requests, which specify how much energy of which type is needed. The pmCHP can generate own requests and handle requests from external sources, this simplifies the handling of requests, since only a single mechanism will control the pmCHP.

The CONTROLLER coordinates the production of the pmCHP with its environment, be it the Smart Grid or the computer of the BEV. It receives energy requests, validates their security, and checks if they can be fulfilled, since requests might not fit in the current operational strategy or the generation capabilities of the pmCHP. The CONTROLLER relies on the STRATEGY component for decision making. Valid requests are handed the PLANNING for further processing. The CONTROLLER fulfills a wide number of functional requirements, since it contains the main logic for generating and handling requests. In further designs it will be split into different components, based upon architectural style.

The component STRATEGY provides the framework for the day-planning, as it decides, which operational strategy is used. The decision for the heat-driven/electricity-driven or combined mode is based upon the detected environment of the pmCHP. It fulfills the requirement GR01.

The component PLANNING is responsible for planning the day-to-day-operation of the pmCHP, according to the operational strategy. To create plans, the component uses the energy requests received fro the controller and further information, like the current operational strategy, data about previous operation and forecasts. After a plan was created, they are handed over to the DRIVER for execution.

The component DRIVER provides the interface between hard- and software. Most importantly, it monitors the pmCHP and provides emergency shutdown functions as outlined in requirements GR03 and GR04. Furthermore, the DRIVER transforms the operational plan into control-commands and provides this information to other parts of the software.

The VISUALIZATION component presents the current state and planned operation to the user of the device (GR06). Also, errors and warnings can be shown to the user, so corrective action can be taken if needed (GR05).

To facilitate remote access to the pmCHP, in case the software needs to be updated or other remote action needs to be taken, the component MAINTENANCE exists. It allows remote software updates, access to log files and current operational status of the pmCHP as well as remote control in case the grid operating company needs to control the pmCHP manually. This covers the requirements HR05 and HR06, as outlined in the requirements for the operation in the house.

Allowing remote access to the pmCHP Software without any security measures would be grossly negligent; therefore, a component SECURITY needs to take care of authentication and authorization of all incoming requests.

Other components like SMART GRID, CURRENT EXCHANGE, etc. are beyond the scope of the software and represent neighboring systems to interact with. However, these can not be ignored, since the necessary interfaces in the pmCHP software need to be considered when designing the software.

Based on the previously shown rough design, three different architectural styles were used to create three architectures: SOA, ED-CEP and layered.

A. *Service-oriented architecture (SOA)*

First, we will give an overview of the service-oriented architectural style itself, before we show our implementation of the SOA for pmCHP. The main drive behind SOA is a very loose coupling and high coherence in components [10]. A special form of SOA are the microservices, in which components are designed for easy rapid replacement. Since a SOA is highly flexible between components, the service oriented architecture is a good match for a cloud environment, in which components can be deployed across different locations without having to adopt the component to the deployment location. Very often web services using REST or SOAP are utilized to facilitate inter-component communication.

A general structure of a very comprehensive service oriented architecture is shown in Figure 2. In total, the general SOA consists of five layers and two cross-cutting parts.

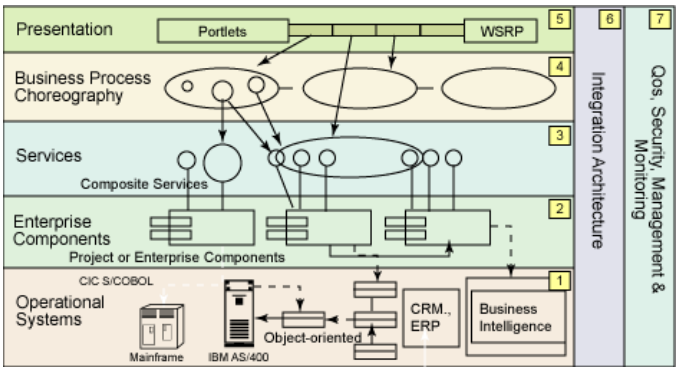


Figure 2. The service oriented architectural style according to [10].

The uppermost layer, also called the presentation layer (5) contains all components that allow interaction with the software system, like the graphical user interface or services that other software systems can use. To access the softwares functionalities, the presentation layer (5) accesses functions from the business process choreography layer (4). In this layer the business processes of the software are modeled and executed. It contains the major functionalities of the software, often

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

modeled in Business Process Model and Notation (BPMN), allowing domain experts the creation and validation of business processes and therefore functionalities. The business processes from layer 4 access the services in the service layer (3). A service is a defined interface enclosing single functions, independent of their implementation. The services can be modeled very fine grained allowing swift replacement. Generally, a business process accesses many services to complete, each service fulfilling a single step in the process. Services are independent of each other and can be recombined easily in new business processes allowing a high degree of re-usability. The services are implemented in applications contained in the application layer (2). An application might implement multiple services or just a single one, depending on the complexity of the desired functionality. Applications cluster services by semantics or other constraints (e.g., all services regarding a single database might be implemented in a single application). Rather unusual, Figure 2 shows the operational systems, which often are not considered in a cloud environment, since the SOA can be developed platform independent. Working in the background, the integration architecture (6) serves as 'glue' providing functions to connect all parts of the SOA. For example, a service bus is needed to connect the services to the implementation and present the endpoints to the business process choreography. Another important cross cutting part of the SOA are the quality monitoring and management components contained in (7). These allow the implementation of quality aspects like redundancy of services and security.

In the following section we are going to design an architecture based on the aforementioned style. Since the general business processes are already roughly designed a top-down approach is chosen for the design, starting at the business process, which will be decomposed into services grouped into applications, iteratively refining the softwares structure. For the services we follow a microservice-approach, defining services small enough so they can be replaced within a day. This allows for easy maintenance as shown later. Not all business processes and services will be shown, rather we will concentrate on the usage of the pmCHP in BEV or house, leaving, e.g., maintenance out of scope for this article.

1) Processing of energy requests: The main operation of the pmCHP revolves around energy requests. These stem from the smart grid, the car or the pmCHP itself and are the driving force behind the pmCHP-operation. An energy request contains information about the energy that is needed in the environment.

We differentiate between external and internal energy requests as shown in Figure 3. External energy requests are first

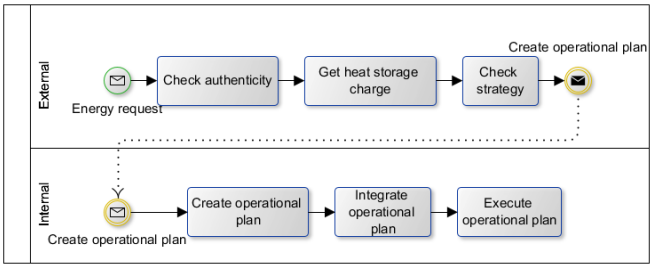


Figure 3. Processing energy requests in the SOA.

checked for authenticity and integrity to ensure no malicious use happens. Afterwards, the current charge of the attached heat storage unit as well as the current operational strategy, has to be checked. If the energy request does not fit the strategy or the generated heat can not be stored in heat-driven-mode, the energy request has to be declined. Otherwise the request meets all requirements and can be turned into an operational plan. This is shown in the internal part of the process, which is also the entry point for internally generated energy requests.

To fulfill the energy request, first a single operational plan containing the single energy request is created. The plan conveys information about the needed energy and the duration of the energy request (e.g., 1 kW of electrical energy for 1 hour). This plan is then integrated into a copy of the current operational plan, which replaces the original when the plan is executed afterwards. If the plan can not be integrated into the current operational plan for whatever reason (e.g., the needed energy exceeds the current production capacity), the energy request is declined and the execution of the original current operational plan continues.

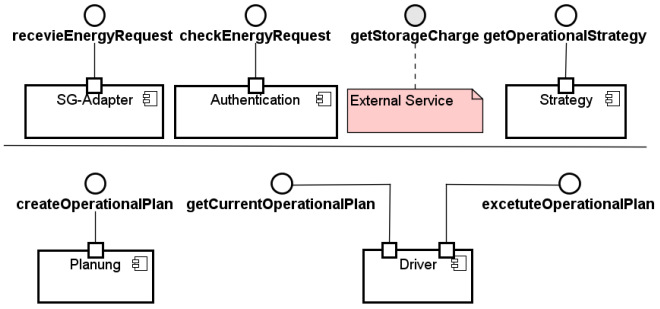


Figure 4. Services for processing energy requests in the SOA.

Figure 4 shows the services used in the business process as well as the applications implementing the services. The services are rather straightforward and fulfill a single responsibility indicated by their name, following the microservice-approach.

2) Energy requests in the BEV: The usage of pmCHP in BEV is a novel deployment of CHP-technology. Here three main business processes can be extracted from the requirements BR01-BR06, as shown in Figure 4. The three processes model the different operational strategies of the pmCHP, using it as a range extender (top), as an air conditioning unit (middle) and as a power conditioning unit (bottom).

In the usage as a range extender the pmCHP checks the currently planned route as well as the current battery charge. If the range is insufficient to complete the planned route, an operational plan has to be created to generate the necessary electrical energy. Using the pmCHP as air conditioner or power conditioning unit works similarly, as the physical attribute of the environment is measured and an operational plan is created to move the temperature to the desired or optimal value.

To create operational plans, the energy requests are sent to the aforementioned internal part of the energy request processing.

The services used for the usage of the pmCHP in the BEV are completely external as all of them have to be implemented

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

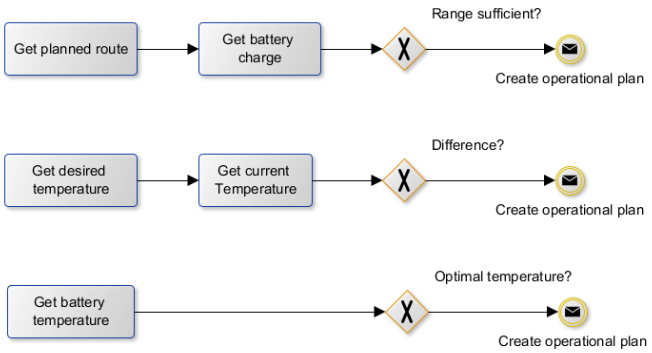


Figure 5. Business processes for the operation in the BEV.

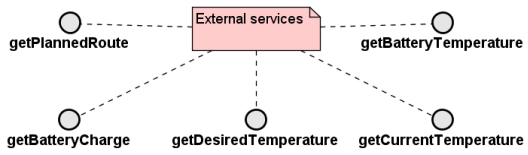


Figure 6. Services for the request creation within the car.

within the cars system software. While they are needed for the correct operation, they are not within the scope of our design.

3) Energy requests in the house: The second cornerstone of pmCHP-operation is the usage of pmCHP in a house, connected to a smart energy grid. Main requirements regarding energy generation are HR01-HR04. In the house, most operational plans are created upon external request through the smart grid, which was shown before. The design of business processes for the internal request generation required for HR04 is left as an exercise for the reader.

4) Overview: Continuing the design as a service oriented architecture, we result in the general architecture presented in Figure 7. Not all processes and services are shown.

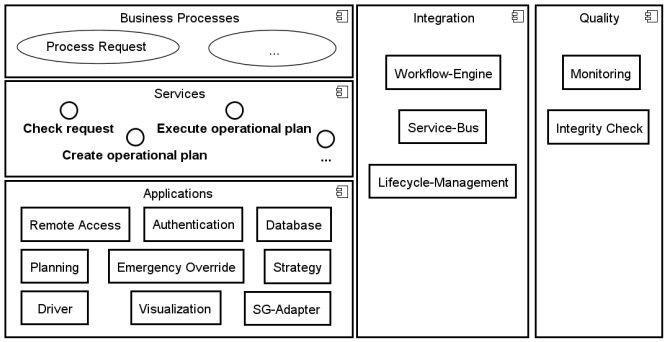


Figure 7. Overview of the complete SOA.

Excluding the presentation layer and the operational systems layer from Figure 2, our architecture conforms to the style presented before. In comparison to the rough sketch in Figure 1, the CONTROLLER was split into different business processes, as expected, while other components like the DRIVER were converted into applications with services. Furthermore, new components were introduced to support the service model, like the WORKFLOW-ENGINE providing orchestration or the SERVICE-BUS, which connects service endpoints to applications.

Overall, to create the SOA we transformed the requirements into business processes, which were decomposed into services. The services were grouped into the applications, which are closely aligned to our previous rough sketch. We achieve comparability to the other architectures by reducing our freedom of design.

B. Event-driven complex event processing (ED-CEP)

In the event-driven complex event processing architecture, processes are mapped to event chains, which start with simple events, later combined them into more complex events. Every event is consumed by an event processor, which either creates a new event, combines multiple events into one or calls an external service.

The general structure of ED-CEP-Architectures is shown in Figure 8.

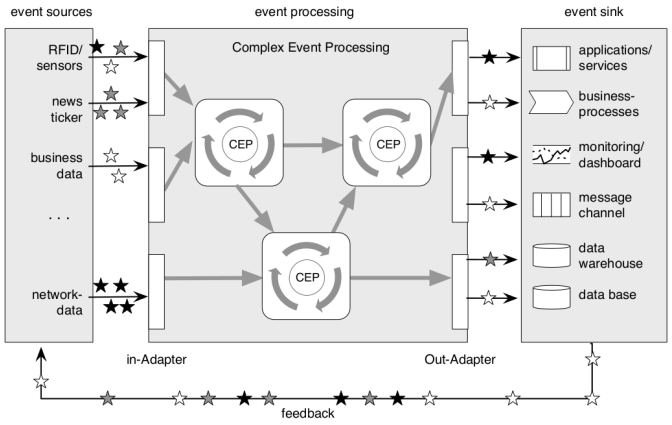


Figure 8. The event-driven complex event processing architecture according to [11].

Events originate in event-sources like sensors or news tickers and are fed into the event processing network (EPN). The EPN contains the event processors, which process different events according to their rule-set. Within the rules, conditions and actions are defined. The conditions can be based on the timing of events, values contained in events or the presence of a combination of events, etc. If a condition is met, the processor executes an action as outlined before. The events 'leave' the EPN through the event sinks, which are special processors that execute actions (e.g., call a certain service of an application) if they receive the corresponding event. Applications in ED-CEP are similar to applications in SOA and group functions that can be accessed by the EPN.

The ED-CEP is similar to the SOA, the functional requirements are modeled within the EPN instead of business processes while a 'framework' takes care of technical functions. It is therefore possible to exchange the orchestration layer of a SOA with an EPN.

To design an ED-CEP architecture, event sources are identified, their events and the transformation of those events are defined. The business processes are modeled as event chains often ending in a service call to an external application.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

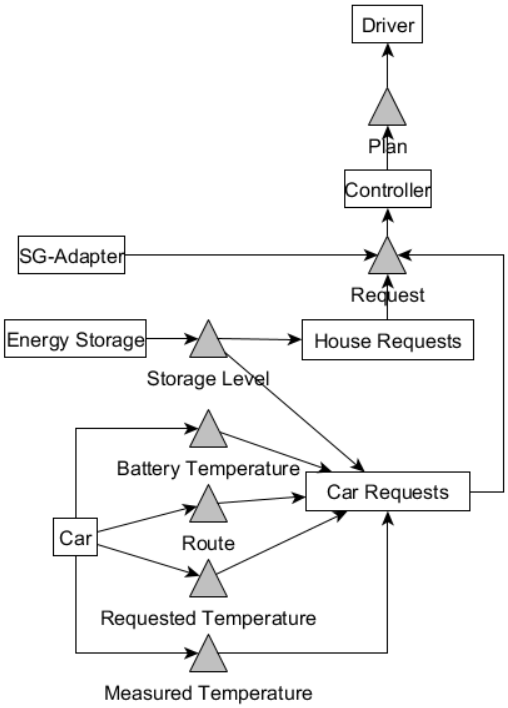


Figure 9. Creation and processing of energy requests in the EPN.

Similar to the SOA we will start by modeling the processing of energy requests as well as their creation in the car.

Figure 9 shows a part of the EPN with event sources marked as triangles and processors as rectangles. On the left side are the event sources involved.

The first event source is the SMART GRID-ADAPTER, which connects the system to the smart grid. Energy requests are received as messages over the network and transformed into *Request* events, which are processed by the CONTROLLER. The CONTROLLER contains the rules, which can be extracted from the business processes shown in the section about the SOA. For example, a (very simple) rule to process an external request might look like this:

```
IF(request.energyNeeded < pmCHP.availablePower)
  CREATE new Plan(Request);
```

The CONTROLLER then generates a new *Plan* event, which is consumed (executed) by the DRIVER.

For the usage in the BEV or the house energy requests are created by the EPN through other chains of events and processors, shown in the bottom half of Figure 9. The CAR is an event source for four different events, containing information about the cars current operational status, which are processed by the CAR REQUESTS processor. Again, the internal modeling of the processor rules can be extracted from the business processes used in the SOA and is therefore not shown here. (Each operational mode is modeled as a single rule, comparing the different events to each other and generating *Request* events if necessary.)

The complete event processing network for the pmCHP-software is shown in Figure 10.

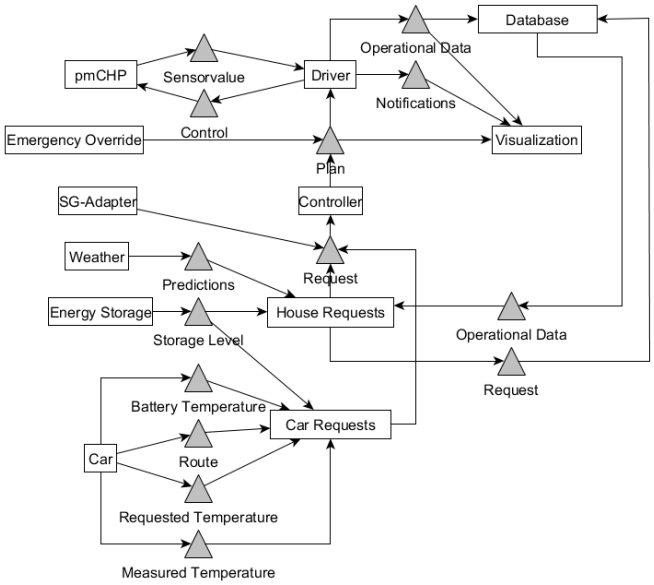


Figure 10. Complete EPN of the ED-CEP architecture.

As opposed to the SOA, there is no clear separation of the business processes but rather a complex network of events and processors with behavior hidden in the internal rules of the processors.

Additionally, further components are required as a framework for the EPN, shown in Figure 11.

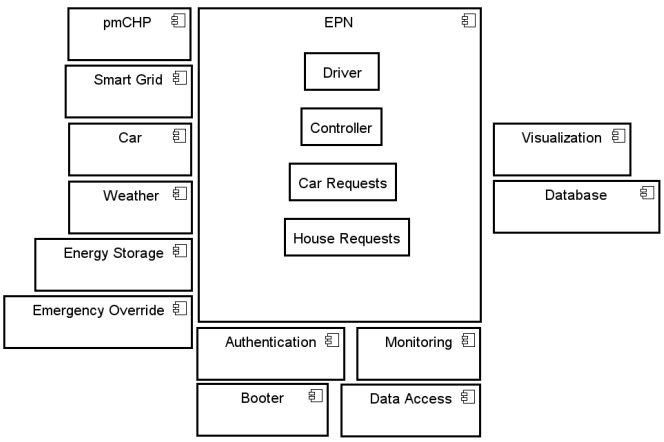


Figure 11. Overview of the ED-CEP architecture.

With the event sources on the left and the event sinks on the right, four components remain in the bottom. These four provide cross cutting functionality, which concerns the EPN or provides functions to the EPN. For example, the MONITORING monitors EPN performance and reliability and takes action if necessary to sustain availability.

Overall, in the ED-CEP business processes are mapped to event chains containing events and rules on how to transform the events into different events or service calls. The modeling of processes as chain of rules allows flexibility in program logic that is directly visible in program structure, as processes

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

are modeled as EPN components.

The similarity to the rough sketch of Figure 1 can be seen in the overview, as components like the CONTROLLER can be clearly identified. Also, a similarity to the SOA is visible, as the EPN can be seen as an orchestration layer for service calls. This confirms the coherent modeling of both architectures under the same constraints, allowing later comparison under non-functional aspects.

C. Layered Architecture

Lastly, we will present an architecture designed using the layered style. The layered style is one of the oldest architectural styles, dating back to a publication by Dijkstra in 1969 [12]. Generally speaking, the software is divided into multiple layers of increasing abstraction, from the concrete physical system towards the ideal software system. Layers can be observed in other styles as well, as already shown in SOA (Figure 2).

The general style of a layered architecture is shown in Figure 12.

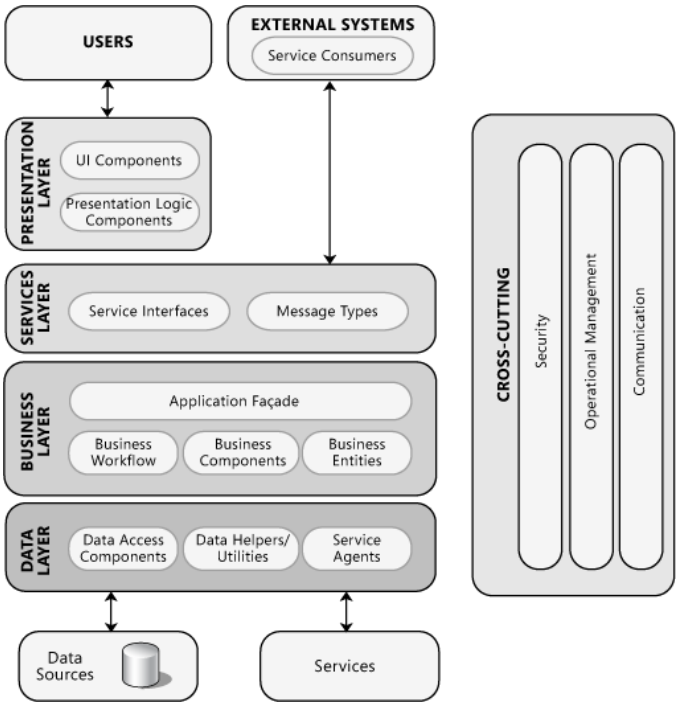


Figure 12. The layered architectural style according to [13].

Each layer can use functions from any layer below it, but not functions from layers above. At the top is the presentation layer, which provides users an interface to the software as well as some logic needed to show the user interface. Below is the service layer, providing interfaces for the presentation layer or external applications. To provide their function, the service layer accesses the business layer, sometimes through the facade, which hides the components of the business layer from external access, decoupling the components further. The whole business logic of the system is implemented within the business layer, comparable to the EPN in the ED-CEP or the business processes in the SOA. The lowest layer, also called the data layer contains technical parts of the system, which allow access to databases or other systems.

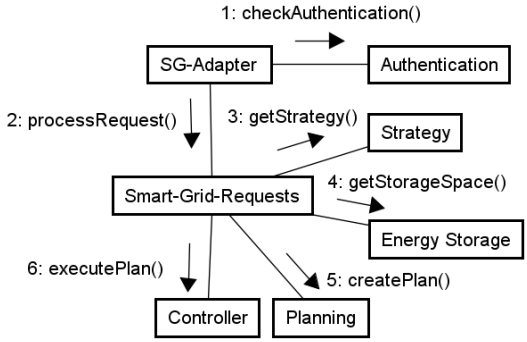


Figure 13. Processing energy requests from the smart grid in the layered architecture.

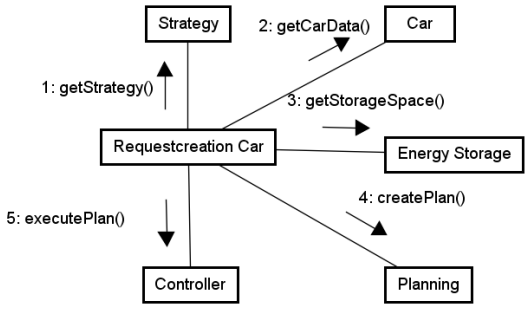


Figure 14. Request-creation for the BEV in the layered architecture.

Some aspects of the software however cannot be attributed to a single layer, since they are needed on all layers, like security. These cross cutting components are often distributed across all layers.

Designing a layered architecture can be done in multitude of ways, we will first model the most important business processes as function calls between components and the order the components into the appropriate layers.

Again, the process of processing an energy requests is modeled first, before we model the creation of energy requests in the car.

Energy requests are received from the smart grid by the SMART GRID-ADAPTER, which is responsible to check the request via the AUTHENTICATION, before handing the request to the SMART-GRID-REQUESTS component, which processes the request. To process the request, first the operational strategy and the current energy storage space is checked, similar to the process shown in the SOA or the ED-CEP. If the request is valid and can be fulfilled an operational plan is created via the PLANNING and sent to the CONTROLLER for execution. The process is modeled as function calls between components.

The same structure can be observed within the request creation for the BEV, shown in Figure 14.

First, the operational strategy is retrieved to determine the functionality needed at the moment (e.g., usage as range-extender or air conditioning). Afterwards the necessary data about the car, like route, temperature, etc. as well as current

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

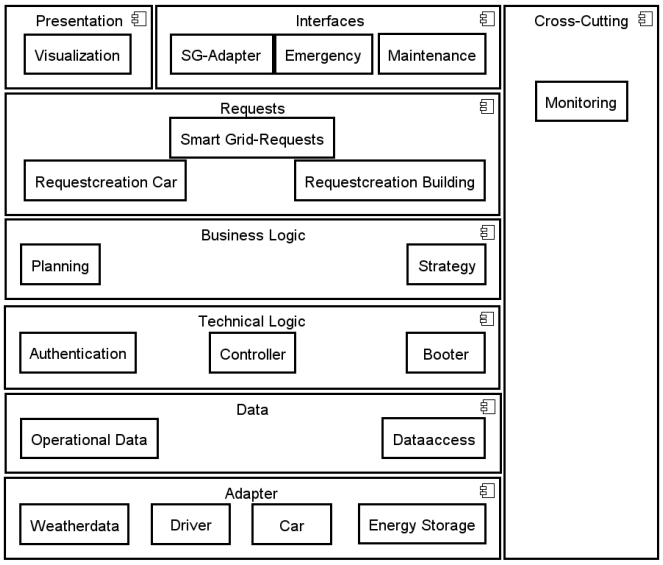


Figure 15. Overview of the layered architecture.

battery charge is retrieved before creating an operational plan to be executed by the controller.

Continuing the process for all business processes the final architecture can be seen in Figure 15.

As shown in the architectural style, the layered architecture contains a presentation and an interface-layer to facilitate access to the software functionalities.

Below those layers, the three logic layers are found. First the request layer, which is responsible of handling and creating energy requests to drive the pmCHP-operation. It is comparable to the business process choreography of the SOA shown in Figure 2. The business logic layer is found in the middle of Figure 15. Containing the PLANNING and STRATEGY components, the business logic layer is responsible for defining the core operation of the pmCHP. It is less abstract than the general request creation occurring above, but not as technical as the layers below, which are system specific. The bottommost logic layer is the technical logic, which contains components that implement system specific logic, like the CONTROLLER, which has to execute operational plans via the driver. Below we find a data layer, allowing access to different data sources within the pmCHP-system. At the bottom of Figure 15 the adapters are found, allowing access to different parts of the pmCHPs environment.

The layered architecture contains a lot of similar components when compared to the SOA or the ED-CEP with the most distinct difference being the spread of the business logic across the three logic layers. All three architectures are designed to be functionally identical, fulfilling all functional requirements allowing the comparison of the architectures without obvious deficiencies being visible at this point.

VI. COMPARISON OF ARCHITECTURES

Comparisons of architectures are a useful tool to increase software quality at an early stage of the development cycle [14]. Choosing the wrong architecture can decrease the maximum achievable quality by forcing bad design. For example, forcing an EPN into a micro-controller controlling a toaster will have less performance and increased development cost over a monolithic software. (Assuming the software is tasked with just turning the toaster off as soon as a signal is received. EPNs most likely handle complex scenarios better than monolithic approaches.)

A. Scenario-based comparison

Multiple ways exist to compare different architectures, but most commonly scenario-based methods are used. Scenario-based methods use scenarios to estimate necessary changes to the architecture, which in turn can be used as an indicator for the quality of the architecture. The first step of a comparison is the definition of the architectures in some form, as already described in the previous section. In a second step, scenarios describing possible usages or changes of the architecture are defined, each providing a measurable way to describe quality. The scenarios are grouped after the five general aspects of software quality and use the rough system sketch as a common baseline for all architectures.

a) Availability: Availability scenarios describe situations where the system has to take certain countermeasures to provide uninterrupted operation. Availability is the most important quality in an energy providing system [8].

Ava01 The DRIVER crashes due to an error, the system realizes the failure and immediately switches to a backup.

Ava02 The DRIVER receives a single incorrect measured value outside of the defined thresholds for this sensor. Instead of immediately shutting down the pmCHP the DRIVER averages values and prevents shutdown due to measurement errors.

Ava03 The connection between the DRIVER and the pmCHP is severed and cannot be reestablished. An error is presented to the user and the pmCHP switches to a safe operating mode instead of shutting down immediately.

Ava04 A usual high amount of energy requests is received from the smart grid. After a certain threshold is reached, the CONTROLLER rejects all further requests to provide protection against overload-attacks.

b) Security: Security-scenarios describe situations where the software is possibly used in a way that it is not intended and unwanted. In an interconnected network with access to physical systems, security is one of the most important qualities the software has to achieve.

Sec01 A different system tries to access a pmCHP-software functionality, the authenticity of the accessing system is checked, before access is granted.

Sec02 When the pmCHP is activated, a minimal software checks the integrity of the pmCHP-software using a digital signature. If the signature is not correct, an error is presented to the user, and the software does not start.

Sec03 A manipulated component tries to access a function of the DRIVER, which it normally would not access and is not authorized to do so. The component SECURITY recognizes the unauthorized attempt, prevents it and produces an error message shown to the user.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

c) Safety: In contrast to security, safety-scenarios are describing potentially dangerous situations in the normal operation of the pmCHP-software. Again, safety is rather important in operating an energy generating device, as failures can harm humans and the operating environment.

Saf01 The DRIVER continuously monitors all of the pmCHPs sensors and detects dangerous operation. If a dangerous operation is recognized, the DRIVER transfers the pmCHP into a safe mode of operations, possibly even shutting it down.

Saf02 All control-signals are checked by the DRIVER, ignoring signals that might damage the pmCHP.

d) Maintainability: Since the smart grid is not completely clear at the moment, adaptability and maintainability is somewhat needed. The following scenarios include likely changes and developmental processes of the software's lifetime.

Mai01 After the end of the pmCHP-development a different developer is tasked to add smart market integration to the pmCHP-software. The smart market component needs to accept requests from the smart grid, overview their execution, and take care of the billing aspects according to the energy contract.

Mai02 The emergency shutdown shall be tested intensively; the required components can be interchanged with mock-ups without changing the DRIVER.

Mai03 The systems architecture is checked by a software architect. Similar problems are solved in similar ways using similar architectural or design patterns.

e) Performance: Performance is not overall important to the pmCHP-software, only a single scenario is presented.

Per01 A malfunction of the pmCHP requires an emergency shutdown, the shutdown happens fast enough to prevent damage.

f) Usability: Usability describes the grade at which the user's interaction is eased by good interface and software design. Since there is almost no interaction of the user with the pmCHP-software, usability is an afterthought.

Usa01 To start or stop the pmCHP, only a single button has to be pressed by the user.

Usa02 After being started the software presents the momentary state of the pmCHP and can display the current operational planning.

B. Application of scenarios

To evaluate the architectures, the aforementioned scenarios are applied to the architectures. For each scenario, necessary changes to the architecture are tracked, differentiating between easy and complicated changes. Easy changes are changes that are most likely to be completed within a day, while complicated changes will most likely take longer than three days. With the scenarios being drawn from different use cases within the softwares life, changes to a lot of components to accommodate a single scenario indicate strong coupling. If only a single component has to be changed for a scenario loose coupling is visible, if no change to any component is necessary the scenario showcases a previously unspoken requirement that was accidentally fulfilled.

Furthermore, scenario interactions are tracked. Two scenarios interact, if they require changes to the same component, indicating a low coherence of the component. If a component is only responsible for a single thing, no two different scenarios will not interact if the scenarios are separate (not two variations of the same situation). A high amount of scenario interactions indicate a badly defined component.

a) Availability: First, we will evaluate the architectures using the scenarios regarding software availability, starting with Scenario Ava01.

Ava01 demands the monitoring of the DRIVER component within the software. A component to monitor the driver can already be found within all three architectures, therefore no additional components are needed. Changes to the MONITORING component are the simple inclusion of the monitoring and restarting of the driver, which we classify as an easy change.

Ava02 requires the Driver to ignore incorrect measurements caused by normal sensor jitter. Again, the no new component has to be introduced, since all architectures already contain a DRIVER, which can be changed. However, only in the ED-CEP this change can be done easily, since the rule-based logic allows for the easy definition of complex conditions. In the SOA or the layered architecture the change is rather complex, requiring a lot of work to correctly filter incorrect data.

Ava03 describes the disconnect of the pmCHP from the Software, an error has to be displayed. All three Architectures already display the status of the system to the user, therefore no change to the architectures has to be made.

Ava04 puts a heavy load of requests on the software and demands continuous service. A simply threshold can be introduced into the corresponding component of the three architectures, discarding all requests beyond that threshold. In the SOA, the WORKFLOW ENGINE creates new processes for every request, a simple change adds a threshold to concurrent processes. In ED-CEP or the layered architecture, the threshold can easily be added to the SMART GRID ADAPTER.

b) Security: Next on the list are the scenarios concerning software security.

Sec01 aims to prevent abuse of software functions, authentication of requests prevents unauthorized or malicious use of the pmCHP. All three architectures already ensure authenticity of requests, no change is needed.

Sec02 is rooted in the softwares update mechanism required by the requirement HR05, allowing software changes in the form of updates. To ensure software integrity the BOOTER within ED-CEP and layered architecture as well as the INTEGRITY CHECK in the SOA are already used to check integrity on startup. No changes are necessary.

Sec03 requires internal shielding of software functions to prevent access to critical functions from unauthorized components. In the SOA all calls to services are done via the SERVICE BUS, which can easily be extended with an access control list, a single easy change is counted. The ED-CEP requires a lot more modification. First, all events consumed by the DRIVER as well as the DRIVER itself have to be modified to allow for authorization checks using a token based system. Additionally, the EMERGENCY CONTROL as well as the PLANNING components have to be modified to use the new token system. Therefore, we count four changes, one of which

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

is a difficult change of the DRIVER. Similarly, the layered architecture requires extensive changes to the DRIVER as well as easy changes to the VISUALIZATION, CONTROLLER and the MAINTENANCE component.

c) Safety: After ensuring availability and security, safety is our primary concern.

Saf01 describes an emergency shutdown of the pmCHP, which is already part of the DRIVER within each architecture. No changes are necessary.

Saf02 requires the DRIVER to check of every command it receives, to prevent damage to the pmCHP. While the Driver already exists, this was not part of the original functionality, therefore, changes are necessary. To add the checks to the DRIVER is relatively easy within the CEP, since complex conditions can easily be translated into rules. Adding checks to SOA or the layered architecture however is rather complicated since no rule language is available.

d) Maintainability: Maintainability is important over the softwares life, as changes might arise, which are not expected at the moment of its design.

Mai01 plans the integration of the pmCHP into the smart market, adding new features. For this a new component BILLING is needed, which monitors the execution of requests and bills the produced energy to the consumer. In the SOA the process to process an energy request has to be extended by the billing of the customer. A new service BILLING has to be added, which needs to be mapped by the SERVICE BUS. The BILLING service accesses process variables already tracked by the MONITORING to fulfill its function. Three easy changes are required. The CEP architecture requires more extensive changes. The *Request* events have to be extended with data about the consumer to be billed. These changes have to be integrated into the CONTROLLER, *Plan* and the DRIVER. Furthermore, the DRIVER needs to create *Bill* events, which are consumed by the BILLING processor to bill the customer. Five components need to be changed and two new components need to be introduced. The layered architecture requires similar changes, since the customer data needs to be introduced to the SMART GRID-ADAPTER and the SMART GRID-REQUESTS component. Additionally, the SMART GRID-REQUESTS component needs to monitor the execution of the operational plan, billing the customer after the energy was provided. To access billing functions, a new adapter BILLING is introduced. This results in two changes and a single new component.

Mai02 describes the unit test of the DRIVER, requiring a mock of all functions needed by the DRIVER. This is easily achievable in all architectures, since the physical pmCHP is designed as an external system, no changes are necessary.

Mai03 requires conceptual integrity within the architectures. After reviewing the architectures we conclude that no changes are necessary to the architectures, as similar problems are solved in similar ways in our designs.

e) Performance: The performance of the pmCHP-Software is not as important as the security and safety, since all time-critical functions are contained within the operating system of the pmCHP.

Per01 demands a 'fast enough' shutdown of the pmCHP in critical situations. This is out of scope for our architectures, therefore no changes are necessary.

f) Usability: Ease of use is a desirable trait for all software systems, but is the least important quality concern for the pmCHP-Software, since almost no user interaction is planned.

Usa01 describes a single button startup of the pmCHP as well as a stop upon pressing the same button. This functionality requires the initialization as well as the correct shutdown of all software components. In the SOA this is already provided by the LIFECYCLE-MANAGEMENT, no change is necessary. The CEP and the layered architecture provide a BOOTER, which only takes care of proper initialization and has to be modified to also provide a shutdown functionality. A single change is necessary to CEP and layered architecture (including a renaming of the BOOTER, which no longer only boots the software).

Usa02 needs the display of status information to user at all times. The component VISUALIZATION is already present in all architectures for this exact purpose.

g) Summary: The results of the scenario based evaluation is presented in Tables I and II.

First we will concentrate on Table I, containing the count of changes to the architectures.

TABLE I. COUNT OF CHANGES TO COMPONENTS NECESSARY TO FULFILL ALL SCENARIOS.

SOA Component	Count	ED-CEP Component	Count	Layers Component	Count
Driver (difficult)	2	Driver	3	Driver (difficult)	3
ServiceBus	2	SG-Adapter	2	SG-Adapter	2
WfE	1	Controller	2	Controller	1
process request	1	Driver (difficult)	1	Maintenance	1
Billing*	1	Planning	1	Visualization	1
		Emergency override	1	Smart Grid-request	1
		Monitoring	1	Billing	1
		Booter	1	Booter	1
		Request	1		
		Billing-event	1		
		Billing	1		
Total:	7		16		13
of total					
- difficult:	2		1		3
- easy:	5		15		10

A lot of easy changes to the ED-CEP architecture and a lot of difficult changes to the layered architecture are evident in the results.

The numerous changes to the ED-CEP are based in its structure, often a lot of small changes to components or new events had to be introduced. Especially the internal shielding required a lot of changes to the architecture. However, changes to the DRIVER often were easy in the ED-CEP but difficult within the SOA or the layered architecture. This is a result of the rule-based nature of the ED-CEP, which is built for easy modeling of complex conditions.

The high amount of difficult changes to layered architecture originate from the scenarios requiring pattern recognition (e.g., Saf01), which is difficult to introduce without specialized tools like the rule language used in the ED-CEP.

The SOA however requires the least amount of changes. Especially the easy extension and modification of services through the use of microservices keeps the necessary changes

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

small. Also, the SERVICE BUS provides a central place to introduce software specific functions like internal shielding.

Table II shows the scenario interactions that occurred during the application of the scenarios.

TABLE II. COUNT OF SCENARIO INTERACTIONS WHEN APPLYING THE SCENARIOS TO THE COMPONENTS OF THE DIFFERENT ARCHITECTURES.

SOA		ED-CEP		Layers	
Component	Count	Component	Count	Component	Count
Driver	2	Driver	4	Driver	3
ServiceBus	2	SG-Adapter	2	SG-Adapter	2
		Planning	2		
		Controller	2		

A lot of scenario interactions are evident for the ED-CEP architecture. This originates from the spread of functionalities over a high amount of small components, which results in unclear boundaries of responsibility.

However, the SOA and the layered architecture suffer from interactions as well, mostly through the DRIVER, which seems to be responsible for multiple scenarios. Splitting the DRIVER into multiple components might be useful.

VII. EVALUATION

Considering the previous results two rankings can be created. First, the architectures are rated by the changes necessary to accommodate the scenarios. The changes are summed up with difficult changes contributing threefold to the score.

1) SOA using microservices (11 points)
2) ED-CEP (18 points)
3) Layered architecture (19 points)

As previously mentioned the amount of necessary changes to the architecture is an indicator of its coupling with a high amount of points indicating a strong coupling. A strongly coupled architecture is difficult to adapt to new circumstances, since every change touches a lot of parts of the architecture making development complicated. This is considered a negative trait of architecture. Therefore, we consider the SOA the best architecture under this metric with ED-CEP and the layered architecture being similarly bad.

To rank the architectures using the scenario interactions, the interactions are simply summed up.

1) SOA using microservices (4 points)
2) Layers (5 points)
3) ED-CEP (10 points)

The scenario interactions provide an indicator for the quality of the component definition, i.e., how clear the functionalities are defined that a component should provide. A high amount of interactions indicate badly defined components that have no clear responsibility. This leads to feature envy in components that accumulate a lot of different functionalities, making changes complicated and the replacement of components a lot of work. Also, a negative trait of software architecture, badly defined components decrease maintainability and are considered bad. We conclude that the SOA contains the most well defined components, with the layered architecture trailing closely.

Considering both rankings we choose the service-oriented architecture for the integration of the pmCHP into the Smart Grid.

VIII. CONCLUSION

In this article, we first presented the problem of distributed power generation in the smart grid and range extension in the BEV. The problems are tackled by the pmCHP as it is developed at the *University of Applied Sciences and Arts Hannover*. To integrate the pmCHP into a smart grid an architecture had to be chosen.

After presenting the requirements posed to the architecture, we designed three different designs using different architectural styles. This was done to find the optimal architecture to implement. To compare the architectures, we presented scenarios of five different aspects of software quality before applying the scenarios to the architectures.

Considering the evaluation results we conclude that the SOA is the best suited architecture to integrate the pmCHP into the Smart Grid. The SOA provides the highest amount of flexibility when compared to our ED-CEP or layered approaches. Especially the usage of microservices helped to define clear functional boundaries and ensure loose coupling within the architecture, both considered to be good traits within software. We conclude that using the SOA design, we will be able to easily accommodate changes in the project field of usage, e.g., adoption to specific BEVs. However, this result only holds true for our narrowly defined domain and might be different for other use cases, like the usage of pmCHP in planes.

Also, since all architectures have been developed by the same person over a short span of time, they likely influence each other. Especially the CEP and the SOA share some applications, which can also be explained by similar design philosophies. Further work combining the two might prove an even better solution for the smart grid integration of pmCHP.

In future steps, we will implement the SOA and evaluate the impact of pmCHPs in a smart grid.

ACKNOWLEDGMENT

This work was supported by the VolkswagenStiftung and the Ministry for Science and Culture of Lower Saxony (project funding number VWZN2891). We would like to thank all our colleagues from the research focus Scalability of mobile Micro-CHP units and the Institute for Engineering Design, Mechatronics and Electro Mobility (IKME) for their support and the productive cooperation.

REFERENCES

[1] R. Pump, A. Koschel, and V. Ahlers, "On microservices in smart grid capable pmchp," SERVICE COMPUTATION 2018, The Tenth International Conference on Advanced Service Computing, pp. 30–35.

[2] H. Farhangi, "The path of the smart grid," Power and energy magazine, IEEE, vol. 8, no. 1, 2010, pp. 18–28.

[3] C. Schmicke, J. Minnrich, H. Rüscher, and L.-O. Gusig, "Development of range extenders to mobile micro combined heat and power units in vehicles and buildings; *Weiterentwicklung von Range Extendern zu mobilem mikro-Blockheizkraftwerken in Fahrzeugen und Gebäuden*," Techniktagung Kraft-Wärme-Kopplungssysteme, April 2014.

[4] N. Reinprecht, J. Torres, and M. Maia, "IEC CIM architecture for Smart Grid to achieve interoperability," International CIM Interop in March, 2011.

[5] H.-J. Appelrath, L. Bischofs, P. Beenken, and M. Uslar, IT-architecture-development in the Smart Grid; *IT-Architekturentwicklung im Smart Grid*. Springer, 2012.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[6] C. Schmicke, J. Minnrich, H. Rüscher, and L.-O. Gusig, "Examination of mobile micro-chp on testbeds of the University of applied Sciences and Arts Hanover; *Untersuchung von mobilen mikro-BHKW an Prüfständen der Hochschule Hannover*," Ingenieurspiegel, vol. 4, 2015, pp. 60–61.

[7] R. Kazman, G. Abowd, L. Bass, and P. Clements, "Scenario-based analysis of software architecture," IEEE Software, vol. 13, no. 6, November 1996, pp. 47–55.

[8] A. Lee and T. Brewer, "Smart grid cyber security strategy and requirements," Draft Interagency Report NISTIR, vol. 7628, 2009.

[9] V. V. der Elektrotechnik Elektronik und Informationstechnik eV, "Deutsche Normungsroadmap: Energy/Smart Grids 2.0."

[10] A. Arsanjani, "Service-oriented modeling and architecture," https://www.ibm.com/developerworks/library/ws-soa-design1/ 2017.11.01, November 2004.

[11] R. Bruns and J. Dunkel, Event-driven architectures: software architecture for event-driven business-processes; *Event-driven architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse*. Springer-Verlag, 2010.

[12] E. W. Dijkstra, "Structure of an extendable operating system," http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD275.PDF 2017.11.01, November 1969, circulated privately.

[13] patterns & practices Developer Center, "Microsoft Application Architecture Guide, 2nd Edition," https://msdn.microsoft.com/en-us/library/ff650706.aspx, October 2009, last visited: 26.08.2016.

[14] P. Clements, R. Kazman, and M. Klein, Evaluating software architectures. Addison-Wesley Professional, 2003.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A Design of Rich Environment for Teaching Meaningful Mathematics to Low-Achieving Students: Research Implications

Orit Broza
Faculty of Education
Levinsky College
Tel-Aviv, Israel
Email: Oritbroza1@gmail.com

Yifat Ben-David Kolikant
School of Education
Hebrew University of Jerusalem
Jerusalem, Israel
Email: Yifat.kolikant@mail.huji.ac.il

***Abstract*—The paper describes implications from a design-based research in which a rich collaborative, computer-supported learning environment was designed to promote meaningful mathematics among low-achieving students. Fifth-grade students interchangeably solved decimal subtraction tasks with peers in the context of a computer game and simulations, and in discussion sessions led by their teachers, in foursomes. We describe the results of the first round of our design-based research, where we traced three such groups, using observations and interviews. We found that the computer context was both constructive and destructive, in terms of students' learning. The group discussions did not yield the rich discussions we had hoped for. Yet, overall, the environment was successful because students gained meaningful mathematical knowledge and practiced active, thoughtful, and collaborative socio-mathematical behavior, which is dramatically different from what they were used to.**

Keywords: low-achieving students; support-rich environment; computer games; scaffolding; computer-supported collaborative learning.

I. INTRODUCTION

The question of how students' construction of meaningful knowledge can be supported represents an important challenge to both researchers and teachers. Teaching the complex topic of mathematics to low-achieving students (LAS) poses a special challenge, owing to LAS's unique cognitive and behavioral characteristics [1]. The teaching and learning processes of LAS have been studied by examining different teaching methods, strategies, and tactics (e.g., [2][3]). However, we found sparse work on the effectiveness of rich environments, let alone environments involving computer-supported collaborative-learning (CSCL), on the learning processes and outcomes of LAS.

In fact, LAS characteristics, which we describe next, might make one doubt the feasibility of teaching LAS basic mathematics, let alone in (Computer Supported) Collaborative Learning (CS)CL settings. Nonetheless, we hypothesized that a rich CSCL environment, involving a computer game, real context mathematics, peer discussions, and teacher mediation may be the key for addressing LAS's unique and diversified needs. Here, we describe the results of the first round of a design-based research we conducted to examine these hypotheses. First, we describe the characteristics of LAS. Then we review the literature and how it influenced our hypotheses and design. Next, we describe a study, the first round of a design-based research in which we examined our hypotheses. We traced the participation of 3 groups of four students each, in the activities we had designed, using various data sources, such as videotapes and audiotapes of the classes, as well as interviews and ad-hoc conversations with students and teachers, along with observations. We will discuss our findings and their practical implications on our design framework and the broader scientific community. Our main conclusion is that CSCL, when carefully designed, can promote LAS learning of meaningful mathematics as well as the development of sociomathematical norms.

In Section II we review the literature on LAS as well as on successful interventions in terms of achieving meaningful learning. Then, we describe our pedagogical design, and the literature that inspired us, such as the decision to involve a computer-game session in which students work in pairs, and small-group discussions led by the teacher (Sections III and IV). We then describe the study (Section V). Next, we examined how a rich environment either hinders or supports students' construction of mathematical meaning. We focused on the mutual interplay between the two contexts in which students worked (on the computer and in group discussions). We present the findings (Section VI) and discuss them (Section VII).

II. LAS AND MEANINGFUL MATHEMATICS

There is no single, definitive profile for LAS [4][5]. In fact, most studies have not focused on the methodological criteria used to identify those students with learning disabilities [5]. LAS are commonly identified based on two factors: teachers' reports and LAS performance on standardized or informal tests (students' score below the 50th percentile on standardized tests; however, they are not diagnosed as having learning disabilities) [2]. In attempting to explain LAS's poor performance, the literature focuses on cognitive deficiencies and on behavioral manifestations of their failures. LAS find it difficult to retrieve basic mathematics knowledge from their memory [6]. Craik [7] terms this difficulty as 'fragile memory', a product of superficial data processing. They also lack meta-cognitive skills [8], and are sensitive to the learning contexts. Thus, they find it much harder than others to solve simple and complex addition and subtraction problems. These

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

difficulties may lead them to use less sophisticated strategies and to make more errors.

Recently, Karagiannakis et al. [9] developed a model that can be used to sketch students' mathematical profiles for four domains (numbers, memory, number line, and reasoning); they empirically examined it to determine whether and how it can differentiate between students with and without difficulties in learning mathematics. According to their analysis, students, both the normal/high achievers and the underachievers, do not all share the same strong or weak mathematical skills. In addition, under achievement in mathematics is not related to weaknesses in a single domain (e.g., numbers, memory, number line, and reasoning). They also suggest that for LAS, just like for other students, cognitive strengths or weaknesses may rely on any of the four domains (mentioned above) of their model. Their findings empirically strengthen the heterogeneity of this population group.

Experiencing repeated failures and difficulties in keeping up with the class might, in turn, decrease LAS's motivation and their sense of internal responsibility and make them more passive learners. It might also lead them to act impulsively, rely on the judgment and feedback of an external authority [3], and avoid collaborative work with peers [10]. Their schooling-purposed interaction in class is largely with the teacher.

These characteristics probably underlie many teachers' beliefs that LAS are unable to deal with tasks involving high-order thinking skills and that the most effective way of promoting mathematical performance in LAS is to 'drill and kill', that is, to focus more on the mathematical algorithms than on the mathematical meaning [11].

However, despite their difficulties, there is empirical evidence that in certain environments LAS are capable of enhancing their mathematical understanding. There is empirical evidence that LAS can exhibit mathematical reasoning orally when placed in intimate and supportive learning environments, such as in small groups where they are tutored [12][11]. Peltenburg et al. [13] show that, in a familiar context with the help of technological tools, LAS can succeed in solving subtraction problems by using an indirect addition strategy spontaneously, rather than the conventional direct subtraction strategy. Karagiannakis and Cooreman [14] suggest that these interventions should be designed for repeated success by building on a student's strengths, while avoiding use of repetitive tasks that cause repetitive failure experiences, thereby maximizing the learning opportunities of all students.

Synthesizing these empirical evidence with the reports on the literature on the diversity among LAS, we assumed that a rich environment that includes technological tools, small groups, and teacher's support building on LAS's strengths might be the key to their success.

III. The Literature that Inspired the Design and Hypotheses

Our design was inspired by the socio-cultural theoretical perspective on learning, especially the notion of distributed scaffolding. Scaffolding is "titrated support that helps learners learn through activity. It helps learners perform tasks that are outside their independent reach and consequently develop the skills necessary for completing such tasks independently" [24, p.306] . Because LAS vary in their behavior, in our design we sought to design distributed scaffoldings [15], i.e., to integrate and sequence multiple forms of support via various means. Different scaffolds interact with each other; sometimes they produce a robust form of support, a synergy [16], and other times, they might sabotage the learning processes and the outcome.

We were also inspired by the Learning in Context approach, namely, the idea of presenting mathematical concepts and procedures in a context relevant to the child's day-to-day life [17], and in particular, the Realistic Mathematics Education (RME) theoretical framework. According to the RME framework, students should advance from contextual problems using significant models that are situation related, to mathematical activity at a higher level (e.g., engaging in more formal mathematical reasoning). As students progress from informal to more formal mathematics, their "*model of*" the situation is transformed into a "*model for*" reasoning. We hypothesized that RME could be the key to promote meaningful learning for LAS, because the subtraction tasks, the mathematics to be mastered, will be associated with real-life experiences, which might mitigate their fragile memory and tendency for superficial processing of new knowledge.

We aimed at transforming students' social and socio-mathematical norms, from passive to active, from isolated to social collaboration, and from impulsive to thoughtful. We were motivated by the premise that digital games, by the nature of their design, have the potential to motivate students to become active rather than passive, by enabling experimentation and exploration without fear of failing in front of the entire class [18][19]. The use of games for teaching may be particularly beneficial for LAS because of their tendency to remain passive and to comply with authoritative voices.

We were aware of the possibility that a hands-on, minds-off strategy might emerge, especially because of the tendency of LAS for impulsivity. This is one of the reasons students were asked to work with peers in front of the computer. We assumed that collaborative settings would trigger twofold interactions: with the system and with the co-learner. Peers would explain their calculations to each other, and question other actions, which would bring about reflection and thoughtfulness [20].

Additionally, every session was designed to include interchangeable students' work in front of the computer with their peers, along with group discussions, led by the teacher. Teachers' interactions with students can create zones of opportunities that can be directed to scaffold students' social and emotional development [21]. The teacher can mediate the use of tools (e.g., computer games, online units), orchestrate the students' activities, and reframe them conceptually [22].

Hence, the students experienced two different collaborative settings. When they worked (in pairs) in front of the computer (computer games or online units), the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

teachers were asked to observe them and to offer help when necessary (for instance, if students maintain trial and error strategies or are stuck in their calculation process). In the group discussions, the teachers were asked to focus their discussions on various strategies that can be used to solve subtraction tasks, to encourage students to verbalize their thoughts, and encourage them to rely on each other's past experience, thereby facilitating students in learning the meaning of how to participate in the community, i.e., support the transformation of their sociomathematical norms [23]. In these discussions, the teachers also introduced students to new tasks and encouraged them to employ the strategies they previously used in a supposedly new context. As we will explain in the next section, in our design we presented tasks sometimes as stories and sometimes as formal subtraction exercises, and gradually increased the difficulty of calculating the numbers whose decimals are half, to numbers whose decimals include individual units. We assumed that students' sense of security when expressing themselves publicly would increase, since they are in a group of equals, and will experience active (and successful) work with their peers in front of the computer.

IV. THE INSTRUCTIONAL DESIGN

We developed an extracurricular program for fifth grade LAS. It consisted of ten weekly sessions that focus on subtraction with decimal numbers, a topic that students had not yet learned in their regular classes. Students were categorized into groups of four, according to their regular class, and each group worked with a teacher trained by the second author.

We utilized a *real-life context* simulated by an ice cream shop computer game. Specifically, during the sessions, students played a computer game in which they received orders from random customers, prepared the orders, calculated the price to be paid, and gave change as needed (Fig. 1). Because of the heterogeneity of LAS and their individual needs, we sought to provide a variety of support types. Therefore, students also worked on supplementary online study units concerned with the transition between money and formal representations, as well as change calculations. Students also enacted game-like situations with play money using Israeli bills and coins: New Israeli Shekels (NIS) and agorot (1NIS = 100 agorot; the smallest coin is 10 agorot). In order to support the transition from the concrete to the abstract, real worksheets were designed, which included exercises in concrete, graphic, and abstract forms.

In order to facilitate the delicate transition from the realistic environment (shop simulation) to formal mathematics, subtraction was first presented through monetary simulations and calculations only, and formal representations were interwoven at a later stage. The program progresses in a *spiral-like manner*. With the help of the teacher, students are expected to progress from one level to the next. The tasks at each level maintain an overall forward trend of increasing complexity, and students are able to revisit earlier levels and solve simpler exercises on the computer on their own. The teachers had the flexibility to fine-tune the program, in response to students' emerging needs.

Fig. 1 A screenshot of an online learning unit, where the task at hand is 50-38.6.

In each session, students spent almost half of their time in front of the computer, working in pairs. They were first introduced through online activity to two avatars, a girl, and a boy, each of whom described a strategy he or she uses for calculating the required change. Then they played or worked in pairs on the computer. The other half of the session time was devoted to class discussions, as described above. Specifically, in order to address LAS's tendency to passively rely on external authority and to encourage them to take personal responsibility, the teachers were not supposed to correct students' strategies directly, but rather, to ask questions to encourage them to talk aloud about their thinking processes, thus, making the diagnosis easier and potentially leading them to correct their own mistakes, re-voicing when needed, and referring them to suitable tools in the environment when necessary. The teachers generally followed these instructions closely.

V. THE STUDY

Our goal was to examine our design's hypotheses, i.e., to examine the students' learning processes, focusing on how the rich environment either hinders or supports students' construction of mathematical meaning, especially the mutual interplay between the two contexts in which students worked (on the computer and in group discussions).

A. Participants

We traced 12 LAS (4 male, 8 female) from 3 fifth grade classes in suburban schools within the same city, and who participated in the program. They learned in 3 groups of 4 students, with 3 different teachers (one of them was the first author). All participants were chosen based on the recommendation of their mathematics teachers. They all performed under the 50th percentile on standardized tests, yet were not diagnosed as having learning disabilities.

B. Data Sources

In two groups all sessions were videotaped. In one group they were audiotaped. We observed students in their regular mathematics class two times before they began participating in our CSCL activity. We also observed all the sessions,

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

focusing on the sequence of activities—of both the teacher (e.g., presenting tasks, intervening during the computer sessions, suggesting a tool, getting students' attention, and answering questions) and the students (e.g., how they interact with the computer, with each other, with the teacher, and so forth). We conducted interviews with the CSCL teachers, after the activity as well as ad hoc conversations after every session. We also talked with their previous mathematics teachers and with each student after the CSCL activity.

C. *Methods of Analysis*

Our report mainly draws on the analysis of the videotapes. A preliminary analysis of the data was presented elsewhere [1]. That analysis was useful to identify patterns of students' interaction with the environment. Here we present in detail the results of a fine-tuned analysis. Specifically, we were inspired by the analysis model of Powell et al. [24] for developing mathematical ideas and reasoning. We fully transcribed one group through videotapes. The transcripts were coded twice by two researchers. We segmented the text into episodes, each beginning with the presentation of a new task and ending with its being accomplished (or the work on it was terminated). For each episode we examined: (1) who participated in it; (2) the knowledge pieces that emerged; (3) the difficulties that arose, including whether they were resolved, and if so, how and by whom, especially (d) the support provided by the teacher; and (5) whether the task was successfully accomplished independently or with help from others. We also coded affective utterances, both positive and negative. We compared the results with the video, audio, and notes taken during the observations in the other groups. Interviews were analyzed thematically. We chronologically traced changes in each student's thinking and behavior, thereby creating data stories. One such data story is presented next.

VI. FINDINGS

A. *Students' Interaction with the Rich Environment*

1) The computer setting: As we hypothesized, the computerized environment, especially the computer game, encouraged the students to be active as well as engaged in their task. For the most part, they were observed to be very focused on the current task. In fact, in 5 sessions, students continued working (or playing) after the class had ended. The students reported in the interviews and ad hoc conversations that they had enjoyed the activity. The following quotes are but two examples of typical phrases heard throughout the entire program: "it was fun…not a regular class", "playing with the computer gives a sense of fun, [vs.] a blackboard, where you just sit and solve exercises".

On the computer the students (who sat in pairs) usually decided to work in turns. In each turn the one on the keyboard gave ice cream, calculated the price, the change, and returned change. For a few couples, we noticed a different division of labor: the one on the keyboard interacted with the avatar clients and in the meantime, the other did the calculations. In a few cases when one student took over the keyboard the teacher interfered.

During the play, each student solved many subtraction exercises, manifested by the need to give change to customers in the shop.

They did not solve all the exercises successfully right away. However, for the most part failures in this context did not discourage or frustrated them. On the contrary, this is when we observed collaboration, mathematical discussions with their peers and with the teacher. Usually, when they received a response from a "customer" indicating that the change they gave was incorrect, they were observed pausing to think and sometimes they turned to their peers and verbalized their "solution process". Sometimes this verbalization was performed after their peers asked them how they had worked. Often the discussion helped them to correct themselves. This behavior was dramatically different from the observed passivity (or impulsivity) in the regular mathematics classes. Moreover, in this context, the students generally welcomed the teachers' intervention and cooperated with them. Hence, the computer and their peers often generated a synergetic effect on the students.

However, we also observed an appreciable number of situations in which students merely employed trial and error, using the immediate feedback of the computer ("too much" and "too little") to guess the correct answer. Usually the partner became silent in these situations. From the conversations in these situations, we learned that the pressure of time and the wish to gain as many points as possible in the game in a designated time frame encouraged this behavior. In one extreme example, one student (Betty) stopped working because the clients became angry (Fig.2 and Fig. 3), because it took her time to calculate. This episode as well as other important episodes and aspects of Betty's learning process within the environment are presented in Section VI.*B*.

Fig. 2 Speech bubbles turn red as a sign for impatient clients.

Fig. 3 An angry face of an impatient client.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

We also noticed that in the initial lessons the teacher had to compete with students' attention to their computer in these situations. We observed the teacher, in such situations, touching the students' hand or shoulder to get their attention.

The next episode demonstrates the teacher's struggle for Etty's attention. Etty stared at the computer screen when the teacher approached her:

282.	Teacher:	Ok, how do you calculate the change?
283.	Etty:	Eh, Eh, Eh…[looking at the screen, trying to concentrate in the game]
284.	Teacher:	Etty, please explain.
285.	Etty:	I am not sure…[keeps playing]
286.	Teacher:	Remember how we did it before?
287.	Etty:	Aha… [Her full body is turned to the computer].
288.	Teacher:	And we saw many ways, the way Dor [avatar] calculated?
289.	Etty	Aha…[she keeps concentrating on the computer]

Obviously, Etty preferred to focus on the computer session. She concentrated on the task, mumbling "Eh…or Aha…" answers to the teachers' requests as if it was difficult for her to split her attention (lines 283, 287).

2) Group discussions: Group discussions revolved around calculations and strategies. Figure 4 illustrates a typical discussion routine.

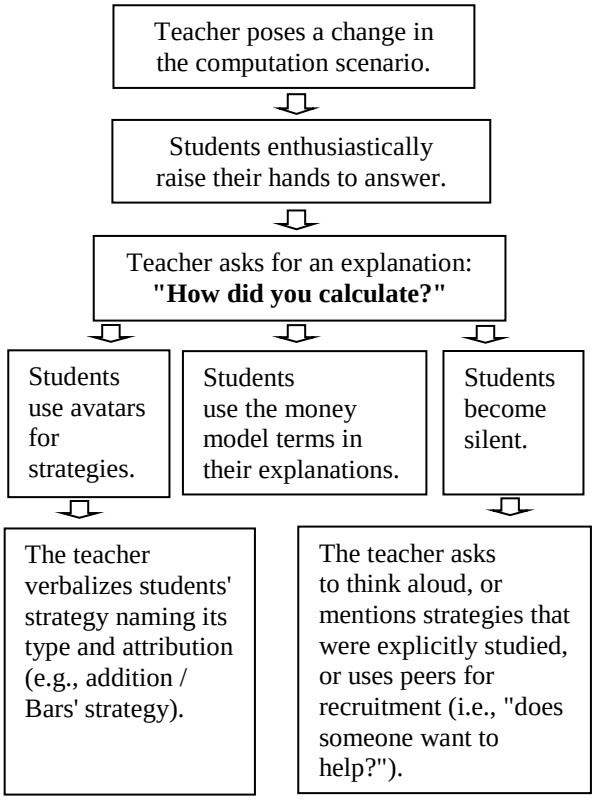


Fig. 4 Group discussion after game play.

The teacher initiated each episode by presenting a subtraction task. After the students solved the task, she then asked them to explain their strategies. We observed many expressions of frustration among the students. The teacher noticed that students tended to take turns when they worked at the computer. She borrowed the idea and asked them to also solve exercises in turns in the group discussions. However, in this setting this idea turned out to be less productive. Generally, the interaction took the form of one student explaining his or her solution process, followed by the teacher's verbalization. Moreover, the teacher sometimes silenced the peers who tried to participate in conversations. In her interview she explained that students' poor discursive habits made her prioritize the individual's learning over building a community and discursive habits. We thus observed almost no rich peer discussions about strategies.

We expected that during the participation the students' ability and wiliness to provide explanations would increase. During the discussion with the teacher (with or without a computer) the students were constantly asked to describe and explain their strategies. The alienation of this request was prominent in their responses. They became silent, gave vague or non-informative answers (e.g., "I just did so"), and sometimes even said, "I don't remember". The following excerpt from lesson 3 illustrates this kind of discourse while a student was struggling in calculating the exercise 10 - 4.1 =

324.	Teacher:	How would you like to solve this problem?
325.	Noya:	I don't know which one is more comfortable to me.
326.	Teacher:	You don't know which one is more comfortable to you. [pause].
327.	Teacher:	Ah! Maybe someone wants to help her explain how should she solve it?
328.	Neomi:	Addition.
329.	Teacher:	Addition. What do you mean? Explain to her.
330.	Neomi:	To add 10 Agorot until we have 10, which means that it is 9 times ten Agorot.
331.	Teacher:	Until we get to 1 Shekel, right? Until we get to a Shekel.
332.	Neomi:	And then we take 5 [Shekels] and 9 like these [10 Agorot coins].
333.	Noya	Aha…

The preference question (line 324) confused Noya. Probably, she was not used to these kinds of questions in her regular math class. When she noticed that Noya became silent, the teacher turned to recruit the group (line 328) and emphasized the meaning of *adding on* strategy (line 331).

The above excerpt also demonstrates that in some of the students' explanations (e.g., for Neomi) there was evidence of a positive change in their discursive manners. In these cases we often found that students relied on the money model (especially the fact that 1 nis = 100 agorot) to explain their subtraction strategies even when the subtraction task

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

was phrased abstractly and not in money terms. Real context mathematics, hence, supported students' leaning.

We expected that the students would develop many strategies for subtraction. Indeed, the teacher posed questions like "in what way would you like to solve this problem?" at least three times in each of the first three sessions. However, we did not observe the emergence of a new strategy. One possible explanation is rooted in our sequencing of students' activities. In the initial lessons, students were introduced by an online unit to two strategies, presented to them by two avatars who dealt with the task of calculating change. Possibly this early exposure, together with students' tendencies to rely on external authoritative voices, brought about a fixation in their thoughts. Moreover, sometimes we were not sure that students understood the meaning underlying these strategies.

Nonetheless, in conversations with the teachers in the regular classes after the program ended, the teachers reported that the behavior of most of the participants in their class improved; specifically, that despite their difficulties they were more motivated and less passive.

B. Betty's data story

Betty was diagnosed by her teacher as a low achieving student due to her low academic achievements compared with other students in her class, her impulsivity, and her "short memory", as Betty testified. The teacher reported that once a week Betty used to leave her regular math class to learn in a small group in order to help her keep up with the class.

Table I describes Betty's performance in the subtraction tasks during the activity. The table includes details about each task (whether it was presented in a story form or as a formal exercise), its context (a group discussion led by the teacher, an individual worksheet, or peer interactions while students worked on the computer), whether it was performed orally or in writing, and finally, whether Betty succeeded in solving it, and whether the success was assisted or not. Betty solved more tasks within the computer context, but since we did not record the screen in the first iteration, we have only the tasks in which the teacher was directly involved. Nonetheless, we took notes on her performance within this context as well.

As shown in Table I, Betty's performance was inconsistent. Betty successfully solved oral subtraction tasks within the context of group discussion (tasks A, C). She also experienced some success in written calculation tasks (tasks G, H, J, K) independently. Her failure occurred partly in the computerized context (tasks D, F), probably due to her impulsivity.

1) Starting point: Observations in the regular math class preceding the CSCL activity indicated that Betty was passive, unmotivated, and unengaged, and that she laid her head on the table for most of the lesson as if she was bored. However, her behavior changed dramatically right after the first lesson when we came up with the computer game. Suddenly, she was dominant, controlling the computer, helping her peer. She even took over her peer's role of play (Fig. 5). We could see that Betty was totally engaged.

TABLE I. BETTY'S TASK PERFORMANCE

Task No.	Lesson No.	Task	Performance Of The Task	Context
A	2	Story (50-41.50)	Oral	Group discussion
B	3	Exercise (10-7.20)	Oral	Group discussion
C	4	Story (20-15.60)	Oral	Group discussion
D	4	Story (15-13.20)	Oral	Peers (with computer)
E	5	Exercise (50-14.80)	Oral +Written	Group discussion
F	6	Exercise (15-13.20)	Oral	Peers (with computer)
G	6	Exercise (15-13.20)	Written	Group discussion
H	6	Exercise (20-16.80)	Written	Group discussion
I	7	Story (50-42.60)	Written	Individual worksheet
J	8	Story (100-38.50)	Written	Individual worksheet
K	8	Story (100-57.30)	Written	Individual worksheet

Failure | Assisted success | Independent success

Although her body language expressed her full engagement in the game, we noticed some of Betty's math difficulties. For example, she used her fingers for counting. She also used the game feedback to calculate basic facts (addition and subtraction of whole numbers up to 20) that she did not master. She also used trial and error impulsively. Research reports also strengthen those impressions:

> *"Betty and Neomi cooperate by consulting each other [e.g., 'it said we gave too much; give a bit less'], clicking their answers on the computer, trying to get to the exact answer. When they get positive feedback from the computer, they laugh about their own errors…* [Taken from the second observation]."

Fig. 5 Betty controlling the computer.

In group discussions the teacher tried to confront Betty's impulsivity by encouraging her to think aloud. Sometimes

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

the teachers even used hand touching in order to prevent her from quickly checking on the computer. A representative example is presented in the next episode taken from lesson 3. Betty had to calculate the change from 10 NIS for ice cream, which cost 7.20 NIS (Table I, task B). Betty hesitantly typed a wrong answer and got negative feedback:

356.	Teacher:	[hushing the rest of the group] Girls, please let her concentrate.
357.	Betty:	I thought it was right.
358.	Teacher:	Why?
359.	Betty:	I don't know [she types the same answer. Gets negative feedback]
360.	Teacher:	Betty, how did you try to solve the problem?
361.	Betty:	[disturbed by her peer] Stop it Eve!
362.	Teacher:	Betty, look at the amount to pay.
363.	Betty:	Ehm..7.20.
364.	Teacher:	Seven NIS and twenty Agorot. And what is the amount that was paid?
365.	Betty:	10 NIS.
366.	Teacher:	Ten NIS. According to which strategy is it easier for you to calculate? Ah? [Betty is clicking on the coins on the computer, thereby returning change. Getting negative feedback]
367.	Betty:	Wait, wait it is too much…[reading the negative feedback]
368.	Teacher:	If you give three NIS [change], and he has to pay seven NIS, what number you get?
369.	Betty:	[facing the computer screen] Come on…. but here he [avatar] said that it is too much, so maybe I will add here one [adding one ten Agorot coin and waiting for the computer's feedback. She gets negative feedback].
370.	Teacher:	No Betty! It is impossible! I want you to think for a minute. [Betty counts quietly using her fingers] Betty, answer me…We have Neomi's strategy ...It helped Noya before, did you listen? Did you understand what she said?

The above excerpt demonstrates the leading questions (lines 358, 362, 364, and 366) that the teacher offered in order to help Betty to concentrate on the calculation instead of the trial and error strategy that Betty had used. Betty ignored her and kept the trial and error method, increasing or decreasing the number of coins in order to reach the correct answer (lines 359, 367).

Betty was emotionally involved in the game. Her stress resulting from the response of the avatars is demonstrated in the following episode, taken from lesson 4. Betty and Neomi played the computer game. The teacher noticed that Betty was stuck so she approached them:

279.	Betty:	[to the client avatar] Stop it!!!! [to Neomi] Your turn…[give up and pass the computer mouse to Neomi]
280.	Teacher:	No, No, No, No, No!!!!
281.	Betty:	I can't, they [client avatars] get angry at me!!!I am scared…
282.	Teacher:	Don't look.
283.	Betty:	At the end they will get out of the game and come to beat me.
284.	Teacher:	What is the bill to be paid? Say it loud, what is the bill?
285.	Betty:	[silent]
286.	Teacher:	Thirteen
287.	Betty:	[Mumbling…thinking…] Sh….wait, wait, wait [using her fingers, turning to the computer, tapping an answer and getting positive feedback] I am a genius!!! This is what I did before….

Although Betty was nervous, the teacher did not give up and in a definite statement of "no" (line 280) she decisively did not accept this behavior. Instead, she supported Betty by breaking down the problem into its smaller components (line 284). The use of fingers (line 287) is additional evidence that this time Betty calculated and did not guess, probably the reason for her success. During the lessons, even in the face of the angry avatars, Betty constantly displayed highly enthusiastic behavior while next to the computer. This behavior contrasted with her attitude towards the initial group discussions or the teacher's requests to work on sheets (from lesson 3 onwards) frustrated her. She was observed as impatient, did not take responsibility for her own work, and often relied on the teacher's support. The teacher focused her support on Betty's needs.

2) Turning point: A great change occurred in lesson 5 during a discussion about different ways to solve 50-14.80= (Table I, task E). Betty initiated her participation by asking the teacher to show her own strategy to calculate by writing it on the board:

52.	Betty:	So, I do 50 minus 10 and I do not calculate it now because I do not have the strength. [Writes: 50-10=____] it equals something...
53.	Eve:	Don't you know how much is it?
54.	Betty:	It's…ah…
55.	Teacher:	Sh…….[teacher silence Eve]
56.	Betty:	Thank you, just a moment. [Writes 40 as an answer]. Now 40 minus 4 equals…[writes 40-4=___]
57.	Betty:	Waits…
58.	Betty:	And then I do…
59.	Teacher:	How much do you get here [points at 40-4=___]? How much does it equal?
60.	Betty:	40 minus 4…emmmm….how you call it

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

		[writes 36]
61.	Teacher:	Great!
62.	Betty:	And then 36 minus…wait not minus…no…
63.	Teacher:	Yes, yes! You are in the correct direction! What is left?
64.	Betty:	Minus 0.8 equals something [writes 36-0.8=__ and turns to the group]
65.	Teacher:	[Puts her hand on Betty's shoulder, turning her back to the board] And how do you solve this?
66.	Betty:	I take from a Shekel…
67.	Teacher:	From a Shekel, great!
68.	Betty:	[Draws circles to represent coins] one, two. I have no power for that…to draw.
69.	Teacher:	OK
70.	Betty:	From a Shekel 8 like this.
71.	Teacher:	Eighty Agorot.
72.	Betty:	So, it comes out 35 plus something [writes 35]
73.	Teacher:	And what is that something? You took a Shekel…
74.	Betty:	[Answers immediately] plus twenty!
75.	Teacher:	Very good, very good! 35 and 20 Agorot because in a Shekel we have 100 Agorot.
76.	Betty:	I am a genius!
77.	Teacher:	Excellent, well done!

Betty's difficulties are manifested by her avoidance of calculating basic facts. Writing on the board helped her to think about intermediate calculations (lines 52, 56, 64, 72). She preferred to focus on the procedure of the strategy using "___" as a place for the calculated result and the teacher accepted that, helping her in critical moments such as encouraging her by saying she is on the correct track (line 63), to focus her attention and avoid distractions (line 55, 65), or giving her hints by using the money terms in critical moments (lines 71, 73).

Most of lessons 6-7 were devoted to practicing next to the computer and solving worksheet tasks. Betty asked to play alone and the teacher let her. Lesson 8 was devoted to the final assessment where Betty succeeded (Table I, tasks J, K).

Overall, Betty's learning process illustrates the fragility of her knowledge, the inconsistency of her performance, her impulsivity in the computerized context, the clear delegitimization of this behavior by the teacher, combined with her support, which led to moments of success by using writing as a tool for self-direction.

VII. DISCUSSION

Here we discuss the hypotheses and factors that influenced the students' learning within the rich environment.

A. *The Computerized Context as a Double-edged Sword*

The computer-peer setting was found to be both supportive and destructive in terms of students' learning. The computer played a major role in making students active and engaged in mathematical discussions about the current subtraction task with their peers and the teacher, despite the students' fragile knowledge. We saw moments of synergy [16] when the presence of peers induced a reflection about a wrong calculation, and a discussion about the strategy applied. Teachers' interventions in this context were welcomed and fruitful. However, as Betty's case study illustrates, there were also situations in which the computer *game* encouraged trial and error because of the time factor and the competitive nature of games. The teacher, in her attempt to disrupt the trial-and-error discourse, often had to compete with students as exemplified above.

B. *Group Discusions: Participation Alongside Silence*

We expected that the group discussions would encourage students to talk about mathematics and therefore, foster their ability and wiliness to provide explanations. The findings indicate that the teachers' requests for explanations, especially the question about which strategy they chose to apply for solving the tasks, were alien. This is probably because in regular math classrooms LAS are rarely asked to explain their answers. Therefore, the group discussions did not yield the rich discussions we had hoped for. Nonetheless, we observed that the ability of most students to provide explanations had developed during their participation. However, these students did not develop new strategies, but rather, used the strategies they had been introduced to at the beginning. This behavior aligns with the LAS's tendency to focus on a given algorithm, given by an external authority. In addition, in this context, as demonstrated in the discussion routine (Fig. 4), students' discursive acts were mostly in response to the teacher and merely addressed her.

C. *Movement within the Rich Environment: Evidence of Diffusion*

In our design we had expected a metaphorical diffusion between the two contexts in which students performed and collaborated—that students' activeness, ability, and wiliness to discuss with their peers when failing to solve a task on the computer would diffuse to the group discussion and that the teacher-led discussions would enrich the mathematical discursive practices, which would then diffuse to the computer context. Apparently, this diffusion is not straightforward and a fine-tuned design is required to support its occurrence.

D. *RME: A Valuable Factor*

In line with other empirical studies [17], RME was found to be a valuable factor in facilitating LAS meaningful learning. Students adapted the real-life money model to resolve the subtraction tasks, even when given in an abstract form. This was evident in their formulation of their solution process in monetary terms as well as in the conceptual scaffold "1 NIS equals 100 agorot", which they often used when they had to reason how they subtracted the decimal.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

VIII. Conclusion And Future Work

A. Implications for the second round

As we had hypothesized, we found that distributed scaffoldings were beneficial to LAS [15]. The premise was that such a heterogenic population, from the cognitive and behavioral aspects, needs a variety of tools. Indeed, the environment simulated a rich "playground" where students experience diverse tools in order to build and develop new knowledge.

Obviously, more work was required to fine tune the design, in order to better support students' learning. Utilizing the insights gained from our analysis, in the next round we re-designed the group discussions in consultation with the literature on Accountable Talk [25], aiming at better facilitation of establishing the norms of mathematical peer discussions. We minimized the time spent in front of the computer game and instead, added time to the online unit, in which students still simulated the ice cream shop, but without the pressure of time and gaining points. Finally, we aimed at setting the students' mindset right from the beginning by explaining to them that this class is about *their* strategies. We omitted the introduction to the two strategies, and instead, simulated in class an affair where students brought personal items and had to give money and get change and then conducted a discussion on their calculation strategies.

B. The next round in a nutshell

Analyzing the rich data of 11 students in the second round, we found vivid mathematical discussions. Discussants were accountable to the group (i.e., engage in talk that builds on the ideas of others), according to the accepted norms of reasoning (i.e., talk that emphasizes logical connections and the drawing of reasonable conclusions using mostly the money model for justifications), and to knowledge (i.e., talk that is based explicitly on facts). While in the first round the teachers' requests for explanations were alien and sometime led to silence among students, in the second round the use of AT practices [25] allow the teacher to explicitly set clear expectations for reasoning and then to proactively support, diagnose, and analyze the development of students' mathematical reasoning.

Consequently, nine of the 11 participating LAS showed evidence of positive change in their mathematical thinking and behavior as a result of their participation in the environment. Students constructed at least one meaningful subtraction strategy using it at a high success rate once it was constructed.

Although the majority of LAS exhibited evidence of meaningful learning of mathematics in constructing and using their own computation strategies, it highlighted the challenges their learning difficulties pose. Their learning processes were inconsistent characterized by progressions and regressions and therefore were difficult to predict by the teacher [26]. In rare cases when the teacher-students discourse reached impasse, the setting of peer work next to the computer on online units was found promoted due to its safe and constructive space for building new knowledge.

Future work is still required. A larger sample of participants is necessary in order to generalize and further explore LAS learning processes and outcomes in this environment and to gain insights as to how to support their learning. Indeed, as shown above, the encounter of LAS with a rich environment might introduce complexity and pose challenges to teachers, who already have to deal with many factors (such as the fragility of students' knowledge, the inconsistency of students' performance, the impulsivity within the computerized context and the silent situations in group discussions). Nonetheless, this study shows that overall, the rich CSCL environment was successful. Not only had students gained meaningful mathematical knowledge, such as strategies to solve subtraction tasks—they also practiced socio-mathematical behavior that differed from what they were used to: they moved from passive reliance on authority, as well as impulsive and individualistic interactions in class, towards active, thoughtful collaboration about mathematical meaning. According to the regular class teachers, to some extent, this behavior has diffused to their regular classes. Thus, we can conclude that meaningful learning of LAS is feasible and furthermore, that LAS can benefit from CSCL settings, which stands in contrast to their characteristics in the literature as passive or even detached individualists [2].

We believe that a rich CSCL environment, involving a computer game, real context mathematics, peer discussions, and teacher mediation may be the key for promoting LAS's learning. In this respect, our work makes a modest step towards achieving equity in mathematics education by extending the teaching of mathematical meaning to academically diverse students.

References

[1] Y. Ben-David Kolikant and O. Broza, "The potential of support-rich environment for teaching meaningful mathematics to low-achieving students," The Tenth International Conference on Mobile, Hybrid, and On-line Learning eLmL, 2018.

[2] S. Baker, R. Gersten, and D. S. Lee, "A synthesis of empirical research on teaching Mathematics to Low Achieving Students," The Elementary School Journal, vol. 103, 2002, pp. 51–73.

[3] D. Haylock, "Teaching mathematics to low attainers," SAGE, 1991, pp. 8-12.

[4] D. C. Geary, "Mathematics and learning disabilities," Journal of Learning Disabilities, 37(1), 2004, pp. 4-15.

[5] K. E. Lewis and M. B. Fisher, "Taking stock of 40 years of research on mathematical learning disability: Methodological issues and future directions," Journal for Research in Mathematics Education, 47(4), 2016, pp. 338-371.

[6] E. Gray, D. Pitta, D. and D. Tall, "Objects, actions, and images: A perspective on early number development," The Journal of Mathematical Behavior, 18(4), 2000, pp. 401-413.

[7] F.I.M Craik, "Memory: Levels of processing. International encyclopedia of the social & behavioral sciences," University of Toronto, Canada, 2002.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[8] S. R. Goldman. “Strategy instruction in mathematics,” Learning Disabilities Quarterly, 12, 1989, pp. 43-55.

[9] G. N. Karagiannakis, A. E. Baccaglini-Frank, and P. Roussos, “Detecting strengths and weaknesses in learning mathematics through a model classifying mathematical skills,” Australian Journal of Learning Difficulties, 2017, pp. 1-27.

[10] R. Alexander, “Culture, dialogue and learning: Notes on an emerging pedagogy,” In N. Mercer and S. Hodgkinson (Eds.), Exploring talk in school, London: SAGE, 2008, pp.91-114.

[11] R. Karsenty, A. Arcavi, and N. Hadas, “Exploring informal mathematical products of low achievers at the secondary school level,” The Journal of Mathematical Behavior, 26(2), 2007, 156-177.

[12] D. Chazan, “Beyond formulas in mathematics and teaching: Dynamics of the high school algebra classroom,” New York: Teachers College Press, 2000.

[13] M. Peltenburg, M. van den Heuvel-Panhuizen, and A. Robitzsch, “Special education students’ use of indirect addition in solving subtraction problems up to 100—A proof of the didactical potential of an ignored procedure,” Educational Studies in Mathematics, 79(3), 2012, pp. 351-369.

[14] G. N. Karagiannakis and A. Cooreman, “Focused MLD intervention based on the classification of MLD subtypes,” The Routledge international handbook of dyscalculia and mathematical learning difficulties, 2014, pp. 265-275.

[15] S. Puntambekar and J. L. Kolodner, "Toward implementing distributed scaffolding: Helping students learn science from design," Journal of research in science teaching, 42(2), 2005, pp. 185-217.

[16] I. Tabak, “Synergy: A complement to emerging patterns of distributed scaffolding,” The Journal of the Learning Sciences, 13(3), 2004, pp. 305-335.

[17] K. Gravemeijer, “Local instruction theories as means of support for teachers in reform mathematics education,“ Mathematical thinking and learning, 6(2), 2004, pp. 105-128.

[18] J. P. Gee, “What video games have to teach us about learning and literacy,” New York: Palgrave Macmillian Publishing, 2014.

[19] K. Squire, “Video game-based learning: An emerging paradigm for instruction,” Performance Improvement Quarterly, 21, 2008, pp. 7-36.

[20] P. Dillenbourg and F. Fischer, “Computer-supported collaborative learning: The basics,” Zeitschrift für Berufs-und Wirtschaftspädagogik, 21, 2007, pp. 111-130.

[21] V. Morcom, “Scaffolding social and emotional learning in an elementary classroom community: A sociocultural perspective,” International Journal of Educational Research, 67, 2014, pp. 18-29.

[22] M. A. Mariotti, “Artifacts and signs after a Vygotskian perspective: the role of the teacher,” ZDM —The International Journal on Mathematics Education, 41(4), 2009, pp. 427-440.

[23] P. Cobb, "Supporting the improvement of learning and teaching in social and institutional context," Cognition and instruction:Twenty-five years of progress, 78, 2001, pp 19-37.

[24] A. B. Powell, J. M. Francisco, and C. A. Maher, “An analytical model for studying the development of learners’ mathematical ideas and reasoning using videotape data,” The Journal of Mathematical Behavior, 22(4), 2003, pp. 405-435.

[25] S. Michaels, C. O’Connor, and L. B. Resnick, “Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life,” Studies in Philosophy and Education, 27(4), 2007, pp. 283-297.

[26] O. Broza and Y. Ben-David Kolikant, "Contingent teaching to low-achieving students in mathematics: challenges and potential for scaffolding meaningful learning," ZDM, 47(7), 2015, pp. 1093-1105.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

An Evaluation of Early Activity Completion Detection Algorithms for Low-Power Peripheral Devices in Embedded Systems

Daniel Ross Moore
Center for Efficient, Scalable and Reliable Computing
Dept. of Electrical and Computer Engineering
North Carolina State University, Raleigh, USA
e-mail: drmoore2@ncsu.edu

Alexander G. Dean
Center for Efficient, Scalable and Reliable Computing
Dept. of Electrical and Computer Engineering
North Carolina State University, Raleigh, USA
e-mail: agdean@ncsu.edu

***Abstract*—Embedded peripheral devices such as memories, sensors and communications interfaces are used to perform a function external to a host microcontroller. The device manufacturer typically specifies worst-case current consumption and latency estimates for each of these peripheral actions. Peripheral Activity Completion, Estimation and Recognition (PACER) is introduced as a suite of algorithms that can be applied to detect completed peripheral operations in real-time. By detecting activity completion, PACER enables the host to exploit slack between the worst-case estimate and the actual response time. These methods were tested independently and in conjunction with Intra-Operation Dynamic Voltage Scaling (IODVS) on multiple common peripheral devices. For the peripheral devices under test, the test fixture confirmed decreases in energy expenditures of up to 80% and latency reductions of up to 67%.**

Keywords-embedded systems; energy aware embedded computing; embedded profiling; embedded performance analysis; Dynamic Voltage Scaling (DVS); low-power; low-energy; wireless sensor node (WSN); adaptive embedded systems.

I. INTRODUCTION

Embedded systems are often constrained by timing and energy budgets because both factors affect the resultant cost and size of the system. Peripheral devices external to the microcontroller (MCU) such as those shown in Figure 1 can play a significant role in system-wide energy consumption. PACER [1] decreases dynamic power consumption and latency by exploiting slack between actual versus worst-case operation time. This contrasts with other methods that focus on decreasing static power usage of peripherals [2] [3] [4].

Device manufacturers derive and specify the worst-case operation duration by summing exacerbating factors including age, temperature and voltage. Using the worst-case operation time as a naïve guideline, the worst-case energy consumption of a given operation is characterized by (1).

$$E_{op-wc} = \int_0^{t_{op}} P_{op}(t)dt + \int_{t_{op}}^{t_{slack}} P_{slack}(t)dt \qquad (1)$$

Where t_{op} and P_{op} are the time and power comprising the actual operation while t_{slack} and P_{slack} are the time and power comprising the period between operation completion and the worst-case execution time.

Most peripheral devices provide a mechanism for signaling that operations completed earlier than the maximum. However, using these mechanisms results in sub-optimal power performance. For example, a common method of detecting write completion on external non-volatile memory relies on polling a status register. Performing this signaled method has power and energy consequences:

$$P_{overhead} = P_{MCU} + P_{MCD} + P_{Comm} + P_{Match} + P_{Dev} \qquad (2)$$

- P_{MCU}: MCU must be active while polling
- P_{MCD}: MCU communications driver must be active
- P_{Comm}: Communications incurs $P = cfV_{dd}{}^2$ penalty
- P_{Matc} : MCU and device voltages must be matched.
  - Neither can use dynamic voltage scaling
- P_{Dev}: Device communications driver must be active

$$E_{op-sig} = \int_0^{t_{op}} \left(P_{op}(t) + P_{overhead}(t)\right)dt \qquad (3)$$

The components of $P_{overhead}$ are highly variable between microcontrollers, systems and devices. The signal may involve protocol-level communication or it may be as simple as an interrupt pin and that signal may traverse PCB traces with considerable capacitance. E_{op-sig} can exceed E_{op-wc}.

Both interface methods incur a power penalty and the naïve worst-case method also incurs a latency penalty. As the energy cost of computation continues to decrease in modern microcontrollers, it becomes more rewarding to use onboard intelligence to minimize the impact of power and latency penalties. PACER develops adaptive timing, current usage and charge consumption heuristics for estimating or recognizing early completion of peripheral operations, thus reducing total latency and energy consumption.

The prediction is verified in real-time against the actual state and the heuristic is updated with the results. In this fashion, the algorithms are resistant to variations in behavior that may occur across the lifecycle of the device. PACER is evaluated against a variety of embedded peripherals and is shown to significantly decrease both energy consumption and latency of peripherals with minimal computational overhead.

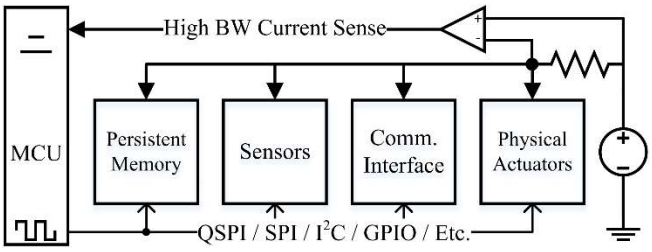


Figure 1: Typical Embedded System with Device Current Feedback

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

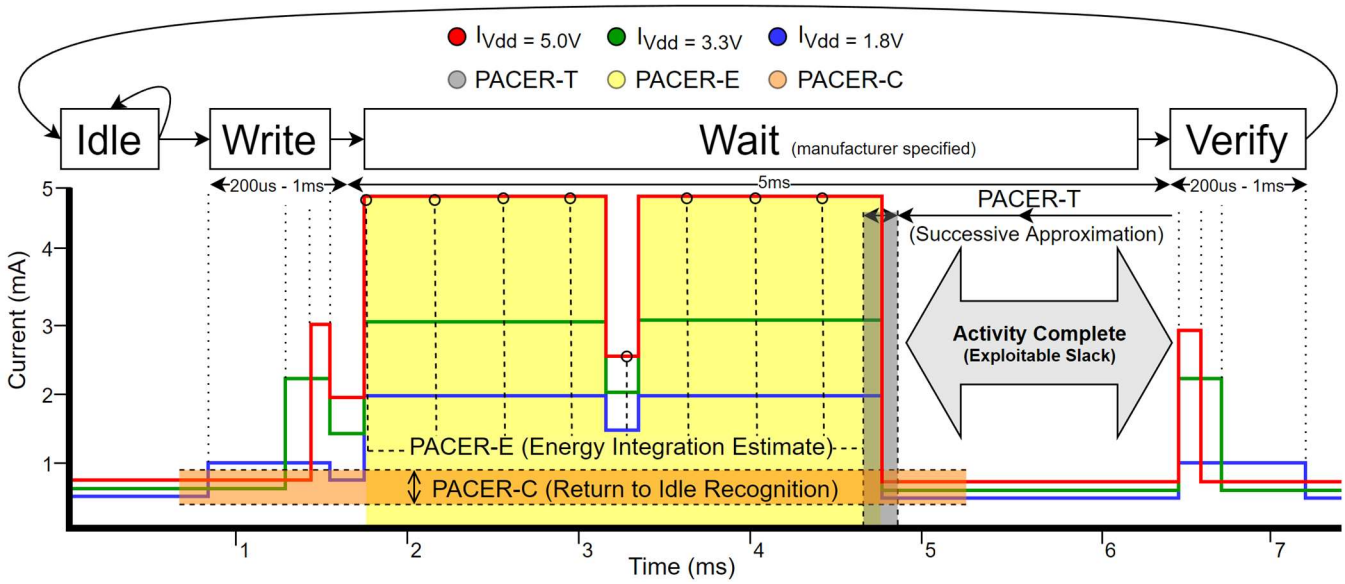


Figure 2: A Typical External Memory Transaction with IODVS and PACER

Figure 2 shows the current profile for the common peripheral operation of writing a page to EEPROM. The manufacturer-specified mandatory wait period is 5ms, beginning about the 1.75ms mark. As the device transitions through the Idle → Write → Wait → Verify states, it can be inferred from the current profile that the operation completed by the 4.75ms mark and that it was not necessary to delay until approximately 6.5ms per the specification. This 1.75ms differential is slack that can be exploited to decrease latency.

There are a wide variety of peripheral devices with a correspondingly wide variety of completion determinism and current profiles. PACER introduces three methods by which the host MCU can estimate or detect early completion of peripheral operations while also minimizing computational overhead. Devices with highly deterministic timing respond best to the timing heuristic while those with variable timing respond best to current or charge heuristics. Through low-overhead early completion detection, PACER is able to decrease both latency and system-wide energy consumption.

II. Related Work

Intra-Operation Dynamic Voltage Scaling [5] (IODVS) has been shown to significantly reduce the energy consumption of embedded peripherals (Flash, EEPROM, sensors, etc.) during their voltage-independent states. These states typically occur during mandatory delay periods while the peripheral completes a specified operation. When implementing IODVS, the host MCU and peripheral devices are placed on different voltage domains throughout the course of the voltage-independent state. Because of this, it is not possible for the MCU to poll the peripheral device for operation completion. Polling is also shown to be a rather costly operation due to (2) and (3). Without the ability to communicate to the peripheral device, PACER is necessary to achieve minimal operation latencies.

A. Timing Heuristic

Peripheral operations can vary in their latency or completion times due to a number of factors. Temperature can significantly affect the completion time for peripherals with deterministic timing requirements such as DRAM [6]. Device aging can also affect timing due to a number of issues resulting from fundamental semiconductor physics [7]. Furthermore, some devices simply have non-deterministic completion times due to features such as MMUs and caches that are implemented in various data storage devices like Micro-SD cards, or age and wear as they effect FLASH storage timing.

Because the latency can vary significantly between operations, it is necessary to develop a timing heuristic that can adapt to slowly changing effects like age and temperature as well as rapidly changing factors like cache hits and misses. Adaptive delay estimation is not a new problem [8] and research continues to compensate for non-deterministic delay with different approaches for wireless communications, control systems and mass storage latency [9].

B. Energy Heuristic

For devices with highly variable timing and dynamic current consumption characteristics, integrating the current consumption of the device throughout an operation can allow for better detection of completion. Some operations can be characterized by the amount of charge necessary to complete them. This technique is referred to as “coulomb counting” and is a common technique used to determine the state of charge in rechargeable batteries [10].

C. Current Heuristic

The completion of some peripheral operations are easily detectable by their current consumption profile. These devices have a distinct and deterministic current profile that can be characterized and used to estimate the moment when an operation completes.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Simple and differential power analysis (SPA and DPA) attacks are performed by monitoring device current consumption with very fine grained detail. These attacks seek to undermine encryption techniques by monitoring the current consumption of the processor and detecting the moment at which the processor executes a branch operation [11]. The attacks have been performed on an ARM Cortex MCU using AES and required an extensive measurement setup to accomplish [12]. PACER is inspired by this previous work using fine-grained in-circuit current measurement and fortunately benefits from much more lenient sampling requirements.

III. METHODS

A. Timing Heuristic PACER-T

Some peripheral operations exhibit highly deterministic timing qualities. Such a device is likely to be internally clocked and the operation is waiting for some number of clock cycles to expire before signaling that the operation completed. Such operations are typified Figure 3 in that neither the total energy consumed, nor the profile of that consumption are necessary to predict completion. Regardless of the power profile, the operation always completes within a deterministic window of time. Erases and write operations to EEPROM and flash are typical examples of this behavior.

PACER-T uses the successive approximation algorithm shown in (4) to determine the optimal delay for an operation. The algorithm begins by executing an operation with the amount of delay specified in the device datasheet. After each iteration, if the operation completed earlier than predicted (Pass), then the amount of delay is halved. Otherwise, the operation was ongoing (Fail) and the next delay is increased by half the distance to the last previously successful operation.

$$\begin{aligned} Pass: &\begin{cases} T_{upper} = T_{lower} \\ T_{lower} = T_{lower} - \dfrac{(T_{upper} - T_{lower})}{2} \end{cases} \\ Fail: &\begin{cases} T_{lower} = T_{lower} + \dfrac{(T_{upper} - T_{lower})}{2} \end{cases} \\ Initial\ Conditions: &\ T_{upper} = T_{worst-case}, T_{lower} = 0 \end{aligned} \tag{4}$$

The algorithm is executed online and provides the tightest possible timing. Upon expiration of the predicted wait period, if the device status register indicates that the operation is still occurring, then the algorithm has yielded an early prediction and it is appropriate to continue to wait. This would be considered the 'Fail' case of (4) and future estimates are increased. Otherwise, if the device status register indicates that the operation is complete, then the algorithm has yielded a late prediction and it is appropriate to reduce future estimates.

B. Energy Heuristic PACER-E

Operations that consume a deterministic amount of energy are better characterized by PACER-E. For example, the operation might involve the charging of a storage element such as an inductor or capacitor. In any case, a certain amount of energy is required to complete the operation and once that energy requirement has been satisfied, the peripheral device considers the operation to be complete. Figure 4 is an example of an energy bound operation.

The energy based heuristic was performed similarly to PACER-T in that successive approximation is used. The system multiply-accumulates voltage and current samples fed to the peripheral device. When the digital integration has reached the test value, the operation is 'complete' and checked for correctness. The mechanics of (4) are applied to PACER-E, except that all T limits are replaced with E energy limits. PACER-E is slightly less precise than the timing based algorithm due to the time required to both sample and perform the digital integration necessary for threshold checking.

The energy consumed throughout a test is calculated using the fundamental relationship shown in (5). The results were calculated offline via (6) and (7), where S is the state of the device, and T_s is the sampling period.

$$P = VI = \frac{E}{t} \tag{5}$$

$$E_s = \sum_{n=0}^{N-1} V_n I_n T_s \tag{6}$$

$$E_{total} = \sum_{S_0}^{S_{n-1}} E_s \tag{7}$$

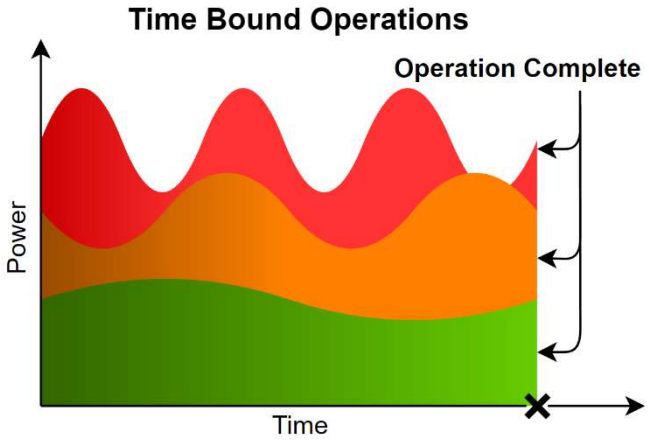


Figure 3: Profile of Three Time-Deterministic Operations

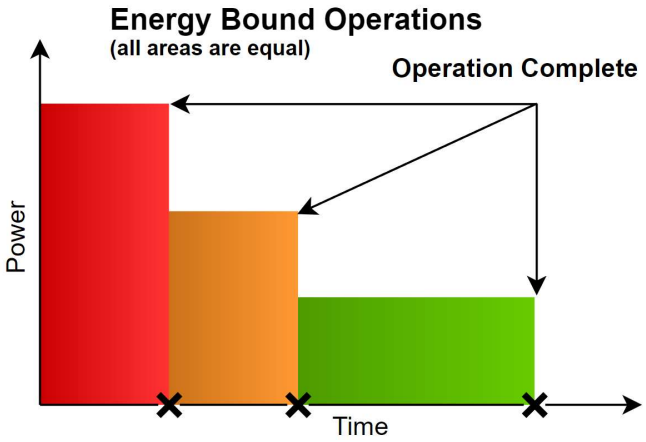


Figure 4: Profile of Three Energy-Deterministic Operations

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

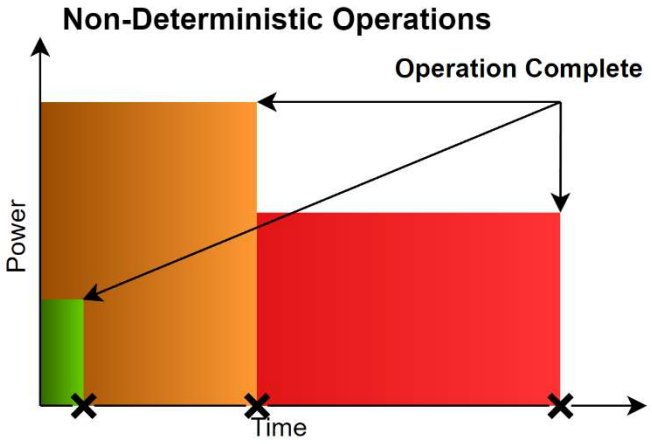


Figure 5: Profile of Three Non-Deterministic Operations

C. Current Heuristic PACER-C

Some operations cannot reliably be defined in terms of time nor energy. One example of a non-deterministic operation would be communications tasks performed by Ethernet or wireless devices that have non-deterministic transmission latencies. Another example would be memory devices that incorporate an onboard memory hierarchy. In such devices, operations are affected by cache latencies.

PACER-C provides recognition that the operation is complete by measuring the idle current usage of the device before the operation begins and marking the operation as complete after the current returns to idle. In order to accommodate operations where the current returns to idle and yet the operation has not yet completed, the algorithm incorporates both a minimum latency and an idle current percent threshold to mark the operation as complete.

Algorithm 1: *PACER-C*

1: ICT = (Idle Current Measurement) * threshold
2: Execute Peripheral Device Operation
2: **While (**t < Minimum Latency) and (I > ICT) **then**
3: I = New Current Measurement
4: **End While**

PACER-C is described by Algorithm 1 and begins by taking a sample of the device input current while idle. Next, the operation is executed and the algorithm waits for a minimum latency period to expire. The operation is considered complete after the output current returns to the threshold percentage of its previous state. The threshold for all following experiments was set empirically at 110%.

PACER-C is the most basic method to determine in real time if an operation is complete and may also be prone to false positives in some cases. There are many more advanced algorithms that can suit the purpose such as a multi-layer perceptron that is used in neural networks that could be used to identify features in real-time. It is notable however, that reducing the complexity of the detector is important so that the algorithm can ensure that it is keeping pace with incoming samples. Naturally, more complex algorithms could be accommodated by a more powerful host microcontroller.

IV. MATERIALS

PACER and IODVS are implemented on an STM32F429 MCU supported by the STMicroelectronics DISCO board and hosted by the PRIME (Precise Real-Time In-Circuit Micro-EMS) assembly. The board provides 64MB of SDRAM which allows for simultaneous sampling throughout the test suite at very high speed. All experiments were sampled at 1MSPS and the SDRAM allowed any individual experiment to last up to 1 full second. All of the analog conversions as well as the device state sampling were performed via DMA. Therefore, the test fixture is expected to have had no impact on the operation under test.

The PRIME assembly, shown in Figure 6, hosts a variety of peripherals (labelled in red as DUT: Devices Under Test) that are common in embedded designs. The board provides access to Bluetooth, Wi-Fi and a Si1143 proximity detector. PACER was evaluated on NAND and NOR FLASH memories, as well as a commercial EEPROM, temperature / humidity sensor and four independent Micro-SD cards.

At 1 MSPS and 4 channel measurements and 2 bytes per sample, each test can result in up to 8 megabytes of data. Because repeatability is so important, each test was run 50 times. Therefore, bandwidth became a limiting factor and a Hi-Speed (480Mbps) USB module was added to the board to allow for rapid development. Operating as a virtual communications port and using MCU parallel bus, actual bandwidth was realized at approximately 120Mbps.

Each of the peripheral devices under test has some method of determining if an operation completed successfully. For the memory devices, a simple read-back verification is sufficient to determine correctness and is a common practice among embedded designs. The temperature and humidity sensor provides a status bit indicating if an operation is in progress, thus indicating that a requested operation has not yet completed.

Power is provided and voltage is modulated to each individual device on the domain using independently configurable power supplies. The ASDM-300F module shown in orange on Figure 6 provides a high-efficiency buck power supply, followed by a linear regulator with a high ripple-rejection ratio. A high-precision and clean power

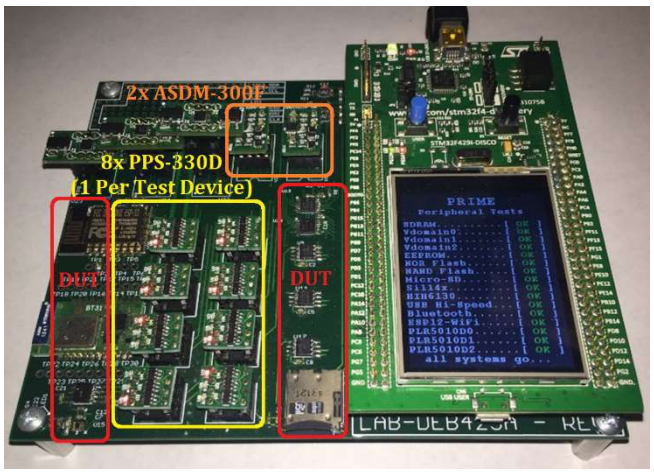


Figure 6: PRIME (Precise Real-Time In-Circuit Micro-EMS)

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

supply is important because PACER uses the current profile to make real-time decisions. If the power supply outputs a significant amount of noise, then it becomes difficult to acquire signal and determine activity completion in real-time.

The ASDM-300F is also outfitted with a dual current measurement circuit using the Maxim MAX4377HAUA+. This circuit allows the host to measure both the input and output current of the power supply with high analog bandwidth. Figure 7 shows peripheral current output (PIOx) and peripheral current input (PIIx) along with peripheral voltage (PVx) routed to their associated ADCs on the MCU. Note that all signals are buffered to provide high-driving capability and therefore a fast response time.

Ultimately, these measurements are used to determine activity completion with the PACER-E and PACER-C algorithms. It is important to note the gain-bandwidth product of the amplifier. High frequency content will be attenuated to some degree and the actionable data output would be of higher quality if a higher frequency device were available.

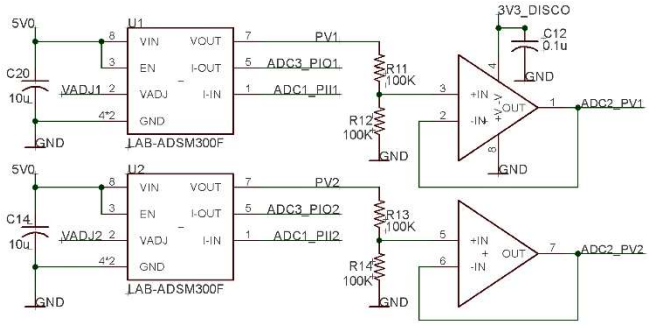


Figure 7: Voltage and Current Measurements from ASDM-300F

While measuring and classifying activity completion, it is important that each device be analyzed independently. The PPS-330D shown in yellow on Figure 6 allows the host to switch the voltage domain of an individual peripheral to any one of three domains, or disconnect the device entirely

PPS-330D devices are connected to each peripheral, and while a peripheral is under test, the remaining devices are switched to an alternate voltage domain. Thus, each device is independently classified in-system without physically removing other devices that may affect current measurements. Once the devices are characterized, then their individual contributions to the power supply current output can be deduced through superposition. The PPS-330D is convenient for initial profiling, but unnecessary for a streamlined implementation. The ASDM-300F is necessary for an IODVS implementation, but PACER-E and PACER-C only require the current measurement component.

V. RESULTS

Initial IODVS results were repeated so as to establish a baseline with which to compare the results of PACER. Previous experiments required the results to be averaged many times over. The PRIME assembly provides high enough signal to noise ratio that averaging multiple test results is unnecessary and a simple 50-sample moving average provides enough filtering while maintaining a quick response time.

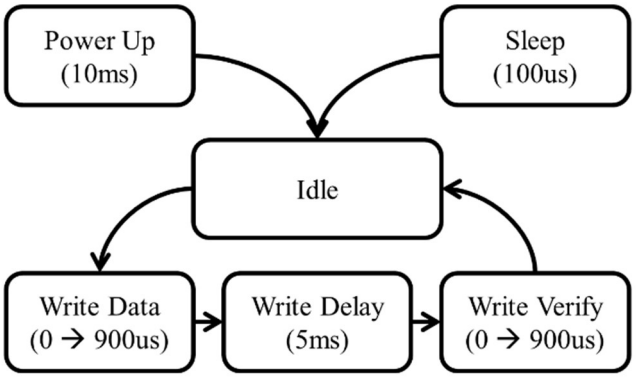


Figure 8: EEPROM Write State Transition Diagram

A. MCP25AA512 EEPROM

The Microchip EEPROM is specified by the manufacturer for a 5ms mandatory wait period following the write command and data. The full state transition diagram is shown in Figure 8. Upon receiving a write command and associated data, the delay begins and the operation completes after a verification stage. This operation is highly deterministic with respect to time, energy and current profile.

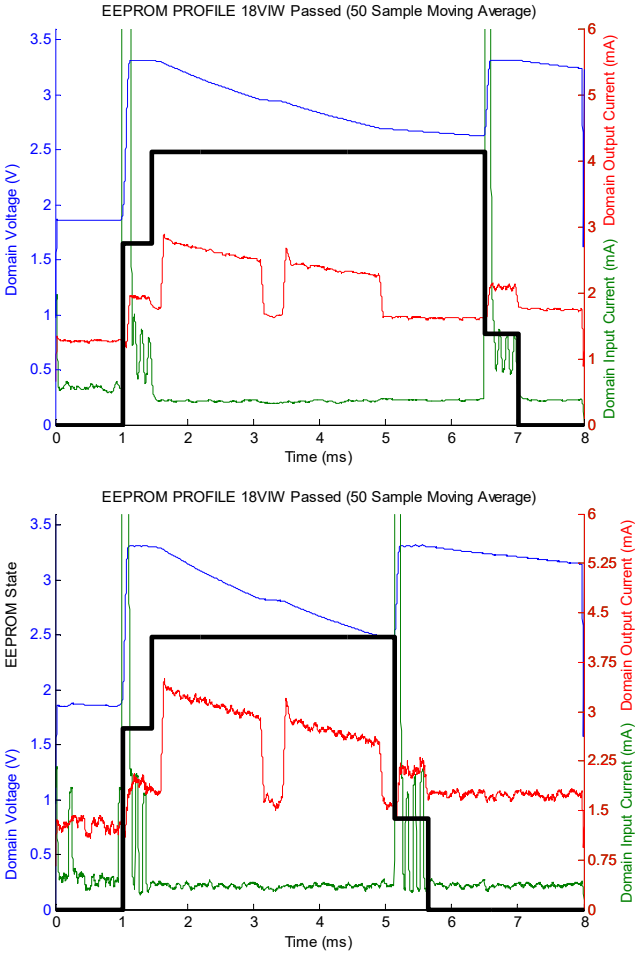


Figure 9: EEPROM Write Cycle Using IODVS and PACER-T

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. MCP25AA512 EEPROM PACER RESULTS

Stage	Latency Results (ms)				
	Control	*PACER-T*	*Diff.*	*PACER+IODVS*	*Diff.*
Wait	5.05	3.51	30.5%	3.51	30.5%
All	5.98	4.44	25.7%	4.44	25.7%
	Energy Results (uJ)				
Wait	46.84	37.89	19.1%	27.85	40.5%
All	53.05	43.91	17.2%	32.40	38.9%

The current waveform of Figure 9 shows that the EEPROM write operation begins at t=1.5ms and indicates completion at approximately t=5ms instead of t=6.5ms as specified by the manufacturer. After applying the PACER-T algorithm, it is indeed true that the operation was complete at the 5ms mark, thus reducing the wait latency by 30.5%.

Due to the deterministic nature of the operation, all algorithms identified activity completion with high accuracy as summarized in TABLE I. The PACER-E and PACER-C algorithms were also successful in identifying activity completion with latency reductions of 25.7% and 23.2%, respectively. The two algorithms do require additional computation to integrate or otherwise observe the current waveform. Given identical performance, PACER-T is the best choice in this application.

B. Numonyx M25PX16 NOR Serial Flash

NOR flash modules sacrifice byte-wise modification for overall capacity. The M25PX16 presents 16MBits of capacity in a small package, but the host must erase sub-sectors of flash (4K) to write pages of flash (128B). To perform a read-modify-write operation, the host must read the contents of a sub-sector, modify the contents locally, erase the sub-sector in flash and finally write the modified contents back to the flash on a page-by-page basis. The complete state transition diagram is shown below in Figure 10.

Both the sub-sector erase and page write have a worst-case delay specified by the manufacturer. PACER algorithms were run against both operations to find the comprehensive result. Although specified for 150ms, the current waveform indicates

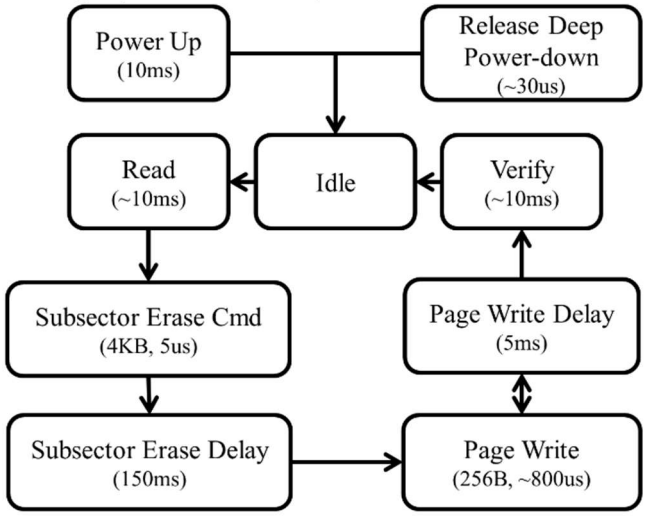


Figure 10: Serial Flash Write State Transition Diagram

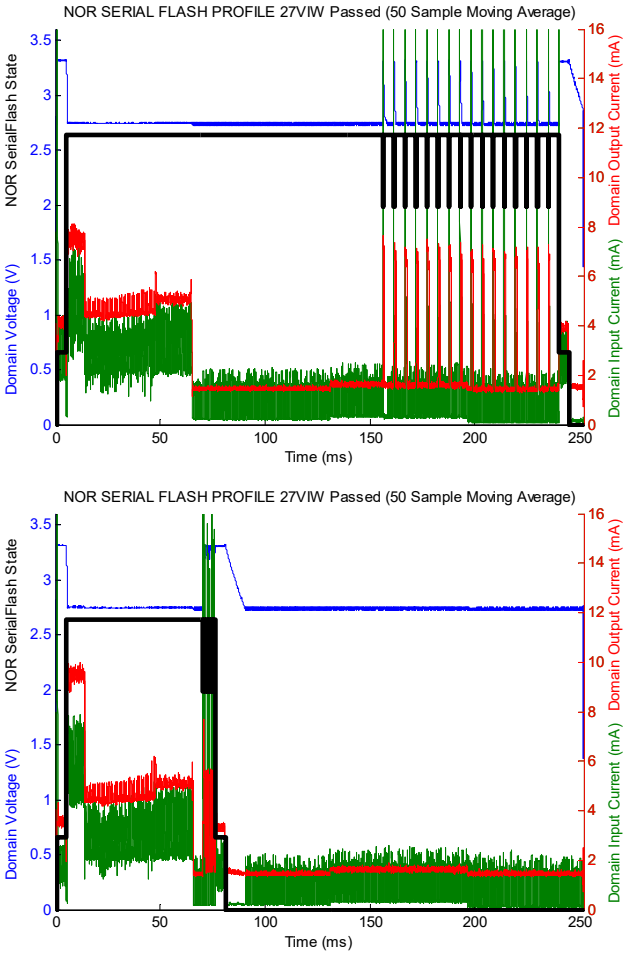


Figure 11: NOR Serial Flash Write-Cycle using IODVS and PACER-T

that the sub-sector erase completed approximately 65ms after it begins. Page writes are specified for a worst-case completion time of 10ms but through the application of PACER-T, they complete much faster as shown in TABLE II. The wait figure is the total amount of time spent waiting for the erase and the aggregate amount of time for each page write. The PACER-T algorithm delivered a 70% decrease in wait latency which yielded a 38.9% decrease in overall energy consumption. The worst-case manufacturer specification appears to be very pessimistic, although may be appropriate across both process and temperature variables.

TABLE II. M25PX16 NOR SERIAL FLASH PACER RESULTS

Stage	Latency Results (ms)				
	Control	*PACER-T*	*Diff.*	*PACER+IODVS*	*Diff.*
Wait	231.57	69.47	70.0%	66.92	71.1%
All	243.87	82.45	66.2%	80.26	67.1%
	Energy Results (uJ)				
Wait	2138.3	1212.0	43.3%	1029.52	51.9%
All	2277.0	1392.0	38.9%	1158.26	49.1%

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

C. Microchip SST26VF016B Serial NAND Flash

The SST26 serial flash module uses NAND-like control logic to provide higher capacity and lower latency than the NOR serial flash. However, the device sacrifices the random-access timing benefit of NOR flash. The serial flash module must therefore read an entire page of flash into a local buffer before providing read data to the host. This can result in non-deterministic read and write access times.

Despite the core logic differing from the M25PX16, PACER-T still performed the best. Application yielded a 66.6% decrease in aggregate wait times and a 17.8% decrease in energy consumption due to waiting as shown in TABLE III.

TABLE III. SST26VF016B NAND SERIAL FLASH PACER RESULTS

Stage	Latency Results (ms)				
	Control	*PACER-T*	*Diff.*	*PACER+IODVS*	*Diff.*
Wait	57.61	19.26	66.6%	19.27	66.6%
All	71.28	32.94	53.8%	32.95	53.8%
	Energy Results (uJ)				
Wait	1053.0	806.2	23.8%	584.87	44.5%
All	1247.9	997.26	17.8%	801.95	35.7%

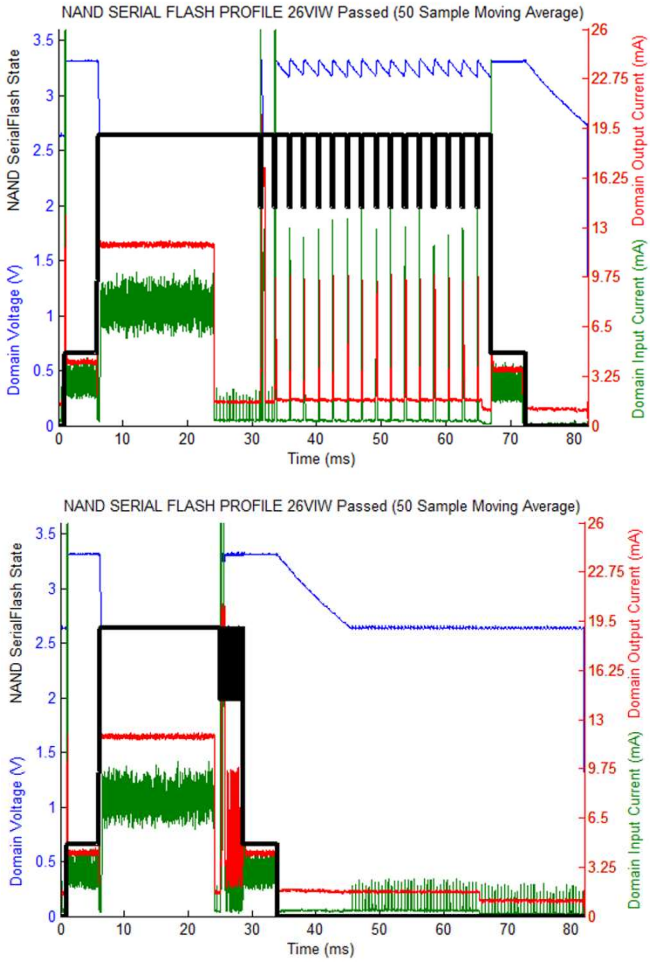


Figure 12: NAND Serial Flash Write Cycle Using IODVS and PACER-T

D. An assortment of Micro-SD Memory Cards

Onboard caches and memory management units cause the write operation of Micro-SD cards to have non-deterministic timing. In this case, PACER-C is the only algorithm that can reliably detect when the operation is finished. As with all memory tests, writes were performed with random data to random addresses throughout the memory space and so the cache performance is thoroughly exercised.

Figure 13 shows the massive power and latency difference between a cache miss and a cache hit. The cache miss has an overall response time of about 150ms, while the cache hit has a response time of about 1ms. It is important to recall that PACER-C is effective in this situation because it samples current on the domain at 1us intervals and makes decisions with a 50 sample moving average.

The SD-Card protocol is polling-based, as shown in the state transition diagram of Figure 14. There can be significant power-up and initialization delays. This prevents traditional dynamic power management (disabling the peripheral when not in use) techniques from being effective. Although constant polling is a good way to minimize latency, it results in extreme power increases. The analysis uses worst-case execution time (WCET) as the control group.

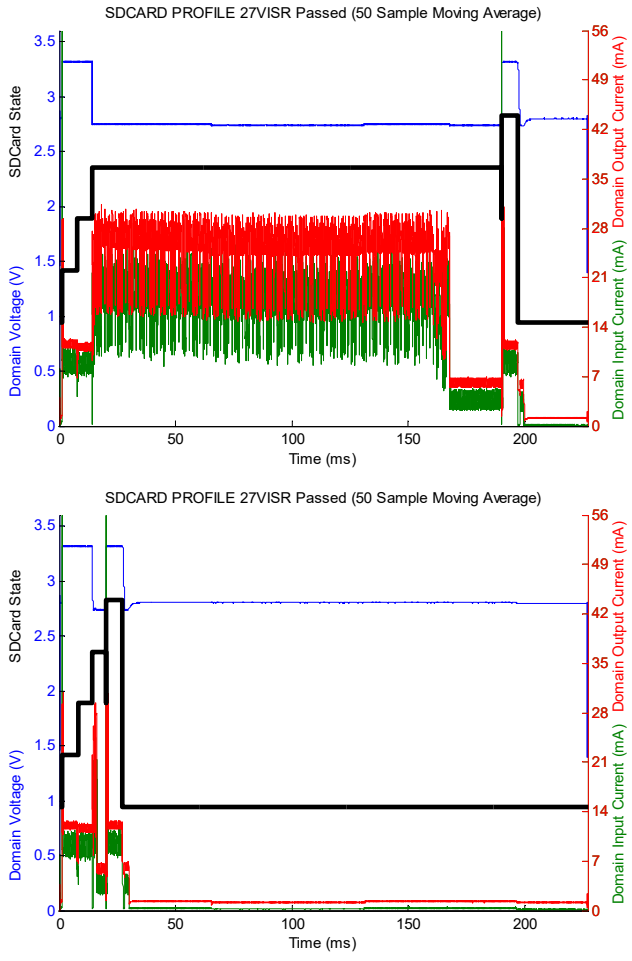


Figure 13: A Micro-SD Card Cache Miss and a Cache Hit

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

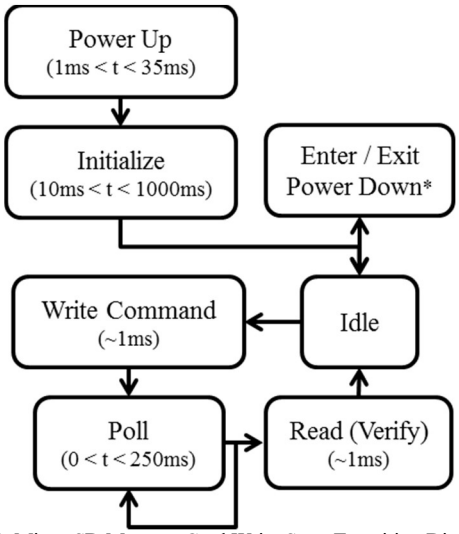


Figure 14: Micro-SD Memory Card Write State Transition Diagram

The WCET for each Micro-SD Card was determined to be the longest-observed operation duration for the measurement set under consideration. This WCET typically corresponds with a cache miss on the device.

Figure 15 helps to describe the performance differences shown in TABLE IV. The control delay is set to the WCET delay for each characterization, PACER-C allows the host to react to those operations deviating considerably from the control. Therefore, the Sandisk and Lexar cards benefitted considerably because they exhibit a bimodal timing distribution. The Swissbit card benefits decisively because of the mostly normal timing distribution. The Kingston card does not benefit as much because write timing exhibits a very low standard deviation. To present complete timing effects, a thorough latency analysis would need to be done on each device. Only energy results are presented here, but they are correlated with overall latency decreases.

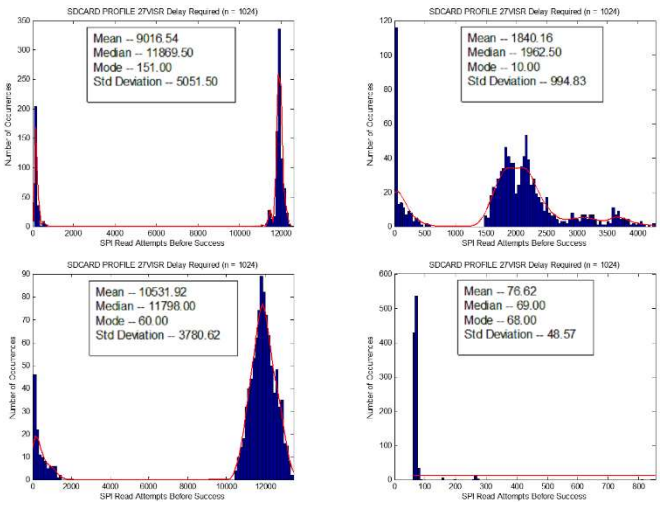


Figure 15: Timing Performance among Tested SD-Cards

TABLE IV. MICRO-SD CARD PACER RESULTS

Stage	Energy Results (uJ)				
	Control	*PACER-C*	*Diff.*	*PACER+IODVS*	*Diff.*
Sandisk	17066	15198	10.9%	11848	30.6%
Lexar	22707	21428	5.6%	16977	25.24
Swissbit	2763	914	66.9%	554	80.0%
Kingston	942	933	0.9%	897	4.8%

The Sandisk, Lexar, Swissbit and Kingston cards were manufactured in 2007, 2008, 2014 and 2015, respectively. It is reasonable to infer that the technology within the card advanced considerably within that time and will continue to do so in the future. Another key-takeaway from the statistical distribution of write operations on random memory locations is that the associativity appears to be different between the cards. The Lexar and Sandisk cards appear to be set-associative while the Swissbit appears to be fully associative. The standard deviation and response time was so low for the Kingston card that it cannot be determined what type of cache structure is used internally.

E. Honeywell HIH-6130 Temperature / Humidity Sensor

The Honeywell HIH-6130 communicates via the I^2C bus. Therefore, the device must only match the host voltage during the host-read states which is convenient for IODVS application. As shown in the state transition diagram of Figure 16, the host requests the sensor to take a measurement and then waits the manufacturer-specified 45ms for the measurement to complete. Finally, the host retrieves the completed measurement. PACER-E demonstrated the best performance among the algorithms, perhaps because of the capacitive nature of the peripheral ADC. The effects are shown in Figure 17 and the numeric results are presented in TABLE V.

The PACER-T algorithm also produced impressive results with a wait latency of 31.66ms and wait energy of 254.14uJ. Compared with PACER-E, the result corresponds with a slightly increased latency of 0.5% and slightly increased energy consumption of 4.3%. For some applications, the simplicity of the PACER-T implementation may be preferable when compared to the best performing PACER-E algorithm.

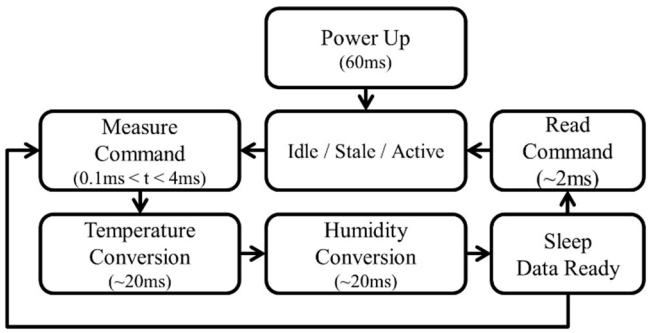


Figure 16: HIH-6130 State Transition Diagram

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

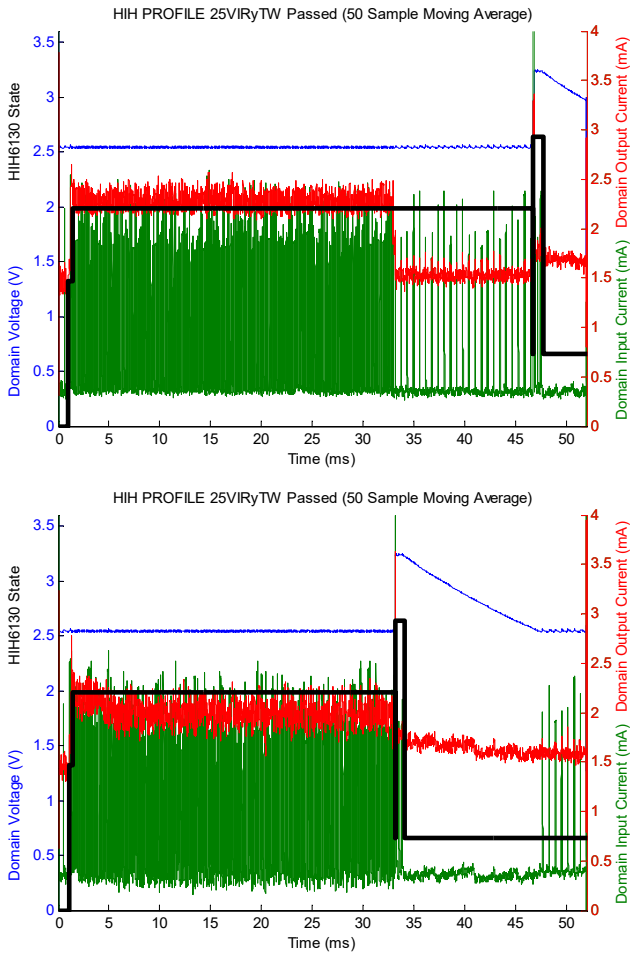


Figure 17: HIH-6130 Measurement Cycle Using IODVS and PACER-E

TABLE V. HONEYWELL HIH-6130 PACER RESULTS

Stage	Latency Results (ms)				
	Control	*PACER-E*	*Diff.*	*PACER+IODVS*	*Diff.*
Wait	45.27	31.45	66.6%	19.27	66.6%
All	45.99	32.17	53.8%	32.95	53.8%
	Energy Results (uJ)				
Wait	325.95	240.29	26.3%	169.62	48.0%
All	330.50	245.39	25.8%	173.89	47.4%

VI. SYSTEM COST / BENEFIT ESTIMATION

Application of the PACER algorithms results in moderate to extreme decreases in peripheral response time and power consumption. These algorithms do require that the application MCU perform some computation before and during the peripheral operation. All of the algorithms require a high-precision (hardware) timer and PACER-E/C require active analog to digital conversion (ADC).

Equation (1) of the introduction demonstrates the difficulty in accurately estimating the power consumption of a polling scenario. Fortunately, keeping the control group as the WCET does allow for an estimate of overall impact.

TABLE VI. ALGORITHM RESOURCE REQUIREMENTS

	CPU Usage	Timer	ADC	DMA,
Waiting	0%	Active	Disabled	Inactive
PACER-T	<1%	Active	Disabled	Inactive
PACER-E	5%	Active	Active	Active
PACER-C	3%	Active	Active	Active

TABLE VII. MODERN MICROCONTROLLER CHARACTERISTICS

MCU	STM32F730	STM32F401	Apollo2
Frequency (MHz)	216	84	48
CPU Current (mA)	90	11.5	0.432
ADC (mA)	1.6	1.6	0.541
DMA (mA)	3.1	1.7	N/A

The resource requirements for implementing both the control group and the various PACER algorithms are shown in TABLE VI. Performance and power estimates of required peripherals among a few modern MCUs are shown in TABLE VII. The most important aspect of this data is the massive increase in CPU energy expenditure in order to enable high-speed analog sampling.

The PACER-E and PACER-C tests were completed with a 1us sampling period. The ADC results were transferred via DMA to a buffer where they were arithmetically manipulated in order to achieve the current and energy results as required by the algorithm. The Apollo2 MCU does not have a DMA unit onboard and therefore the sampling period would need to be reduced and CPU energy expenditure would rise. Fortunately, due to the sub-threshold switching nature of that MCU, the expected utilization increase would not result in substantially higher overall energy expenditure.

Primarily, these tables indicate that PACER-T is generally the best algorithm to implement from a system-level evaluation. For non-deterministic peripherals such as the Micro-SD cards, only the PACER-C algorithm was able to detect early completions. In this case, a balance needs to be achieved between the energy costs of sample rate and peripheral latency/power reductions.

VII. EXTENDED ALGORITHM RESULTS

Previous sections presented only the result of the best algorithm. Complete results are presented in this section for each peripheral against each and every algorithm.

There is an exception for the Micro-SD cards because those devices are non-deterministic and therefore the results presented in that section are already comprehensive.

Note that the abridged results of the previous sections have had the final idle-phase of each test removed because that additional power consumption and latency is an artifact of the test fixture rather than real-world application. The following tables TABLE VIII through TABLE XV present the complete and comprehensive results of the tests run on each device under test.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE VIII. EEPROM OPERATION ENERGY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	6.03	2.47	6.04	2.52	6.06	2.47	6.13	2.48
Writing	2.71	2.55	2.73	2.51	2.69	2.50	2.80	2.58
Waiting	46.84	33.85	37.89	27.85	33.59	25.06	35.47	26.24
Read	3.50	3.48	3.29	2.04	3.51	3.49	3.55	3.21
Idle	5.89	5.76	15.30	13.42	14.97	14.16	14.50	14.03
Active Total	*64.97*	*48.11*	*65.24*	*48.34*	*60.82*	*47.69*	*62.44*	*48.54*
Delta	**0.00%**	**-24.83%**	**-17.24%**	**-38.93%**	**-25.00%**	**-41.46%**	**-21.17%**	**-39.62%**

TABLE IX. EEPROM OPERATION LATENCY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	1.025	1.025	1.026	1.025	1.026	1.025	1.038	1.038
Writing	0.429	0.43	0.429	0.429	0.429	0.43	0.429	0.429
Waiting	5.045	5.044	3.508	3.507	3.54	3.508	3.603	3.653
Read	0.506	0.508	0.506	0.507	0.506	0.506	0.507	0.508
Idle	0.994	0.992	2.53	2.531	2.498	2.53	2.422	2.371
Active Total	*5.98*	*5.982*	*4.443*	*4.443*	*4.475*	*4.444*	*4.539*	*4.59*
Delta	**0.00%**	**0.03%**	**-25.70%**	**-25.70%**	**-25.17%**	**-25.69%**	**-24.10%**	**-23.24%**

TABLE X. NOR SERIAL FLASH OPERATION ENERGY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	6.04	4.66	6.05	4.61	6.07	4.65	6.15	4.63
Reading	49.99	50.08	58.08	50.21	50.13	49.89	49.87	50.14
Erase	0.95	0.96	0.91	0.94	0.96	0.97	0.99	0.95
Total Write	39.17	36.31	63.05	41.26	37.68	20.40	37.36	17.81
Total Wait	2138.32	1713.89	1211.99	1029.52	1501.73	1178.32	1319.34	1040.74
Reading	48.60	48.78	51.91	31.72	48.59	33.76	48.47	42.02
Idle	37.05	33.13	929.24	711.59	689.63	576.83	859.97	657.46
Active Total	*2277.02*	*1854.68*	*1391.98*	*1158.26*	*1645.17*	*1287.98*	*1462.18*	*1156.28*
Delta	**0.00%**	**18.55%**	**38.87%**	**49.13%**	**27.75%**	**43.44%**	**35.79%**	**49.22%**

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE XI. NOR SERIAL FLASH OPERATION LATENCY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	1.04	1.04	1.04	1.04	1.04	1.04	1.05	1.05
Reading	4.27	4.27	4.27	4.27	4.27	4.27	4.27	4.27
Erase	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
Total Write	3.31	3.32	3.31	3.32	3.31	3.32	3.31	3.32
Total Wait	231.57	231.57	69.47	66.92	104.81	120.17	80.15	79.06
Reading	4.64	4.64	4.27	4.63	4.64	4.73	4.64	4.64
Idle	7.09	7.08	169.55	171.74	133.86	118.38	158.50	159.57
Active Total	*243.87*	*244.92*	*82.45*	*80.26*	*118.14*	*133.62*	*93.50*	*92.43*
Delta	**0.00%**	**-0.43%**	**66.19%**	**67.09%**	**51.55%**	**45.21%**	**61.66%**	**62.10%**

TABLE XII. NAND SERIAL FLASH OPERATION ENERGY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	5.62	4.12	5.54	4.07	5.58	4.08	5.64	4.14
Reading	71.49	71.57	72.58	71.52	71.45	72.10	71.54	71.39
Erase	1.58	1.59	1.53	1.58	1.55	1.49	1.55	1.52
Total Write	51.88	48.14	75.40	70.70	73.01	70.19	51.63	44.66
Total Wait	1052.98	806.15	802.63	584.87	817.59	596.84	887.28	670.32
Reading	69.97	69.81	73.11	73.27	72.90	73.11	72.75	72.85
Idle	52.31	44.66	249.37	187.25	234.28	158.54	149.37	133.95
Active Total	*1247.90*	*997.26*	*1025.25*	*801.95*	*1036.50*	*813.72*	*1084.76*	*860.73*
Delta	**0.00%**	**-20.08%**	**-17.84%**	**-35.74%**	**-16.94%**	**-34.79%**	**-13.07%**	**-31.03%**

TABLE XIII. NAND SERIAL FLASH OPERATION LATENCY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	1.04	1.04	1.04	1.04	1.04	1.04	1.05	1.05
Reading	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00
Erase	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
Total Write	3.31	3.32	3.31	3.31	3.31	3.32	3.31	3.31
Total Wait	57.61	57.62	19.26	19.27	25.92	25.92	36.19	36.20
Reading	5.28	5.28	5.29	5.29	5.28	5.28	5.28	5.29
Idle	9.68	9.67	48.02	48.01	41.37	41.36	31.09	31.07
Active Total	*71.28*	*71.29*	*32.94*	*32.95*	*39.59*	*39.60*	*49.86*	*49.87*
Delta	**0.00%**	**0.01%**	**-53.79%**	**-53.78%**	**-44.46%**	**-44.45%**	**-30.05%**	**-30.03%**

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE XIV. HIH-6130 OPERATION ENERGY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	5.19	3.60	5.28	3.64	5.24	3.62	5.33	3.67
Writing	1.75	0.98	1.76	0.93	1.73	0.98	1.76	1.00
Waiting	325.95	231.17	254.14	120.39	240.29	169.62	223.65	159.00
Idle	0.08	0.05	0.09	0.05	0.12	0.10	0.09	0.05
Reading	2.80	2.64	2.98	3.10	3.37	3.30	2.95	2.92
Idle	28.04	26.18	105.49	83.50	106.34	84.62	105.31	83.60
Active Total	*330.50*	*234.79*	*258.88*	*124.41*	*245.39*	*173.89*	*228.36*	*162.91*
Delta	**0.00%**	**-28.96%**	**-21.67%**	**-62.36%**	**-25.75%**	**-47.39%**	**-30.91%**	**-50.71%**

TABLE XV. HIH-6130 OPERATION LATENCY

State	Control	IODVS	PACER-T	PACER-T + IODVS	PACER-E	PACER-E + IODVS	PACER-C	PACER-C + IODVS
Idle	1.01	1.01	1.01	1.01	1.01	1.01	1.03	1.03
Writing	0.23	0.44	0.23	0.44	0.23	0.44	0.23	0.44
Waiting	45.27	45.27	31.66	31.44	31.45	31.41	31.70	31.60
Idle	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Reading	0.49	0.95	0.49	0.95	0.49	0.95	0.49	0.95
Idle	4.98	4.32	18.59	18.15	18.80	18.18	18.54	17.98
Active Total	*45.99*	*46.65*	*32.38*	*32.82*	*32.17*	*32.80*	*32.42*	*32.98*
Delta	**0.00%**	**1.44%**	**-29.59%**	**-28.63%**	**-30.05%**	**-28.69%**	**-29.50%**	**-28.28%**

VIII. CONCLUSIONS

Applying the PACER suite of algorithms to a variety of common embedded peripherals resulted in significant reductions to both latency and energy consumption. The PACER-T algorithm performed best against time-bound operations and was very competitive in energy-bound operations. For non-deterministic operations, the PACER-C algorithm performed well. When measured against a median baseline, the algorithm performed even better as the operational latency increased in randomness.

The PACER-T and PACER-E algorithms use successive approximation and the PACER-C algorithm uses a return-to-idle measurement to determine activity completion. It is likely that the performance of both methods could be enhanced further through the application of more complex algorithms. PACER-T could be applied to memory operations with a bimodal delay distribution by first testing for a cache hit and then delaying for a determined cache-miss time. Likewise, the PACER-C algorithm could be modified online so as to identify the current waveform features corresponding varying latencies, thus allowing the MCU to sleep longer.

The designer must be judicious in selecting the appropriate algorithm in order to minimize total energy expenditure. For most cases, the PACER-T algorithm provides sufficient performance improvements without the added computational and peripheral energy costs. For devices with a non-deterministic operation execution time, then PACER-C is the only option. However, the designer can still adjust the sample rate so as to perhaps eliminate the need for DMA and reduce the duty cycle of the ADC. Reducing the sample rate will indeed increase the latency of the algorithm, but the overall implications and design tradeoffs should be handled at the system level.

Combining the best aspects of PACER-T and PACER-C would be an interesting topic for further research. Predicting that a memory access will result in either a cache hit with minimal latency. Failing that, the algorithm could assume a cache miss and delay for a certain amount of time before resuming measurements to determine if current consumption has returned to the idle state. This method would strive to reduce the duty cycle of all three MCU energy consumers, the CPU, the ADC peripheral and the DMA peripheral. The algorithm could also eliminate the need to sample the voltage channel because only the current channel indicates activity.

As the cost of computation in embedded systems continues to decrease, it is natural to devote more computational resources to minimizing system-wide energy consumption and latency. The PACER suite of algorithms use minimal computational resources and are shown to decrease latency by up to 67% and device energy consumption by up to 80% when compared to the naïve worst-case estimate.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

REFERENCES

[1] D. Moore and A. Dean, "PACER: Peripheral Activity Completion Estimation and Recognition," in *The Thirteenth International Conference on Systems (ICONS 2018) IARIA*, Apr. 2018, pp 38-45, ISBN: 978-1-61208-626-2.

[2] B. Brock and K. Rajamani, "Dynamic power management for embedded systems [SOC design]," in *Proc. IEEE International [Systems-on-Chip] SOC Conference 2003*, IEEE Press, Sep. 2003, ISBN: 0-7803-8182-3.

[3] C. Kumar, M. Sindhwani and T. Srikanthan, "Profile-based technique for Dynamic Power Management in embedded systems," in *International Conference on Electronic Design (ICED 2008)*, IEEE Press, Dec. 2008, ISBN: 978-1-4244-2315-6.

[4] W. Dargie, "Dynamic Power Management in Wireless Sensor Networks: State-of-the-Art," *IEEE Sensors Journal,* vol. 12, no. 5, pp. 1518 - 1528, 2012, ISSN: 1558-1748.

[5] D. Moore and A. Dean, "Intra-Operation Dynamic Voltage Scaling," in *2015 IEEE 3rd International Conference on Cyber-Physical Systems, Networks, and Applications*, Hong Kong, IEEE Press, Aug. 2015, ISBN: 978-1-4673-7785-0.

[6] D. Lee, Y. Kim, G. Pekhimenko, S. Khan, V. Seshadri, K. Chang and O. Mutlu, "Adaptive-latency DRAM: Optimizing DRAM timing for the common-case," in *IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, IEEE Press, Feb. 2015, ISBN: 978-1-4799-8930-0.

[7] S. Sadeghi-Kohan, M. Kamal, J. McNeil, P. Prinetto and Z. Navabi, "Online self adjusting progressive age monitoring of timing variations," in *10th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, IEEE Press, Apr. 2015, ISBN: 978-1-4799-1999-4.

[8] D. S. S. Etter, "Adaptive Estimatation of Time Delays in Sampled Data Systems," in *IEEE Transactions on Acoustics Speech and Signal Processing (Volume: 29, Issue: 3)*, IEEE Press, Jun. 1981, ISSN: 0096-3518.

[9] S. G. P. A. Z. E. Tarasov V, "Efficient I/O Scheduling with Accurately Estimated Disk Drive Latencies," in *The Proceedings of OSPERT 2012*, 2012.

[10] H. Macicior, M. Oyarbide, O. Miguel, I. Cantero, J. Canales and A. Etxeberria, "Iterative capacity estimation of LiFePO4 cell over the lifecycle based on SoC estimation correction," in *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*, IEEE Press, Nov. 2013, ISBN: 978-1-4799-3832-2.

[11] H. Mahanta, A. Azad and A. Khan, "Power analysis attack: A vulnerability to smart card security," in *International Conference on Signal Processing And Communication Engineering Systems (SPACES)*, IEEE Press, Mar. 2015, ISBN: 978-1-4799-6109-2.

[12] M. Petrvalsky, M. Drutarovsky and M. Varchola, "Differential power analysis attack on ARM based AES implementation without explicit synchronization," in *2014 24th International Conference Radioelektronika*, IEEE Press, Jun. 2014, ISBN: 978-1-4799-3715-8.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Transient Analysis of a Single-stage Vapor Compression Refrigeration System Using Lumped Parameter Approaches

Analysis and simulation validation based on a reduced order differential equation with few degrees of freedom

Guillermo Domínguez Librado
Cooling Systems Dynamics Modeling
Engineering Center for Industrial Development (CIDESI)
Querétaro, México
e-mail: gdominguez@posgrado.cidesi.edu.mx

Eloy Edmundo Rodríguez Vázquez
National Research Laboratory on Cooling Technology
Engineering Center for Industrial Development (CIDESI)
Querétaro, México
e-mail: eloy.rodriguez@cidesi.edu.mx

Luis Alvaro Montoya Santiyanes
Rotordynamics for Cooling
Engineering Center for Industrial Development (CIDESI)
Querétaro, México
e-mail: lmontoya@posgrado.cidesi.edu.mx

J. Hernán Pérez Vázquez
Heat Interchangers with Local Compression
Engineering Center for Industrial Development (CIDESI)
Querétaro, México
e-mail: jperez@posgrado.cidesi.edu.mx

C. Alexander Nuñez Martín
Nation Dynamic Optimization of Cooling Devices
Engineering Center for Industrial Development (CIDESI)
Querétaro, México
e-mail: cnunes@posgrado.cidesi.edu.mx

***Abstract*—Refrigeration and air conditioning systems need to have enough capacity to maintain the desired temperature at a worst-case, design load operating condition. In this paper, a dynamic analysis of a single-stage vapor-compression refrigeration system is presented. The model is constructed by applying the lumped parameter approach to each component of the refrigeration system; the first low of thermodynamic is applied to individual components to determine the mass and energy balances; then, a linear dynamical system is obtained. The model is implemented by MATLAB and simulation results are given for comparison with real values. The results of the simulation match with the manufacturer's specifications.**

Keywords-Heat exchangers; Refrigerants; Dynamic Model; Household refrigeration; Transient conditions; Control volume.

I. INTRODUCTION

Refrigeration and air conditioning are an active and fleet developing technologies. These devices are closely related to the living standard of people and to the outdoor environment, due to ozone depletion and global warming.

Mathematical modeling is the most practical way of studying the basic behavior of cooling cycle performance, the relative losses in various components and their interactions. Standard science and engineering formulations are applied to describe mathematically the basics processes occurring in the Vapor Compression Refrigeration (VCR) systems. Mathematical modeling is a step towards simulation optimization [1], [2].

Dynamic models are often classified using such terms as white box, gray box, or black box. The term white-box models refer to physics based models that are described using physical laws, such as conservation equations. These models also appear in the literature as mechanistic models or first principles models [1], [3].

At the other extreme, black-box models refer to empirical or data-driven models, where transient experimental data is used to identify a dynamic model. This process is also known as system identification or time-series analysis, and it can be used to construct models in the time or frequency domain. In black-box model one tries to estimate the functional form of relations between variables and the numerical parameters with no need of detailed information about the components of the system [1], [3]. Examples of empirical models include regression analysis, polynomial curve fits and artificial neural networks.

The bulk of modeling efforts for VCR systems are most appropriately termed as gray-box, due to they are largely based on the governing physics but including semi-empirical terms, such as efficiency maps, heat transfer correlations, etcetera, that come out from experimental test. Physics-based modeling paradigms include;

- lumped parameter approaches that capture the gross pressure and cooling transients qualitatively,
- moving boundary approaches, which model the dynamic variations in phase transition points, and

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

- finite control volume approaches, which use discretized models including temperature and parameter gradients, in an effort to achieve greater accuracy [1], [3].

TABLE I. NOMENCLATURE

Symbol	Description	Units SI
$c_{p,m}$	Specific heat of material	J/kg-K
c_{pi}	Specific heat of refrigerant	J/kg-K
C_{Ti}	Thermal capacity	J/K
g	Gravity	m/s^2
h	Refrigerant enthalpy	J/kg
m_r	Mass flow rate	Kg/s
n	Politropic coefficient	
Q_1	Evaporator heat rate	W
Q_3	Codenser heat rate	W
Q_{P1}	Heat transfer rate to the surrounding in the evaporator	W
Q_{P2}	Heat transfer rate to the surroundings in the compressor	W
Q_{P3}	Heat transfer rate to the surroundings in the condenser	W
Q_{P4}	Heat transfer rate to the surroundings in the expansion device	W
P	Pressure	Bar
T	Temperature	K
T_a	Ambient temperature	K
T_1	Evaporator temperature	K
T_2	Compressor temperature	K
T_3	Condenser temperature	K
T_4	Expansion valve temperature	K
R_{Ti}	Thermal resistance	K/W
T_R,W_C	Compressor power input	W
ω	Angular velocity	rad/s
V_c	Volumetric displacement of compressor	m^3
η_p	Volumetric efficiency of compressor	
υ	Specific volume	m^3/kg
u	Specific internal energy	J/kg
z	Height	m

The dynamic modeling of VCR systems has been subject on interest since the late 1970s, where first principle models were used to describe the heat exchangers. Lumped parameter and moving boundary models are shown in [4]-[6]. In [7], MacArthur initiated a series of works focusing on a distributed parameter formulation. Nonlinear models have shown good approximation [8], [9], [10], but the complexity level increases. Later on, in 1990s, traditional feedback control has been investigated [11], as well as the multivariable control strategy [9], [10], [12], [13]. Other strategies have been developed to select among the degrees of freedom of the control variables, so that an optimal operation is closely obtained [14], [15].

Due to thermal dynamics of VCR systems are typically slower than the mechanical dynamics, the model complexity generally resides in the heat exchangers. Previous literature reviews [16], [17] indicate that most of research efforts are focused on capturing two-phase flow dynamics in the heat exchangers, seeking a balance between simplicity and fidelity. For the purpose of this paper, the four elements of the thermal system will be classified into lumped parameter models. Lumped parameter models refer to models that apply lumped parameter assumptions to the entire heat exchanger or to fluid phases within the heat exchanger (i.e., Individual lumped models for superheated vapor, two-phase fluid, and subcooled liquid), and the result is a set of algebraic and first-order ordinary differential equations to render a simpler model computationally. In this paper, the term "lumped parameter model" means that each heat exchanger is modeled as a single-control volume (or multiple control volumes for each fluid phase). Most of the literature in the lumped parameter classification are early efforts [3], [17], carrying out a computational simplicity to ensure feasible calculation times. These modeling efforts use few dynamic equations and few (lumped) parameters.

Most of analytical models are used to simulate steady-state performance, but leaving out the transient evolution. In this paper, a method for predicting the cooling performance of a VCR system during transient and steady-state is presented. The dynamic model proposed in this paper is similar as in [18], an advantage in this analysis is because the refrigeration system is simplified only four control volumes.

This research is developed in CIDESI Queretaro, for the National Laboratory for Cooling Technologies Research (LaNITeF).

The paper is structured as follows. In Section II, a review of the refrigeration cycle principles is shown. In Section III, an explanation of the linearized model of the VCR system is presented. In Section IV, the balance of mass and energy of each element of the system, as well as the four ordinary differentials equations modelled in Matlab Simulink, are presented. In Section V, the validation of the model is shown. In Section VI, conclusions are described.

II. REFRIGERATION SYSTEMS

Vapor-compression refrigerating systems used in modern refrigerators vary considerably in capacity and complexity, depending on the application. They are hermetically sealed and normally require no replenishment of refrigerant or oil during its useful life. System components must provide optimum overall performance and reliability at minimum cost. In addition, all safety requirements of the appropriate safety standard (i.e., IEC Standard 60335-2-24 [19]; UL Standard 250 [20]) must be accomplished. The fully halogenated refrigerant R-12 was used in household refrigerators for many years. However, because of its strong ozone depletion property, appliance manufacturers have replaced R-12 with environmentally acceptable R134a or isobutene [21].

Design of refrigerating systems for refrigerators and freezers has been improved through new refrigerants and oils, wider use of aluminum, and smaller and more efficient

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

motors, fans, and compressors. These refinements have kept the vapor-compression system in the best competitive position for household application.

A. *Refrigerating circuit*

A VCR system, in its simplest form, consists of two heat exchangers, an expansion valve, and a compressor, Fig. 1. The ideal VCR system consists of four processes:

- Isentropic compression,
- Isobaric heat rejection and condensation,
- Isenthalpic expansion, and
- Isobaric heat absorption and evaporation.

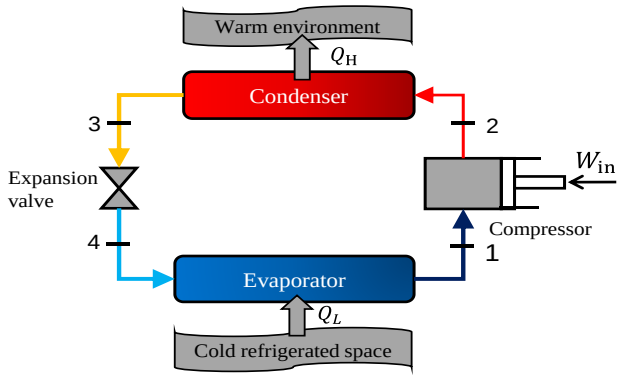


Figure 1. Single stage ideal vapor compression cycle: schematic diagram [21].

Fig. 2 shows the refrigeration cycle on p-h diagram. The refrigerant evaporates entirely in the evaporator and produces the refrigerating effect. Then, it is extracted by the compressor at state point 1, compressor suction, and is compressed isentropically from state point 1 to 2. Next, it is condensed to liquid in the condenser, and the latent heat of condensation is rejected to the heat sink.

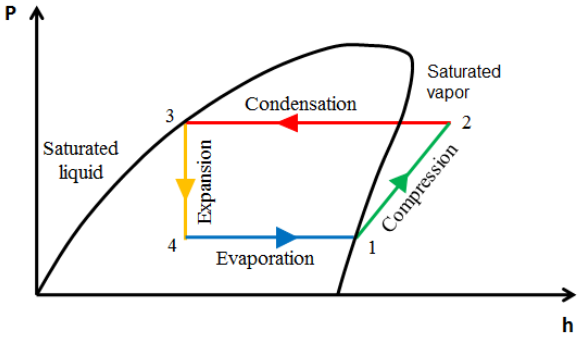


Figure 2. Diagram pressure (p) vs enthalpy (h) [22].

The liquid refrigerant, at state point 3, flows through the expansion valve, which reduces it to the evaporating pressure. In the ideal vapor compressor cycle, the throttling process at the expansion valve is the only irreversible process, usually indicated by a dotted line. Some of the liquid flashes into vapor and enters the evaporator at state point 4. The remaining liquid portion evaporates at the evaporating temperature, thus completing the cycle [21].

Note that energy enters the system through the evaporator (heat load) and through the compressor (electrical input). Thermal energy is rejected to the environment by the condenser and compressor shell. A portion of the capillarity tube is usually soldered to the suction line to form a heat exchanger. Cooling the refrigerant in the capillary tube with the suction gas increases the capacity and efficiency.

A strainer-drier is usually placed ahead of the capillary tube to remove foreign material and moisture. Refrigerant charges of 150 g or less are common. A thermostat (or cold control) cycles the compressor to provide the desired temperatures in the refrigerator. During the off cycle, the capillary tube allows pressures to equalize throughout the system [21], [22].

III. Analysis of the Linearized Model

Fig. 3 summarizes the thermodynamic model of commercial VCR system. The internal temperature of the cooler is a function of the angular speed of the compressor motor and the high temperature of the coolant, which can be obtained from the condenser model. It can be seen the system has several degrees of freedom represented by: the speed of the compressor, the difference of the expansion valve, and the temperature differences of the compressor and the evaporator. The steam compression cooling system can be studied by a system of multiple input and single output states (MISO).

Due to the number of degrees of freedom at the entrance and the singular existence of the exit, the MISO control systems are highly effective; however, the technological limitations of the devices implemented in the VCR systems are responsible for the algorithms developed for these systems to be of the single input and single output (SISO).

This is because commercial devices only apply the control action in the form of activating and disabling a constant speed compressor, since both the expansion valve area differential, the temperature differentials in the condenser and in the evaporator are all almost constant.

In Fig. 3 the input signal can be a change in the angular speed of the compressor motor, and with regard to the control of these devices, several control algorithms have been tested for the internal temperature [2], [23]. The best temperature regulation results are obtained from controls where the thermodynamic model has been simplified but retaining its non-linear nature [2], [23]. If the compressor motor speeds are pre-calculated to keep the internal temperature constant in a permanent state, basic control algorithms such as the PID have been applied to maintain the speed of the compressor, whose dynamics are modeled as a second order system [24].

The technology developed and applied in most vapor compression systems have constant speed compressors, and therefore, the control of the temperature inside the cooler is carried out through on-off control actions [25]. Also, there are some applications of on-off controls adjusted according to the thermodynamic model defined in Fig. 3 [26], [27].

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

These controls have been developed to be able to propose algorithms for the optimization of energy consumption, because is the main objective to cover for most research groups working in the area of cooling systems.

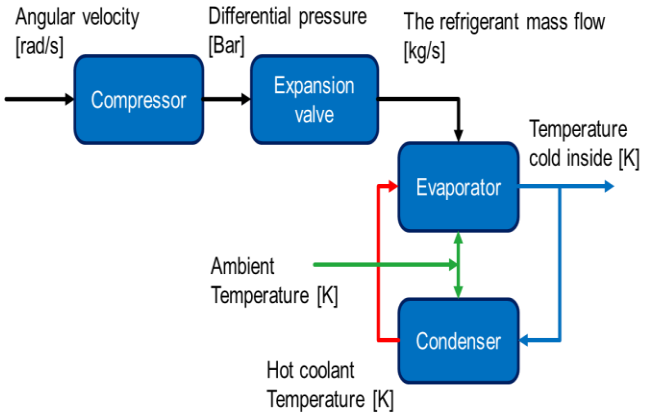


Figure 3. Thermodynamic model of commercial coolers based on vapor compression [2].

Considering the dynamic response of commercial coolers based on vapor compression, various specialists have decided to simplify the dynamic model based on the property of retain heat and the thermal resistance of its barriers [2], [28], [29]. These models consider the evaporator element as a source of heat, which can extract or inject energy from the thermodynamic system. Thus, *C* is considered as the thermal capacity of the volume inside the cooler and, *R* is heat resistance of its barriers. The linearized mathematical model is established in (1).

$$Q(t) = C\frac{dT_i}{dt} + [T_i - T_e]/R \quad (1)$$

Q(t) is the heat injected or absorbed by the source, in this case the evaporator. T_i and T_e are considered internal and external temperatures, respectively. The model is easily recognizable as a first-order dynamic system and the internal temperature depends on the motor compressor angular speed, as shown in Fig. 4.

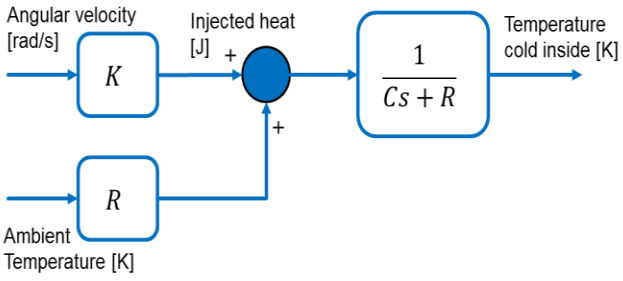


Figure 4. Linearized model dependig on the speed of the compressor [1].

In the same approach of the open-loop model, there are several control strategies implemented to regulate the internal temperature, by applying a control law to manipulate the compressor speed.

Such strategies are tuned considering the parameters of the linearized model [30], [31]. Similar analysis is employed to define the conditions of the hysteresis [32], [33].

The typical tools to model dynamics of the vapor compressor systems are Neural Networks [34], [35], fuzzy logic [36], and genetic algorithms [37], which are heuristic algorithms.

Artificial intelligence techniques, such as AAN (artificial neural networks), fuzzy theory and expert system, belong to non-model method. They do not need mathematical models but have high adaptability. The artificial intelligence technique was used to predict the performance of refrigeration and air conditioning appliances [38], [39], [40], [41]. But, the unsolvable problem in using such a method is because of the imperfection of the artificial intelligence technique itself and the limitation of the user's understanding.

The conventional mathematical model method has been theoretically studied and practically applied for many years. With the mathematical model is more likely to ensure the qualitative precision of simulation than the intelligent method. It is a good way to combine the conventional mathematical method with the intelligent method together in order to take the advantages and to avoid the shortages of both methods. When the modern artificial intelligence techniques are combined with mathematical models of refrigeration systems, called as model based-intelligent simulation [42], the simulation software has certain "intelligence" for simulating the actual complex objectives and becomes more practical.

With the model-based intelligent simulation method, the predicted result of the model can well fit the experimental data as its empirical coefficients can be adapted by an artificial intelligence module. The training task of the artificial intelligence module will be reduced, and the training speed can be accelerated if the calculated results by the theoretical model are used as the initial or prior assumed values for the artificial intelligence module. The adjustment process of the empirical coefficients in the mathematical model can be converted into the training process of the artificial intelligence module, and can be executed by the computer itself.

In this way, less or even no artificial adjustment is needed in the simulation, and self-learning, self-adjusting and self-adapting function can be realized. On the other hand, the number of input parameters and the dimension of the artificial intelligence module will be decreased since many important parameters including configuration parameters are already included in the mathematical model.

Those complicated, empirical and even uncertain factors can be incorporated into the artificial intelligence module and so the mathematical model can be simplified [42].

Fig. 5 shows the simulation process of volumetric efficiency with compound fuzzy model.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

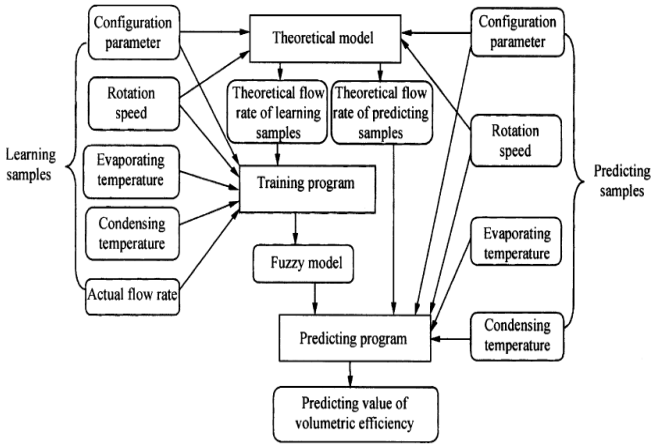


Figure 5. Fuzzy simulation for volumetric effiency of refrigeration compressor [43].

The disadvantage of these heuristic models is that internal variables do not have any physical interpretation, although they have an effective estimation of internal temperature. The non-linearity influences the performance of the linearized model described above.

IV. DYNAMIC MODEL OF THE VCR SYSTEM

Henceforth, the refrigeration cycle of a reciprocating refrigerating system is considered a closed cycle, and the system is operated in steady state (i.e., in an equilibrium state).

A. Energy Conservation Law

Therefore, according to the principle of continuity of mass and energy balance, the mass flow rates of refrigerant flowing through the evaporator, compressor, condenser and expansion (float) valve must all be equal. In addition, the total amount of energy supplied to the refrigeration system must be approximately equal to the total energy rejected from the system.

The flow is continuous, and the properties of the refrigerant at any point in the system do not vary over time. Therefore, during the design of a refrigeration system, the system components selected should have approximately equal mass flow rates of refrigerant at steady conditions [22], [44].

For the general case of multiple mass flows with uniform properties in and out of the system, the energy balance can be written (2):

$$\dot{Q} + \dot{W} = \dot{m}\left[\left(h_o - h_i\right) + \frac{v_o^2 - v_i^2}{2} + g\left(z_o - z_i\right) \right] \quad (2)$$

In (2), $\dot{Q}$ is transferred to the system by the surrounding heat flow, $\dot{W}$ is the work performed by the electric motor of the compressor, h is enthalpy function that is associated with the sum of the internal energy and the work flow, $u + Pv$, the linear kinetic energy $v^2/2$ and potential energy is caused by attractive forces existing between molecules, or the elevation of the system gz. The subscripts i and o refer to the initial and final states, respectively. In the absence of appreciable variations of kinetic and potential energy, the equation above reduces to:

$$\dot{Q} + \dot{W} + \dot{Q}_{Pi} = \dot{m}_r c_p \left(T_o - T_i\right) \quad (3)$$

This relationship is based on the consideration that all variables along the finite volume (control volume) are homogeneous. Equation (3) applies to each volume of control of the four stages of the refrigeration cycle. In (3), $\dot{m}_r$ is the mass of the refrigerant in circulation, c_{pi} is the specific heat of the refrigerant at room temperature (considered constant), see Table II. Q_{Pi} represents the heat loss or heat generated by the system, and corresponds to the Newton's law of cooling, dT/dt is the cooling speed. For this analysis, the refrigerant temperature change is as shown in [45].

$$Q_{Pi} = C_{Ti} \frac{dT_i}{dt} + \frac{\left(T_i - T_o\right)}{R_{Ti}} \quad (4)$$

In the volume control 1, C_{Ti} is the thermal capacitance to the interior of the space confined to the evaporator element and for other elements is the capacitance of the material, expressed as $C_{Ti} = mc_{p,m}$. Thus, m is the mass of the elements and $c_{p,m}$ is the heat capacity of the material, T_i is the temperature inside the system, and T_a is the room temperature. The enthalpy of the model is considered for an ideal gas in (5).

$$\Delta h = c_p (T_o - T_i) \quad (5)$$

Fig. 6 shows the physical quantities involving the system analyzed.

The following simplifications are considered:

- The physical properties related to the refrigerant are considered uniform in the heat exchanger transversal section.
- The refrigerant liquid and vapor phases are in thermodynamic equilibrium.
- The heat exchangers have a perfect thermal insulation.
- The axial heat conduction in the pipes is ignored.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

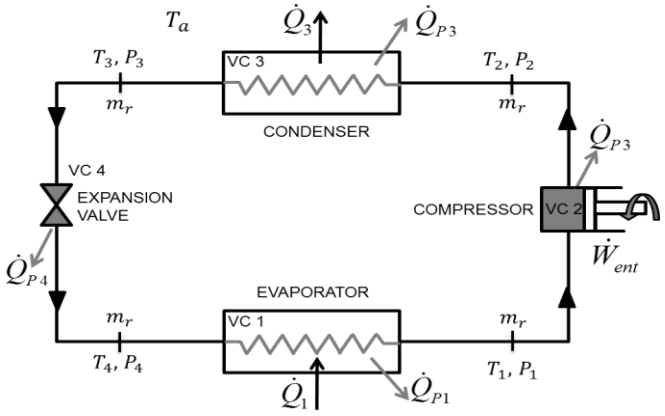


Figure 6. The Dynamic Model Analyzed Scheme [1].

B. Evaporator Mathematical Model

The term $\dot{Q}$ is considered as the heat removed by the refrigerant from the confined space, which is also known as thermal load. Considering the work $\dot{W} = 0$ in equation (3) arises:

$$\dot{Q}_1 + \dot{Q}_{Pi} = \dot{m}_r c_p (T_1 - T_4) \quad (6)$$

Substituting (4) in (6), provides the following equation:

$$\frac{dT_1}{dt} = T_1\left(\frac{c_p \dot{m}_r}{C_{T1}} - \frac{1}{C_{T1}R_{T1}}\right) - T_4\frac{c_p \dot{m}_r}{C_{T1}} + \frac{T_a}{C_{T1}R_{T1}} - \frac{\dot{Q}_1}{C_{T1}} \quad (7)$$

C. Mathematical Model of Compressor

Motor shaft dynamics are modeled from an angular momentum balance between the driving and braking torques. The torque-speed characteristics of the motor are obtained from manufacturer's specifications. The driving torque and speed of the compressor are known.

The following hypothesis is established; there is no friction in the compressor and is an adiabatic process $\dot{Q}_1 = 0$. The compressor equation is as follows [18].

$$\dot{W}_C = \frac{\mathrm{n}}{(n-1)}\eta_p V_c \omega_c p_1 \left[1 - \left(\frac{p_2}{p_3}\right)^{\frac{n-1}{n}}\right] \quad (8)$$

The torque provided by the motor of the compressor $\dot{W}_C = \dot{T}_R$ and ω is the angular velocity of the shaft. Using equation (3),

$$\dot{W}_R + \dot{Q}_{P2} = \dot{m}_r c_p (T_2 - T_1) \quad (9)$$

Substituting (4) in (9), the resulting balance of energy is:

$$\frac{dT_2}{dt} = -T_1\frac{c_p \dot{m}_r}{C_{T2}} + T_2\left(\frac{c_p \dot{m}_r}{C_{T2}} - \frac{1}{C_{T2}R_{T2}}\right) + \frac{T_a}{C_{T2}R_{T2}} - \frac{\dot{W}_1}{C_{T2}} \quad (10)$$

D. Mathematical Model of Condenser

Only the superheat vapor is considered. The vapor phase is considered in the thermal equilibrium and moving in the same velocity. Work is not performed in this element $\dot{W} = 0$. $\dot{Q} = 0$ it is the heat released into the environment. From equation (3), it is obtained the following,

$$\dot{Q}_3 + \dot{Q}_{P3} = \dot{m}_r c_p (T_3 - T_2) \quad (11)$$

Substituting (4) into (11), you get the following:

$$\frac{dT_3}{dt} = -T_2\frac{c_p \dot{m}_r}{C_{T3}} + T_3\left(\frac{c_p \dot{m}_r}{C_{T3}} - \frac{1}{C_{T3}R_{T3}}\right) + \frac{T_a}{C_{T3}R_{T3}} - \frac{\dot{Q}_3}{C_{tT3}} \quad (12)$$

E. Mathematical Model of the Expansion Valve

It is considered no interaction of work and heat $\dot{W} = 0$ and $\dot{Q} = 0$. The expansion is isenthalpic. Thus, from equation (3), it is obtained the following,

$$\dot{Q}_{P4} = \dot{m}_r c_p (T_3 - T_4) \quad (13)$$

Substituting (4) into (13), the balance of energy gives:

$$\frac{dT_4}{dt} = -T_3\frac{c_p \dot{m}_r}{C_{T4}} + T_4\left(\frac{c_p \dot{m}_r}{C_{T4}} - \frac{1}{C_{T4}R_{T4}}\right) + \frac{T_a}{C_{T4}R_{T4}} \quad (14)$$

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

F. State-space Representation of VCR System

Previously, equations (7), (10), (12) and (14) have been presented as first-order linear ordinary differential equations, with the temperature of the refrigerant as the system output. The following expressions are expressed in matrix form [2], [46]:

$$\begin{aligned} \dot{T}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{15}$$

Equations (7), (10), (12) and (14) must be replaced in (15):

$$\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \\ \dot{T}_3 \\ \dot{T}_4 \end{bmatrix} = \begin{bmatrix} \left(\frac{c_p \dot{m}_r}{C_{T1}} - \frac{1}{C_{T1}R_{T1}}\right) & 0 & 0 & -\frac{c_p \dot{m}_r}{C_{T1}} \\ -\left(\frac{c_p \dot{m}_r}{C_{T2}}\right) & \left(\frac{c_p \dot{m}_r}{C_{T2}} - \frac{1}{C_{T2}R_{T2}}\right) & 0 & 0 \\ 0 & -\left(\frac{c_p \dot{m}_r}{C_{T3}}\right) & \left(\frac{c_p \dot{m}_r}{C_{T3}} - \frac{1}{C_{T3}R_{T3}}\right) & 0 \\ 0 & 0 & -\frac{c_p \dot{m}_r}{C_{T4}} & \left(\frac{c_p \dot{m}_r}{C_{T4}} - \frac{1}{C_{T4}R_{T4}}\right) \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{1}{C_{T1}R_{T1}} & -\frac{1}{C_{T1}} & 0 & 0 \\ \frac{1}{C_{T2}R_{T2}} & 0 & -\frac{1}{C_{T2}} & 0 \\ \frac{1}{C_{T3}R_{T3}} & 0 & 0 & -\frac{1}{C_{T3}} \\ \frac{1}{C_{T4}R_{T4}} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_a \\ \dot{Q}_1 \\ \dot{T}_R \\ \dot{Q}_3 \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} + 0$$

G. Numerical prediction

The mathematical model obtained in the previous section for the prediction of the dynamic behavior, was programmed in Matlab Simulink. The typical values of input parameters are presented in Table II. The input parameters include refrigerant type, environmental temperature and the initial temperature of the compartment, etcetera.

The values of c_{pi} of the refrigerant were defined according the average of the phase in relation to *p* vs *h* diagram.

H. Modeling

Considering the thermodynamic cycle starts-up when the system is powered by the angular speed of the electric motor of the compressor, the flow of the refrigerant quickly tends to a steady state. It can be noted that the refrigerant tends to decrease its temperature at the evaporator element, trying to keep the relationship of equilibrium of pressure and temperature. Fig. 7 shows the behavior of the refrigerant in the evaporator [47].

TABLE II. Input Values of MATLAB Algorithm

Value	Units	Value	Units
C_{p1} = 1330	J/kg-K	R_{T1} = 0.090	K/W
C_{p2} = 1400	J/kg-K	R_{T2} = 0.025	K/W
C_{p3} = 1138	J/kg-K	R_{T3} = 0.048	K/W
C_{p4} = 1318	J/kg-K	R_{T4} = 3.20	K/W
C_{T1} = 4500	J/K	Q_1 = 195	W
C_{T2} = 2500	J/K	Q_3 = -200	W
C_{T3} = 1250	J/K	T_a = 298	K
C_{T4} = 500	J/K	T_R = 5	W
m_r = 0.000035	kg/s		

The behavior of the refrigerant in the compressor where the temperature and pressure increase, the curve tends to rise starting from room temperature as shown in Fig. 8.

The refrigerant follows its course towards the condenser, where the heat extracted from the confined spaces towards the environment that surrounds it is released, we can see that the coolant temperature is approximately the same, receiving compressor discharge line as it is located in the area of high *p* and *t*, Fig. 9 [2], [47].

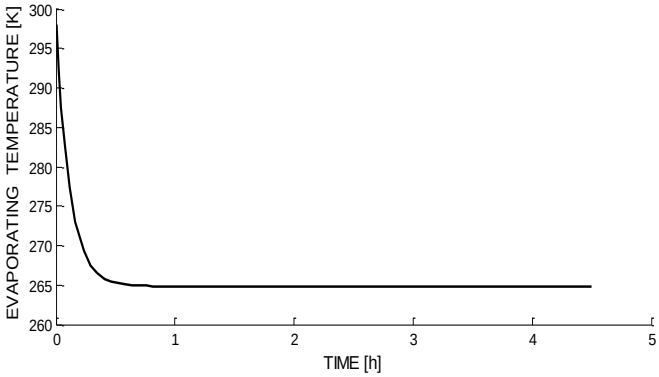


Figure 7. Evaporation temperature of the refrigerant.

In the capillary tube or expansion valve, temperature and refrigerant pressure decreases due to strangulation, to a lower temperature so that heat transfer can be done appropriately. Fig. 10 shows the temperature of the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

refrigerant which tends to decrease from ambient temperature.

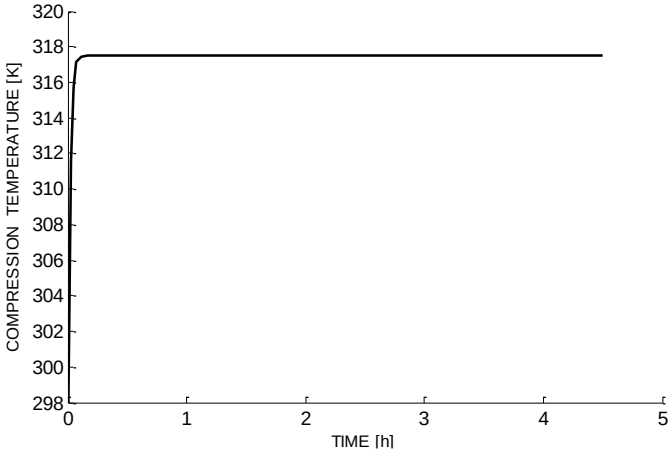


Figure 8. Behavior of the refrigerant in the compression process.

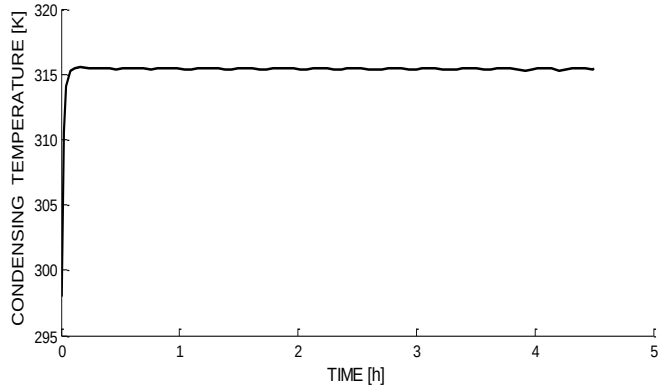


Figure 9. Behavior of the temperature in the condensing process.

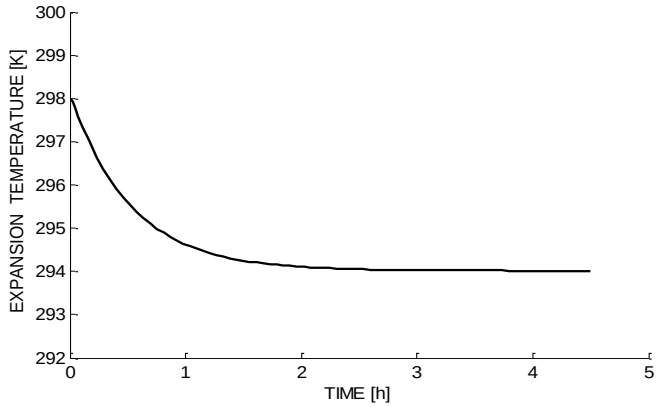


Figure 10. Behavior of the temperature in the strangulation process.

In order to have a frame of reference with which to validate the results obtained from the predicted model are compared with the manufacturer's specification of refrigerant R134a.

V. VALIDATION OF THE MODELING

In this Section, the estimated model is validated with specifications of the refrigerant R134a, since it is the fluid that is most often used in domestic refrigerators.

The following figures show the dynamic evolution of the refrigeration cycle of two critical variables during 4.5 hours in the heat exchangers: temperature and pressure starting from environmental temperature; then, it approaches the steady state.

Fig. 11 shows the temperature evolution in the compartment of the refrigerator. The model is compared with Embraco Data [47], using R-143a. It can be observed that takes about 10 minutes in the transient state, then the room temperature reaches the steady state condition.

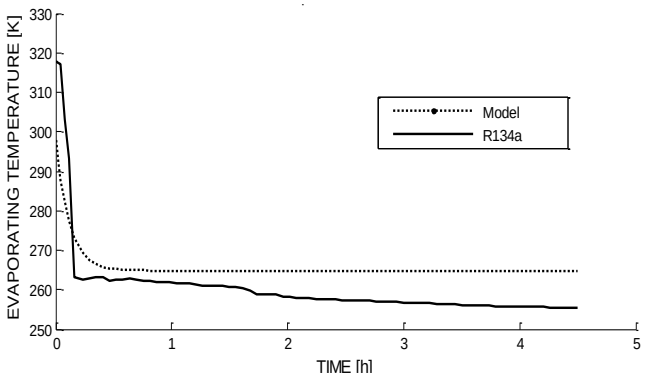


Figure 11. Behavior of the temperature of the evaporation system.

Fig. 12 shows the pressure evolution in the compartment of the refrigerator. It can be observed the same approach when it starts from environmental temperature related to the manufacturer's refrigerant. Fig. 12 shows that the pressure reaches the steady state condition, which replicate the process 4 to 1 in the *p* vs *h* diagram, in Section II.

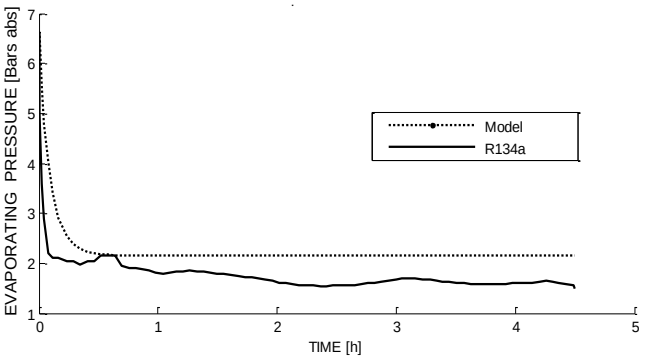


Figure 12. Behavior of the pressure of the evaporation system.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Fig. 13 shows the behavior of the refrigerant temperature in the condenser element. The temperature of R134 reaches 335 K in the first 5 minutes and the estimated model only reaches 315 K. This value is acceptable since it is above the environment and there is a difference greater than 7 K that guarantees the transfer of energy to the surroundings.

The temperature of the estimated model is enough to guarantee the operation of vapor compression equipment that works with natural convection. If a refrigerator is considered to work with forced convection, this reached value guarantees the release of enough energy for the refrigerant to change of phase from gaseous state to liquid state.

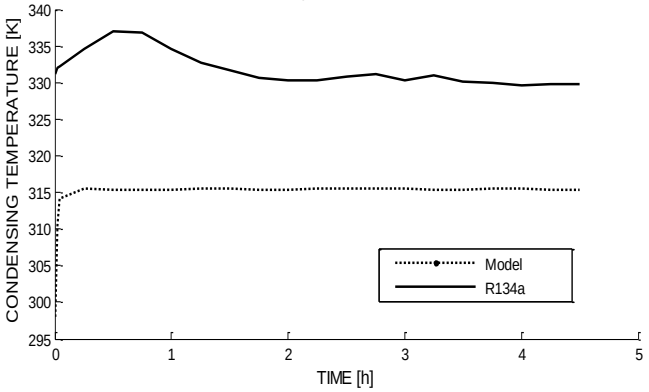


Figure 13. Behavior of the condenser temperature.

Fig. 14 shows the behavior of the refrigerant pressure. The fluid maintains its relationship T and p to reach condensation temperature above the environment and must also maintain a high pressure, 17 bars of the R134a.

For the estimated model, the pressure of 10.5 bars is reached. In the case of a domestic refrigerator we observe that the value reached in the pressure compliance for optimal performance.

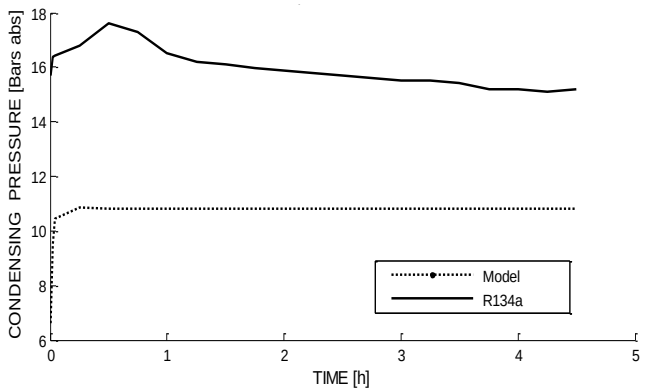


Figure 14. Behavior of the pressure in the condenser.

A good agreement between the estimated model and measure values from [22], [47] was observed for the whole period. In the evaporator, during the first 10 minutes, the pressure decreases from 6.5 bars to 2.2 bars, then tends to decrease slowly until reach the steady state operation.

The behavior of the evolution of the cooling curve are similar, although there is a difference in terms of temperatures, however, are considered within the working area in the diagram pressure-enthalpy [22].

VI. Conclusion and Future Work

In this work, a methodology to model the dynamic behavior of the refrigerator has been developed. This would act as basis for improvements on modeling domestic refrigerators, using lumped transient model. The lumped parameter modeling will reduce the overall cycle time used to predict the compartment temperature, decreasing the experimental effort.

The results agree qualitatively with temperature time evolution shown in the literature. The transient analysis of domestic refrigerators, using the computer program developed in this paper, will be further validated with experimental testing. An advantage of the simplified model here described is the possibility of using personal computer due to the relatively low computational effort required for the calculations.

A valid starting point for the study of cooling systems applying state variables has been introduced in this paper. The first improvement strategy is a state feedback, since the model considers a linear approximation; however, there are nonlinearities, so a parameters identification algorithm and adaptive control strategies are required.

Acknowledgments

The authors thank the Mexican Council for Science and Technology (CONACYT) for its support of the student's scholarships No. 599179, 492895, 437556, 555423 and 486129 of the graduate students; also because of the support through the consolidation project for the National Research Laboratory on Cooling Technology (LaNITeF) budget 293784. These results were gotten by the synergy of both projects the LaNITeF consolidation as well as the P02 of the Mexican Center for Innovation in Eolic Energy (CEMIE Eólico), supported also by the secretary of Energy, through the budget for Sustainable Energy.

References

[1] G. Domínguez, L. A. Montoya, E. E. Rodríguez, and H. Gamez, "Dynamic Behavior Model for Cooling System Based on Vapor Compression," Proceedings of the 30th International Conference on Systems (ICONS 2018), Apr. 22, 2018, Paper No. 48003, pp. 57-62. ISSN: 2308-4243.

[2] G. Domínguez, E. E. Rodríguez, J. H. Pérez, and H. Gamez, "Modelo dinámico del circuito térmico de un aparato comercial de refrigeración por compresión de vapor," Memorias del Congreso Nacional de Control Automático 2016, Querétaro, México, Septiembre 28-30, 2016. [online]. Available from: http://amca.mx/memorias/amca2016/Articulos/0043.pdf

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[3] B. P. Rasmussen, "Dynamic Modeling for Vapor Compression systems," Part I literature review. HVAC and Research, vol. 18, No. 5, pp. 934-955, 2012, https://doi.org/10.1080/10789669.2011.582916

[4] G. L. Wedekind, B. L. Bhatt, and B. T. Beck, "A system mean void fraction model for predicting various transient phenomena associated with two phase evaporating and condensing flows," International Journal of Multiphase Flow, vol. 4, No. 1, pp. 97–114, 1978. https://doi.org/10.1016/0301-9322(78)90029-0

[5] M. Dhar and W. Soedel, "Transient analysis of a vapour-compression refrigeration system parts 1 and 2," In: 15th International Congress of Refrigeration, 1979.

[6] J. Chi and D. A. Didion, "Simulation model of the transient performance of a heat pump," International Journal of Refrigeration, vol. 5, No. 3, pp. 176–184, 1982.
https://doi.org/10.1016/0140-7007(82)90099-8

[7] J. W. MacArthur and E. W. Grald, "Prediction of cyclic heat pump performance with a fully distributed model and a comparison with experimental data," ASHRAE Transactions, vol. 93, No. 2, pp. 1159-1178, 1987.

[8] S. Bendapudi and J. E. Braun, "A Review of Literature on Dynamic Models of Vapour Compression Equipment," Technical Report HL2002-8. Ray W. Herrick Laboratories, Purdue University, 2002.

[9] B. P. Rasmussen and A. G. Alleyne, "Dynamic Modeling and Advanced Control of Air Conditioning and Refrigeration Systems," Dept. of Mechanical Engineering, University of Illinois, 2005.

[10] L. C. Schurt, Ch. J. L. Hermes, and A. T. Neto, "A model-driven multivariable controller for vapor compression refrigeration systems". International Journal of Refrigeration, vol. 32, No. 7, pp. 1672-1682, 2009.
https://doi.org/10.1016/j.ijrefrig.2009.04.004

[11] X. D. He, S. Liu, and H. H. Asada, "Modeling of vapor compression cycles for multivariable feedback control of HVAC systems," ASME Journal of Dynamic Systems, Measurement and Control, vol. 119, No. 2, pp. 183-191, 1997. doi:10.1115/1.2801231

[12] L. C. Schurt, C. J. L. Hermes, and A. T. Neto, "A model driven multivariable controller for vapor compression refrigeration systems," International Journal of Refrigeration, vol. 33, No. 7, pp. 1672-1682, 2009.
https://doi.org/10.1016/j.ijrefrig.2009.04.004

[13] B. P. Rasmussen, A. Alleyne, C. Bullard, P. Hrnjak, and N. Miller, "Control-Oriented Modeling and Analysis of Automotive Transcritical AC System Dynamics," Proc. American Control Conference, vol. 4, pp. 3111-3116, 2002.
DOI: 10.1109/ACC.2002.1025267}

[14] J. B. Jensen and S. Skogestad, "Optimal operation of simple refrigeration cycles: Part I: Degrees of freedom and optimality of sub-cooling," Computers and Chemical Engineering, vol. 31, No. 5-6, pp. 712–721, 2007a.
https://doi.org/10.1016/j.compchemeng.2006.12.003

[15] J. B. Jensen and S. Skogestad, "Optimal operation of simple refrigeration cycles: Part II: Seleccion of controlled variables," Computers and Chemical Engineering, vol. 31, No. 12, pp. 1590–1601, 2007b.
https://doi.org/10.1016/j.compchemeng.2007.01.008

[16] J. Lebrun and J. P. Bourdouxhe, "Reference guide for dynamic models of HVAC equipment," Project 738-TPR, 1998.

[17] S. Bendapudi and J. E. Braun, "Development and validation of a mechanistic, dynamic model for a vapor compression centrifugal chiller," Report No. 4036-4 ASHRAE, Atlanta G.A. 2002b.

[18] J. V. C. Vargas and J. A. R. Parise, "Simulation in transient regime of a heat pump with closed-loop and on-off control," International Journal of Refrigeration, vol. 18, No. 4, pp. 235-243, May 1995.
https://doi.org/10.1016/0140-7007(95)96862-Z

[19] IEC, Household and similar electrical appliances – Safety: Particular requirements for refrigerating appliances, ice-cream appliances and ace-makers. Standard 60335-2-24. International Electrotechnical Commission, Geneva. 2007.

[20] UL. Household refrigerators and freezers. ANSI/UL Standard 250, CAN/CSA Standard C22.2. Underwriters Laboratories, Northbrook, IL. 1993.

[21] Handbook-Refrigeration, ASHRAE Chapter 17, pp. 262-264, 2010.

[22] Handbook-Fundamentals, ASHRAE, Chapter 2, pp. 9-11, 2009.

[23] N. Lu and D. P. Chassin, "A state-queueing model thermostatically controlled appliances," Power Transactions on, vol. 19 (3). 2004.
DOI: 10.1109/TPWRS.2004.831700

[24] SECOP, "Variable speed drive compressor NLV-F and BD150F," operating instructions. DES.S.300.02.02. Seccop 2013, SLV Compressors with 105N4XX series controller, [online]. Available from: http://www.secop.com/news/solutions/slv-compressors.html 2017.07.25

[25] LG, LFX21960 Series User Manual "Botton freezer refrigerator" 2009. [Online]. Available from:
https://www.manualslib.com/manual/93518/Lg-Lfx21960-Series.html 2018.06.28

[26] S. Soyguder, M. Karakose, and H. Alli, "Design and simulation of selft-tunning PID-type fuzzy adaptive control for an expert HVAC systems," Expert Systems with Applications, vol. 36, pp. 4566-4573, 2009.
DOI:10.1016/j.eswa.2008.05.031

[27] M. Affenzeller and S. Wagner, "A Selft-adaptive Model for Selective Pressure Handling within the Theory of Generic Algorithms," Computer Aided Systems Theory, vol. 2809, pp. 384-393, 2003.
DOI:10.1007/978-3-540-45210-2_35

[28] M. Stadler, W. Krause, and M. Sonnenschein, "Modelling and evaluation of control schemes for anhancing load shift of electricity demand for cooling devices," Enviromental Modelling and Software, vol. 24, No. 2, pp. 285-295, 2009.
https://doi.org/10.1016/j.envsoft.2008.07.003

[29] M. Ouzzane and Z. Aidoun, "Model development and numerical procedure for detailed ejector analysis and design," Applied Thermal Engineering, vol. 23, No. 18, pp. 2337-51, 2003. https://doi.org/10.1016/S1359-4311(03)00208-4

[30] Secop, 2011, SLV Compressors with 105N46XX series controller, [online]. Available from: https://www.secop.com/updates/news-secop-slv-compressors-with-105n46xx-series-controller/ 2017.07.25

[31] J. Dieckmann, D. Westphalen, W. Murphy, P. Sikkirm, and C. Rieger, "Application of a variable speed compressor to a residential no-frost freezer," TIAX LLC, Jan 27, 2004. [online]. Available from:
https://www.slideserve.com/Mia_John/application-of-a-variable-speed-compressor-to-a-residential-no-frost-freezer

[32] J. M. León, Design of automated reactor cooling system, University of Carabobo. Venezuela. 2003.

[33] W. R. Chang, D. Y. Liu, S. G. Chen, and N. Y. Wu, "The components and control methods for implementation of inverter-controlled refrigerators / Freezer," Purdue University, N. Y., 2004. http://docs.lib.purdue.edu/iracc

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[34] M. V. Aydinalp, V. I. Ugursal, and A. S. Fung, "Modeling of appliance, lighting, and space cooling energy consumption in the residential sector using neural networks," Applied Energy, vol. 71, pp. 87-110, 2002.
https://doi.org/10.1016/S0306-2619(01)00049-6

[35] M. V. Aydinalp, V. I. Ugursal, and A. S. Fung, , "Modeling of the space and domestic hotwater heating energy consumption in the residential sector using neural networks," Applied Energy, vol. 79, No. 2, pp. 159-178, 2002.
https://doi.org/10.1016/j.apenergy.2003.12.006

[36] M. Knezic, Fuzzy Logic Apparatus Control, United States Patern, Number 5,261,247, 1993.

[37] M. Mraz, "The Design of Intelligent Control of a Kitchen Refrigerator," Mathematic and Computing in Simulation, Elsevier, vol. 56, No. 3, pp. 259-267, 2001.
https://doi.org/10.1016/S0378-4754(01)00281-6

[38] G. Diaz, M. Sen, K. T. Yang, and R. L. McClain, "Simulation of heat exchanger performance by artificial neural networks," HVAC&R Research, vol. 5, No. 3, pp. 195-208, 1999.

[39] A. Pacheco-Vega, M. Sen K. T. Yang, and R. L. McClain, "Neural network analysis of fin-tube refrigerating heat exchanger with limited experimental data," International Journal of Heat and Mass Transfer, vol. 44, No. 4, pp.763-770, 2001. https://doi.org/10.1016/S0017-9310(00)00139-3

[40] D. J. Swider, M. W. Brownie, P. K. Bansal, and V. Kecman, "Modeling of vapour-compression liquid chiller with neural networks," Applied Thermal Engineering, vol. 21, No. 3, pp. 311-329, 2001.
https://doi.org/10.1016/S1359-4311(00)00036-3

[41] H. Bechler, M. W. Brownie, P. K. Bansal, and V. Kecman, "New approach to dynamic modelling of of vapour-compression liquid chillers: artificial neural networks," Applied Thermal Engineering, vol. 21, No. 9, pp. 942-953, 2001. https://doi.org/10.1016/S1359-4311(00)00093-4

[42] G. L. Ding and C. L. Zhang, "Intelligent Simulation of Refrigeration and Air Conditioning Appliances," Science Press, Beijing 2002 (in Chinese).

[43] G. L. Ding, C. L. Zhang, T. Zhan, and H. Li, "Compound fuzzy model for thermal performance of refrigeration compressors," Chinese Science Bulletin, vol. 45 No. 14, pp. 1319-1322, 2000.

[44] W. Kenneth and E. R. Donald, Thermodynamics, 6th ed., Mc Graw Hill: Spain, pp. 190-192, 2001.

[45] W. Bolton, Mechatronics, electronic control in the mechanical and electrical engineering systems, 5th ed., Alfaomega: Mexico D. F., pp. 261-264, May 2013.

[46] K. Ogata, Modern control engineering, 5th ed., Pearson, pp. 136-139, 2010.

[47] Embraco, technical information. [Online]. Available from: http://www.embraco.com/DesktopModules/DownloadsAdmin/Arquivos/xd0vsAjtYw.pdf 2017.09.05

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Dynamics Analysis of the Rijke Tube Thermoacoustics

Mathematical Modeling using Luikov and Wave Equations with its Experimental Validation

I. Mejía Alonso
Thermoacoustics for Cooling
Engineering Center for Industrial Development
Querétaro, México
imejia@cidesi.edu.mx

E. E. Rodríguez Vázquez
National Research Laboratory for Cooling Technology
Engineering Center for Industrial Development
Querétaro, México
eloy.rodriguez@cidesi.edu.mx

C. A. Núñez Martín
Dynamic Optimization of Cooling Devices
Engineering Center for Industrial Development
Querétaro, México
canunez@posgrado.cidesi.edu.mx

L. A. Montoya Santiyanes
Rotordynamics for Cooling
Engineering Center for Industrial Development
Querétaro, México
lmontoya@posgrado.cidesi.edu.mx

H. J. Zúñiga Osorio
Flow Dynamics for Cooling
Engineering Center for Industrial Development
Querétaro, México
hzuniga@posgrado.cidesi.edu.mx

***Abstract*— This paper deals with the mathematical modeling for the thermoacoustic phenomena in a simple geometry combustion chamber represented as the Rijke tube. The methodology developed to get an analytical model from the Luikov equations for the dynamical behavior of both variables of interest the temperature and pressure distributions into the cited chamber is described briefly. Through this model, the time and space Eigenvalues are calculated to interpret them in terms of the combustion chamber geometry; also the spatial distribution (modal shapes or Eigenvectors) of both variables into the confined volume in the chamber are analyzed by the boundary conditions. The obtained analytical model from the Luikov modified equations is complemented with an experimental modeling for the dynamics of the same variables of interest, but based on the wave equation and considering the same boundary conditions. To validate both models for the thermoacoustics dynamics, a Rijke tube experiment was developed, by getting the experimental data to support the analytical hypothesis from the synthetized models. The main purpose of this paper is to compare the two mathematical modeling methodologies to know the set of assumptions needed to have a real idea of the fundamental modal shapes for the temperature and pressure in this combustion chamber geometry when the thermoacoustic phenomena is presented.**

Keywords-thermoacoustics; Rijke tube experiment; Luikov equations and wave equation.

NOMENCLATURE

X	: a dimensional Cartesian coordinate
T	: time
ρ	: density
T	: temperature
Q	: thermal energy
q	: acoustical perturbation
a_m	: diffusion coefficients for gases
a_q	: diffusion coefficients for temperature
C_q	: heat capacity
λ	: heat of phase change
ε	: ratio of vapor diffusion coefficient
k	: wavenumber
κ	: bulk modulus for the air
ξ	: displacement
P	: pressure
c	: sound speed in the medium
ω	: angular frequency

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

I. INTRODUCTION

Modern society needs to be conscience about all energy conversion process which affects the natural environment health. There are several efforts to maximize the natural energetic resources use by the synergy of thermodynamic technologies that combines systems for cooling and heating with several applications [1]. However, the achieved magnitude of both the worldwide population as well as the energy consumption for the ambient conditioning for living or for the perishables products conservation are considerable with the current natural energetic resources availability [2] [3].

In terms of the power generation technology which takes advantage from the synergy of two complemented thermodynamically cycles, we have the gas turbine technologies. The ammonia gas turbines combine the absorption ammonia cycle to diminish the surface temperature of the combustion chamber housing; then this transported energy heats the ammonia-air mixture to increase the combustion performance into the cited chamber [4]. This technology is applied for naval transportation where the ammonia use does not affect the population health and the engine performance is critical because of the fuel availability. In this context, the new trend for this technology is to apply it for the home services suppling using the industrial waste as fuel.

Other example of the synergy from a modified thermodynamic cycle is the heat pumps technology; which works based on the vapor compression so it complements several air conditioning systems and this device works in both directions to heat or cooling [5]. Nowadays, because of the time market availability this technology cannot be considered innovation but it is one of the most representative examples of efficiency in terms of the resources use for the ambient conditioning.

In the same technological synergy and because of its thermo-electrical nature Peltier's devices are used for the heating and cooling purposes. In these experimental cases, the energy transportation is mainly performed by conduction (not convection) [6], so their implementations are more punctually located in terms of energy transportation.

Cited technologies have been carefully designed to have the best thermodynamic performance in terms of the variables of interest listed as the temperature, energy and pressure; but these design engineering are based on the steady state behavior of the named variables, where the relationships between them do not considers the dynamical evolution of each one neither and their cross effects.

The thermoacoustic phenomena can be described as the acoustical energy caused from an uncontrolled density of heat in a confined volume [7]. In the gas turbine technology point of view, thermoacoustics means that the heat released in a combustion chamber generates pressure oscillations inside which decreases the engine performance, [8]. This relationship between heat and air pressure (sound waves) was discover by Rayleigh [9] and formulate by Merkli [10].

Thermoacoustic unstable evolution may cause several problems in the fuel system of a gas turbine, therefore it induces: efficiency degradation, premature wear of components and catastrophic failure [11]. Also, thermoacoustic phenomenon combined with the high levels of both pressures and temperatures affects the fuel air mixture process and then de combustion performance, producing also highly polluting particles such as NOx [12].

Nowadays, the state of the art concerned with the thermoacoustic phenomenon effects on the gas turbines performance, has been mainly focused on the development of some techniques as well as the swirling flows [13-14]. These swirling flows provide aerodynamic stability to the combustion process by producing regions of recirculating flows which diminish the flame length and increase the residence time of the reactants in the flame zone [14]. Experimental analysis from combustion test rig using different kind of injectors, constrictors, air-fuel mixes have been performed to find the best technique for combustion system [15]. Also, in terms of the flow, others studies have included premixed fluid and Helmholtz resonator [16].

The gas turbine combustion modelling and simulations, along with the acoustic phenomenon perturbation, implies the necessity of implementing a control algorithm. The basic gas turbine model equations [17] are important for analysis, design and simulation of control system, especially for Combined Cycle Power Plants (CCPP) [18] and [19].

Focusing on acoustics, new passive and active control techniques for such thermoacoustic instabilities have been developed and implemented [20] as well as techniques of: Adaptive Sliding Phasor Averaged Control (ASPAC: adaptation of the valve-commanded fuel phase for flow variations) [21] and Multiscale extended Kalman (MSEK: prediction of the time-delayed states) are promising techniques to reduce the energy consumption produced by pressure oscillations [22], nevertheless, more research is required.

In the same way, into the perturbation control topic, the dynamics model for the thermoacoustic phenomena proposed in this work has been envisaged to be the first part of a continuous control algorithm to diminish its effects into a combustion chamber. This research is carried out for the Engineering Center for Industrial Development (CIDESI) at Queretaro México, by the National Laboratory for Cooling Technologies Research (LaNITeF).

This paper structure has been defined to follow up the methodology proposed to validate a mathematical model for the thermoacoustic dynamics in a Rijke tube combustor; which consist in a theoretical modeling based on the Luikov equations adapted for the confined volume of the gas flowing in the cited chamber, and which is described in Section II. Then in Section III the procedure to develop the analytical model of pressure oscillation trough air columns is presented. Section IV presents the results from a Rijke tube experiment; here the required Fourier analysis is detailed to find the fundamental oscillation mode, where because of the prediction the speed of sound and the medium properties such as density, are required to determined parameter values of pressure distribution model. Section V describes the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

results and validation of both theoretical and analytical models. Conclusions are presented in Section V.

II. THEORETICAL MODEL FOR THE TEMPERATURE AND PRESSURE DISTRIBUTION FROM THE LUIKOV EQUATIONS

Luikov proposed a model for the heat and mass transfer in capillary porous bodies at 1966 [23], which mainly consist in a differential equation system of first order with two degrees of freedom (temperature and mass). This model has been modified several times as an alternative model from the Navier – Stokes equation [24] when its analytics solutions are needed for flow dynamics through different kinds of medias [25] [26] and [27].

A. Analitical solution of the Luikov equations

Originally the Luikov equations were developed for the mass and energy transportation through porous bodies; but in this case, the confined volume into the combustion chamber is not a solid media neither a porous one. Therefore, the hypothesis to use this equation system is based on the concept of finite volume, where the mass transportation does not face an interface between them; it just does at the boundary conditions. The heat transportation between the finite volumes is no considered, because the energy transportation is done mainly by convection.

Then the Luikov model is defined by a first order differential equation system, for temperature and pressure as well as it is:

$$\begin{aligned}\frac{\partial T(t,x)}{\partial t} &= a_q \frac{\partial^2 T(t,x)}{\partial x^2} + \frac{\lambda\varepsilon}{C_q}\frac{\partial \rho(t,x)}{\partial t} + a_1 Q(t,x)\\ \frac{\partial \rho(t,x)}{\partial t} &= a_m \frac{\partial^2 \rho(t,x)}{\partial x^2} + a_m \delta \frac{\partial^2 T(t,x)}{\partial x^2} + a_2 q(t,x)\end{aligned} \tag{1}$$

where T and ρ are the temperature and air density respectively, Q is the thermal energy supplied and q the acoustical perturbation. a_m and a_q are the diffusion coefficients for the air density and temperature. If it is considered that the physical properties, which define the coefficients of (1), are constants and executing the substitution of the first order timed differential for the air density magnitude, equation (1) becomes

$$\begin{aligned}\frac{\partial T(t,x)}{\partial t} &= A_1 \frac{\partial^2 T(t,x)}{\partial x^2} + A_2 \frac{\partial^2 \rho(t,x)}{\partial x^2} + B_1 Q(t,x) + B_2 q(t,x)\\ \frac{\partial \rho(t,x)}{\partial t} &= A_3 \frac{\partial^2 T(t,x)}{\partial x^2} + A_4 \frac{\partial^2 \rho(t,x)}{\partial x^2} + B_3 q(t,x)\end{aligned} \tag{2}$$

Another significant consideration in this proposal is that all function can be separable from the time domain to the space domain, so dimensional functions can be taken as

$$\begin{aligned}T(t,x) &= f(t)g(x)\\ \rho(t,x) &= p(t)h(x)\\ Q(t,x) &= r(t)c(x)\\ q(t,x) &= q(t)\end{aligned} \tag{3}$$

then (2) takes the form of:

$$\begin{aligned}g(x)\frac{df(t)}{dt} &= A_1 f(t)\frac{d^2 g(x)}{dx^2} + A_2 p(t)\frac{d^2 h(x)}{dx^2} + B_1 r(t)c(x) + B_2 q(t)\\ h(x)\frac{dp(t)}{dt} &= A_3 f(t)\frac{d^2 g(x)}{dx^2} + A_4 p(t)\frac{d^2 h(x)}{dx^2} + B_3 q(t)\end{aligned} \tag{4}$$

By defining the next list of matrices

$$\begin{aligned}\mathbf{F}(t)_{2\times 2} &= \begin{bmatrix} f(t) & 0 \\ 0 & p(t)\end{bmatrix}_{2\times 2} \quad \mathbf{G}(x)_{2\times 2} = \begin{bmatrix} g(x) & 0 \\ 0 & h(x)\end{bmatrix}_{2\times 2}\\ \mathbf{R}(t)_{2\times 2} &= \begin{bmatrix} r(t) & 0 \\ 0 & q(t)\end{bmatrix}_{2\times 2} \quad \mathbf{C}(x)_{2\times 2} = \begin{bmatrix} c(x) & 0 \\ 0 & 1\end{bmatrix}_{2\times 2}\\ \mathbf{A}_{2\times 2} &= \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4\end{bmatrix}_{2\times 2} \quad \mathbf{B}_{2\times 2} = \begin{bmatrix} B_1 & B_2 \\ 0 & B_3\end{bmatrix}_{2\times 2}\end{aligned} \tag{5}$$

equation (3) can be organized as:

$$\mathbf{G}(x)\frac{d}{dt}\mathbf{F}(t)\begin{bmatrix}1\\1\end{bmatrix} = \mathbf{A}\frac{d^2}{dx^2}\mathbf{G}(x)\mathbf{F}(t)\begin{bmatrix}1\\1\end{bmatrix} + \mathbf{BC}(x)\mathbf{R}(t)\begin{bmatrix}1\\1\end{bmatrix} \tag{6}$$

The theory for the dynamics analysis for systems is focused on the analysis of the impulse response, because it represents the system transient response when it changes from an equilibrium state (Transfer Function). Therefore, to analyze this system dynamics, it is considered that there is not heat injection (energy source) neither pressure perturbation

$$\mathbf{G}(x)\frac{d}{dt}\mathbf{F}(t)\begin{bmatrix}1\\1\end{bmatrix} = \mathbf{A}\frac{d^2}{dx^2}\mathbf{G}(x)\mathbf{F}(t)\begin{bmatrix}1\\1\end{bmatrix} \tag{7}$$

The different domains functions can be now separated from (7); therefore, by equating them to a constant parameter which has almost the same interpretation than the wave number concept

$$\begin{aligned}\frac{1}{f(t)}\frac{d}{dt}f(t) &= \left[\frac{1}{g(x)}\right]\left[A_1\frac{d^2}{dx^2}g(x) + A_2\frac{d^2}{dx^2}h(x)\right] = -k_1^2\\ \frac{1}{p(t)}\frac{d}{dt}p(t) &= \left[\frac{1}{h(x)}\right]\left[A_3\frac{d}{dx}g(x) + A_4\frac{d}{dx}h(x)\right] = -k_2^2\end{aligned} \tag{8}$$

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

It is shown that the referred equality constants are the time domain Eigenvalues, from

$$\frac{d}{dt} f(t) + k_1^2 f(t) = 0 \qquad \lambda_f = -k_1^2 \qquad f(t) = c_1 e^{-k_1^2 t}$$
$$\frac{d}{dt} p(t) + k_2^2 p(t) = 0 \qquad \lambda_p = -k_2^2 \qquad p(t) = c_2 e^{-k_2^2 t} \tag{9}$$

Into the space domain, the analytical solution needs to solve the next differential equations system

$$A_1 \frac{d^2}{dx^2} g(x) + A_2 \frac{d^2}{dx^2} h(x) + k_1^2 g(x) = 0$$
$$A_3 \frac{d^2}{dx^2} g(x) + A_4 \frac{d^2}{dx^2} h(x) + k_2^2 h(x) = 0 \tag{10}$$

which can be simplified as

$$[A_2 A_3 - A_1 A_4] \frac{d^4}{dx^4} h(x) - [A_1 k_2^2 + A_4 k_1^2] \frac{d^2}{dx^2} h(x) - k_1^2 k_2^2 h(x) = 0$$
$$[A_2 A_3 - A_1 A_4] \frac{d^4}{dx^4} g(x) - [A_1 k_2^2 + A_4 k_1^2] \frac{d^2}{dx^2} g(x) - k_1^2 k_2^2 g(x) = 0 \tag{11}$$

The special Eigenvalues from (11) are imaginaries, as it is easier shown next

$$C_1 \frac{d^4}{dx^4} h(x) + C_2 \frac{d^2}{dx^2} h(x) + C_3 h(x) = 0$$
$$C_1 \frac{d^4}{dx^4} g(x) + C_2 \frac{d^2}{dx^2} g(x) + C_3 g(x) = 0$$

where

$$\alpha_{1,2} = \frac{-C_2 \pm \sqrt{C_2^{\,2} - 4C_1C_3}}{2C_1} \quad \Rightarrow \quad \lambda_{x1,x2} = \pm j\sqrt{\alpha_1}, \quad \lambda_{x3,x4} = \pm j\sqrt{\alpha_2} \tag{12}$$

Then the especial solution is given by

$$g(x) = c_3 \cos(\sqrt{\alpha_1} x) + c_4 \sin(\sqrt{\alpha_1} x) + c_5 \cos(\sqrt{\alpha_2} x) + c_6 \sin(\sqrt{\alpha_2} x)$$
$$h(x) = c_7 \cos(\sqrt{\alpha_1} x) + c_8 \sin(\sqrt{\alpha_1} x) + c_9 \cos(\sqrt{\alpha_2} x) + c_{10} \sin(\sqrt{\alpha_2} x) \tag{13}$$

B. Rijke boudary conditions

The next step to get the analytical solution is to calculate the constants in terms of the boundary conditions, so if it is considered the ambient temperature and air density as the reference for the first boundary condition of the Rijke tube shown in Fig. 1, then the constants for the dual dominium solution functions given by

$$T(t,x) = b_1 e^{-k_1^2 t} \cos(\sqrt{\alpha_1} x) + b_2 e^{-k_1^2 t} \sin(\sqrt{\alpha_1} x) + b_3 e^{-k_1^2 t} \cos(\sqrt{\alpha_2} x) + b_4 e^{-k_1^2 t} \sin(\sqrt{\alpha_2} x)$$
$$\rho(t,x) = b_5 e^{-k_2^2 t} \cos(\sqrt{\alpha_1} x) + b_6 e^{-k_2^2 t} \sin(\sqrt{\alpha_1} x) + b_7 e^{-k_2^2 t} \cos(\sqrt{\alpha_2} x) + b_8 e^{-k_2^2 t} \sin(\sqrt{\alpha_2} x) \tag{14}$$

result on

$$T(0,0) = 0 = b_1 + b_3 \qquad b_3 = -b_1$$
$$\rho(0,0) = 0 = b_5 + b_7 \qquad b_7 = -b_5 \tag{15}$$

The Rijke tube setup consist just in an open tube vertically oriented, in this case the longitude of this tube is L and the thermal energy is punctual supplied along the axial direction at $x=L/5$ (see Fig. 1).

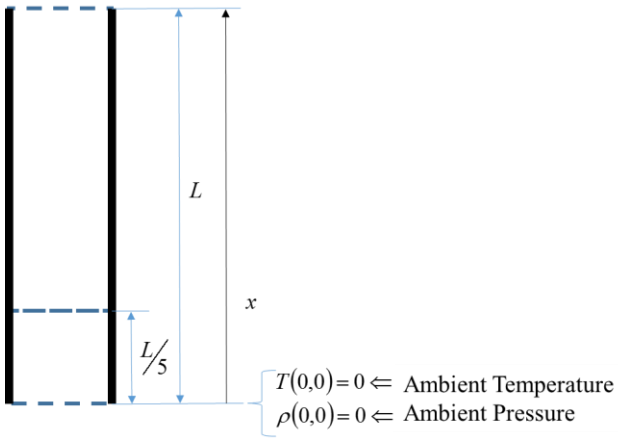


Figure 1. First boundary condition of the Rijke tube.

The second set of coefficients are calculated by considering also the second boundary condition of the Rijke tube (see Fig. 2), as

$$b_1 = -\frac{\left(b_2 \sin(\sqrt{\alpha_1} L) + b_4 \sin(\sqrt{\alpha_2} L)\right)}{\cos(\sqrt{\alpha_1} L) - \cos(\sqrt{\alpha_2} L)} = \frac{b_2 \sin(\sqrt{\alpha_1} L) + b_4 \sin(\sqrt{\alpha_2} L)}{\cos(\sqrt{\alpha_2} L) - \cos(\sqrt{\alpha_1} L)}$$
$$b_5 = -\frac{\left(b_6 \sin(\sqrt{\alpha_1} L) + b_8 \sin(\sqrt{\alpha_2} L)\right)}{\cos(\sqrt{\alpha_1} L) - \cos(\sqrt{\alpha_2} L)} = \frac{b_6 \sin(\sqrt{\alpha_1} L) + b_8 \sin(\sqrt{\alpha_2} L)}{\cos(\sqrt{\alpha_2} L) - \cos(\sqrt{\alpha_1} L)} \tag{16}$$

Here the entire solution can be simplified as

$$T(t,x) = e^{-k_1^2 t} \left[\frac{b_2 \sin(\sqrt{\alpha_1} L) + b_4 \sin(\sqrt{\alpha_2} L)}{\cos(\sqrt{\alpha_2} L) - \cos(\sqrt{\alpha_1} L)} \left(\cos(\sqrt{\alpha_1} x) - \cos(\sqrt{\alpha_2} x)\right) + b_2 \sin(\sqrt{\alpha_1} x) + b_4 \sin(\sqrt{\alpha_2} x) \right]$$
$$\rho(t,x) = e^{-k_2^2 t} \left[\frac{b_6 \sin(\sqrt{\alpha_1} L) + b_8 \sin(\sqrt{\alpha_2} L)}{\cos(\sqrt{\alpha_2} L) - \cos(\sqrt{\alpha_1} L)} \left(\cos(\sqrt{\alpha_1} x) - \cos(\sqrt{\alpha_2} x)\right) + b_6 \sin(\sqrt{\alpha_1} x) + b_8 \sin(\sqrt{\alpha_2} x) \right],$$

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and their spatial derivation are given by

$$\frac{\partial T(t,x)}{\partial x} = e^{-k_1^2 t}\left[\frac{b_2\sin\left(\sqrt{\alpha_1}L\right)+b_4\sin\left(\sqrt{\alpha_2}L\right)}{\cos\left(\sqrt{\alpha_2}L\right)-\cos\left(\sqrt{\alpha_1}L\right)}\left(\sqrt{\alpha_2}\sin\left(\sqrt{\alpha_2}x\right)\right.\right.$$
$$\left.\left.-\sqrt{\alpha_1}\sin\left(\sqrt{\alpha_1}x\right)\right)+\sqrt{\alpha_1}b_2\cos\left(\sqrt{\alpha_1}x\right)+\sqrt{\alpha_2}b_4\cos\left(\sqrt{\alpha_2}x\right)\right]$$

$$\frac{\partial \rho(t,x)}{\partial x} = e^{-k_2^2 t}\left[\frac{b_6\sin\left(\sqrt{\alpha_1}L\right)+b_8\sin\left(\sqrt{\alpha_2}L\right)}{\cos\left(\sqrt{\alpha_2}L\right)-\cos\left(\sqrt{\alpha_1}L\right)}\left(\sqrt{\alpha_2}\sin\left(\sqrt{\alpha_2}x\right)\right.\right.$$
$$\left.\left.-\sqrt{\alpha_1}\sin\left(\sqrt{\alpha_1}x\right)\right)+\sqrt{\alpha_1}b_6\cos\left(\sqrt{\alpha_1}x\right)+\sqrt{\alpha_2}b_8\cos\left(\sqrt{\alpha_2}x\right)\right] \quad (17)$$

The third boundary condition of the Rijke tube is located at point in which the thermal energy is supplied. Therefore, the temperature has a maximum point there while the air density is minimal. These boundary conditions are illustrated in Fig. 3. With these considerations the third set of coefficient are estimated as:

$$b_2 = -Db_4$$
$$b_6 = -Db_8 \quad (18)$$

where

$$D = \frac{\left(\sin\left(\sqrt{\alpha_2}L\right)\left(\sqrt{\alpha_2}\sin\left(\sqrt{\alpha_2}\frac{L}{5}\right)-\sqrt{\alpha_1}\sin\left(\sqrt{\alpha_1}\frac{L}{5}\right)\right)+\left(\cos\left(\sqrt{\alpha_2}L\right)-\cos\left(\sqrt{\alpha_1}L\right)\right)\sqrt{\alpha_2}\cos\left(\sqrt{\alpha_2}\frac{L}{5}\right)\right)}{\left(\sin\left(\sqrt{\alpha_1}L\right)\left(\sqrt{\alpha_2}\sin\left(\sqrt{\alpha_2}\frac{L}{5}\right)-\sqrt{\alpha_1}\sin\left(\sqrt{\alpha_1}\frac{L}{5}\right)\right)+\left(\cos\left(\sqrt{\alpha_2}L\right)-\cos\left(\sqrt{\alpha_1}L\right)\right)\sqrt{\alpha_1}\cos\left(\sqrt{\alpha_1}\frac{L}{5}\right)\right)}$$

Given the final model version as:

$$T(t,x) = b_4 e^{-k_1^2 t}\left[\frac{\sin\left(\sqrt{\alpha_2}L\right)-D\sin\left(\sqrt{\alpha_1}L\right)}{\cos\left(\sqrt{\alpha_2}L\right)-\cos\left(\sqrt{\alpha_1}L\right)}\left(\cos\left(\sqrt{\alpha_1}x\right)\right.\right.$$
$$\left.\left.-\cos\left(\sqrt{\alpha_2}x\right)\right)+\sin\left(\sqrt{\alpha_2}x\right)-D\sin\left(\sqrt{\alpha_1}x\right)\right] = b_4 e^{-k_1^2 t} f_x(x)$$

$$\rho(t,x) = b_8 e^{-k_2^2 t}\left[\frac{\sin\left(\sqrt{\alpha_2}L\right)-D\sin\left(\sqrt{\alpha_1}L\right)}{\cos\left(\sqrt{\alpha_2}L\right)-\cos\left(\sqrt{\alpha_1}L\right)}\left(\cos\left(\sqrt{\alpha_1}x\right)\right.\right.$$
$$\left.\left.-\cos\left(\sqrt{\alpha_2}x\right)\right)+\sin\left(\sqrt{\alpha_2}x\right)-D\sin\left(\sqrt{\alpha_1}x\right)\right] = b_8 e^{-k_2^2 t} f_x(x) \quad (19)$$

Both variables distributions along the axial longitude are the same; but, as well as it was discussed before, the temperature $T_L(t)$ at $x=L/5$ is maxima, and through convection with a controlled energy source $Q_L(t)$ coefficient b_4 can be written as:

$$b_4 = \frac{1}{f_x\left(\frac{L}{5}\right)}\left(T_L(t)-\frac{a_1}{k_1^2}Q_L(t)\right)e^{k_1^2 t} \quad (20)$$

In this point the temperature behavior can be written in terms just of b_4 as

$$T(t,x) = \frac{1}{f_x\left(\frac{L}{5}\right)}\left(T_L(t)-\frac{a_1}{k_1^2}Q_L(t)\right)\left[\frac{\sin\left(\sqrt{\alpha_2}L\right)-D\sin\left(\sqrt{\alpha_1}L\right)}{\cos\left(\sqrt{\alpha_2}L\right)-\cos\left(\sqrt{\alpha_1}L\right)}\left(\cos\left(\sqrt{\alpha_1}x\right)\right.\right.$$
$$\left.\left.-\cos\left(\sqrt{\alpha_2}x\right)\right)+\sin\left(\sqrt{\alpha_2}x\right)-D\sin\left(\sqrt{\alpha_1}x\right)\right] \quad (21)$$

By taking advantage of the equivalent axial distribution of both variables of interest, the b_8 coefficient can be considered as:

$$b_8 = b_4\frac{\rho(t,x)e^{k_2^2 t}}{T(t,x)e^{k_1^2 t}} = b_4\frac{\rho\left(t,\frac{L}{5}\right)}{T\left(t,\frac{L}{5}\right)}\frac{e^{k_2^2 t}}{e^{k_1^2 t}} \quad (22)$$

Finally, the dynamics of the spatial behavior of both estimation variables can be written in terms of the boundary conditions and the energy supplied to the Rijke tube.

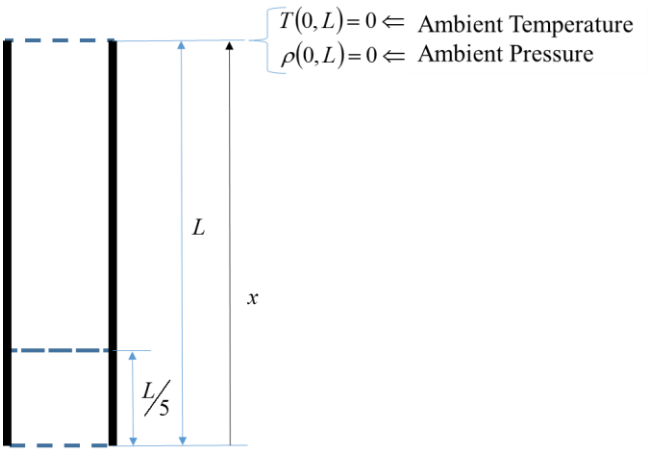


Figure 2. Second boundary condition of the Rijke tube.

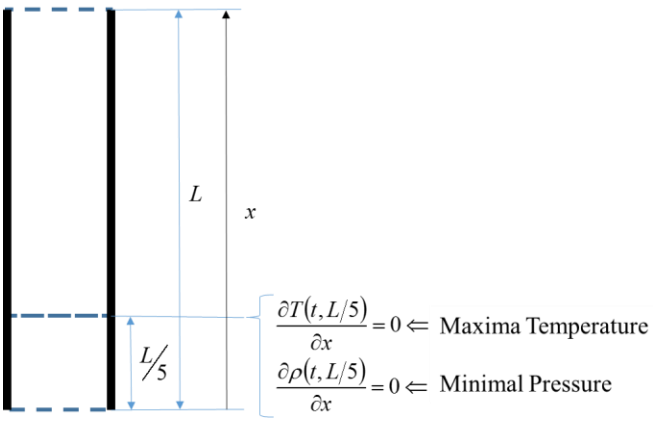


Figure 3. Thrid boundary condition of the Rijke tube.

Here we can say that because of the constant volume, the air density and pressure variables have the same dynamical and spatial behavior.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

III. ANALYTICAL MODEL FOR PRESSURE OSCILLATION FROM THE WAVE EQUATION

Thermoacoustic phenomenon is defined as the direct transformation from thermal energy into acoustic energy. Three conditions are required for this transformation [1]:

1. The medium must be a compressible fluid,
2. A temperature gradient must exist, and
3. The control volume must be contained by a physical border (chamber housing) [28].

The wave equation is a linear second-order differential equation for both the time as spatial domains, which describes the propagation of oscillations in a continuous media. The goal of this section is to establish a wave equation relation for the pressure variable in an air column when the thermoacoustic phenomena appears into a Rijke tube.

A. Stationary wave in an air column

The wave equation for acoustic pressure is given by [29].

$$\frac{\partial^2 \xi}{\partial t^2} = \frac{\kappa}{\rho_0}\frac{\partial^2 \xi}{\partial x^2} \tag{23}$$

with ξ the displacement in the air, κ the bulk modulus for the air and ρ_0 the air density under equilibrium conditions.

As well as it was performed for the Luikov equations, the wave equation solution for the acoustic was solved via the variables separation strategy.

The concept of a standing wave is based on the spatial superposition of two waves with the same time and space frequencies, which travel in opposite directions in the medium with the same amplitude. For this Rijke tube experiment, the standing wave resulting from the wave equation for the air behavior, which is given as

$$\xi(x,t) = 2\xi_0 \sin(kx)\cos(\omega t) \tag{24}$$

where k index does correspond to the wavenumber, ξ_0 is the wave amplitude and ω its angular frequency. The Taylor law for fluid gradient [30] was used to formulate the pressure of standing waves in an air column, where

$$p - p_o = \kappa\frac{\partial \xi}{\partial x}; \tag{25}$$

therefore, the pressure behavior can be written as

$$P(x,t) = -2\xi_0 \kappa k \cos(kx)\cos(\omega t) \tag{26}$$

Equation (26) describes the distribution of pressures with no boundary conditions. Then again, as well as it occurs for the Luikov equation solutions, the boundary conditions for the air wave describes the Eigenvectors, which for this case are analyzed as nodes (see Fig. 4).

B. Node analysis (modal shapes or Eigenvectors)

A tube open at both ends provides a physical border for the concerned air column contained. Reflection causes nodes and antinodes of pressure and air displacement, as well as it is illustrated by Fig. 4.

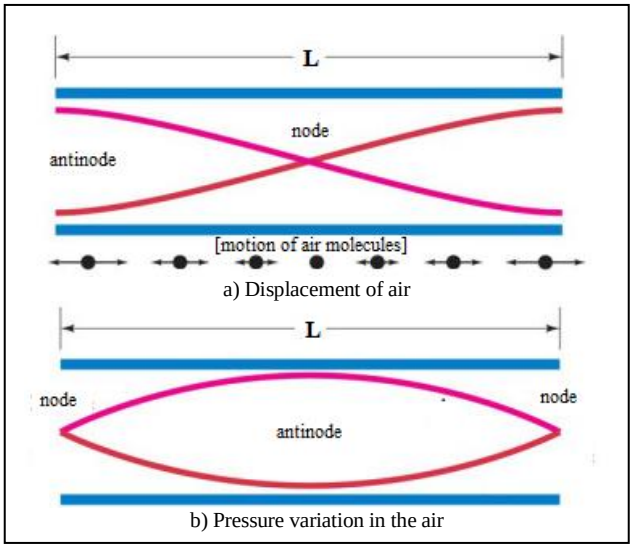


Figure 4. Pressure and displacement for the air in the columns.

Air displacement nodes are associated with the motion of its molecules, which produce in this case the modal nodes. Therefore, at both ends of the column into the Rijke tube boundaries must be a pressure node because the atmosphere cannot present a significant pressure change induced by this phenomena [31]. Then the relationship between pressure and motion of the air can be written as (24).

The generalized one-dimensional equation that describes the pressure distribution of an air contained in a cylindrical combustion chamber is obtained by considering the boundary conditions illustrated on Figs. 1 and 2. Then from (24), the pressure distribution model can have been formulated:

$$P(x,t) = -2\kappa k\xi \cos(k(x + \pi/2k))\cos(\omega t) \tag{27}$$

from the boundary conditions evaluation, the obtained model coefficients are:

Condition 1:

$$\begin{aligned} P((0),t) &= -2\kappa k\xi \cos(k((0) + \pi/2k))\cos(\omega t) \\ 0 &= \cos(\pi/2)\cos(\omega t) \\ 0 &= 0 \end{aligned}$$

Condition 2:

$$\begin{aligned} P((L),t) &= -2\kappa k\xi \cos(k((\pi/k) + \pi/2k))\cos(\omega t) \\ 0 &= \cos(3\pi/2)\cos(\omega t) \\ 0 &= 0 \end{aligned}$$

So the one-dimension pressure distribution model on the can combustor results [1].

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

$$P(x,t) = -2kc^2 \rho\xi \cos(k((\pi/k) + \pi/2k)) \cos(\omega t) \quad (28)$$

IV. RIJKE TUBE EXPERIMENTE

With the intention of confirming, thermoacoustic is concerned with the interactions between heat (thermos) and pressure oscillations in gases (acoustics). Then a "Rinke Tube", named after its inventor, is a fundamental tool for studying thermoacoustic phenomenon. Rijke tube turns heat into sound, by creating a self-amplifying standing wave. This open cylinder resonator contains a metallic copper mesh positioned about one-fifth of the total tube length. When the mesh is heated by a burner flame and then when the flame is take out, the concerned mesh will produce a strong sound pitch at its resonant frequency for several seconds.

A Rijke tube experiment was performed considering the experimental setup shown in Fig. 5 [32]. This Rijke tube consist on: a steel pipe of 0.5 m long and a 0.04445 m (1 ¾ in) diameter with a thickness of 0.00121 m, is selected and instrumented, as seen in Fig. 5. Let x_m be the distance (0.12m) where metallic mesh was placed.

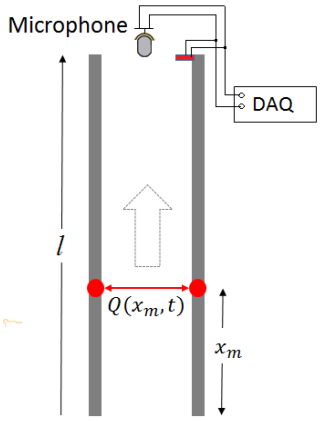


Figure 5. Data adquisition for Rijke tube experiment .

Rijke tube turns heat into sound, by creating a self-amplifying standing wave just after the heat source provided by a gas burner was taken out. Gas burner add energy to the system until to metallic mesh rise a 600 °C of temperature; then, the heat source is moved aside and thermoacoustic phenomenon appears. It is important to consider that the position of pipe must be maintained vertical.

A. *Fourier analysis*

Time domain analysis is the interpretation of physical signals evolution with respect to time. It is beneficial when observing data such as temperature. However, because of the time response some applications require analyzing the frequency components of signals, such as pressure oscillation of the air.

The acoustic pressure was measured using a microphone during Rijke tube experiment [33]. Time domain acoustic signal does not provide significant information for analysis. Hence, Fourier analysis is required. A fast Fourier transform (FFT) is an algorithm that samples a signal in time domain (or space domain) and turns it into frequency domain, exhibiting its frequency components. FFT helps to find the fundamental frequency produced by the air column resonance.

The experiment is executed taking 44100 samples per second using the microphone. The Fast Fourier Transform was applied with a resolution of 15 bits. Frequency resolution is 1.34 Hz, i.e., every 743 ms the algorithm takes a package sample to analyze.

The signal for amplitude in time domain and frequency domain obtained with the microphone is shown in Fig. 6. Here the first modal mode presence is identified at 301 Hz with an amplitude of 0.1941 Pa.

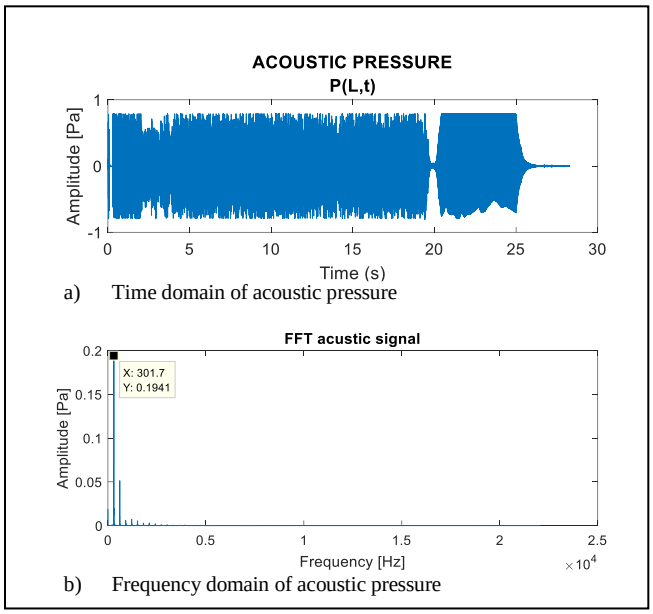


Figure 6.Signal analysis of acoustic pressure

B. *Sound speed in the air*

The speed of sound is affected by changes in the medium (density, humidity, temperature, etc.). For our study case, the element that changed significantly is the temperature.

The geometry of the combustor determines frequency and wavelength of the oscillating pressure produced by the thermoacoustic effect. The symbol λ is the label for length of the wave. Fig. 7 illustrates a reflection wave on the pipe open ends. The output wave (green) and the reflected wave (red) only perform half a cycle each. The tube length is half the Wavelength.

Considering idealistic conditions, the fundamental frequency is inversely proportional to the length of the tube.

$$c/2\pi\acute{\imath}=f \quad (29)$$

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

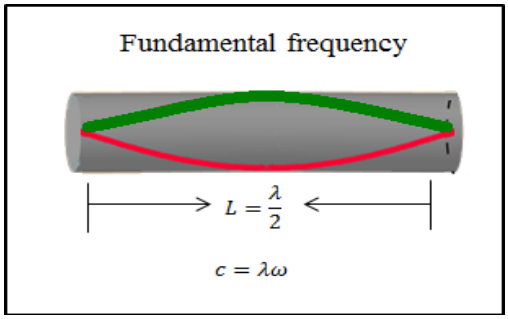


Figure 7. First vibration mode in air column.

In an ideal tube, the wavelength of the sound produced is directly proportional to the length of the tube. A tube with aperture at one end and closed at the other one produces sound with a wavelength equal to four times the length of the tube. In acoustics, end correction factor is a short distance added to the actual length of a resonance pipe, to calculate the precise resonance frequency of the pipe. The tone of a real tube is lower than the predicted theoretically. The end correction for a pipe is given as

$$\lambda = 2(L + 0.61D) \tag{30}$$

At the end, the speed of sound resulted is 379.18 m/s.

C. Temperature of the air

Temperature is a physical quantity produced by motion of the molecules, giving a perception of hot and cold. Temperature is associated with friction. Molecular friction is associated with pressure and density in the air. The ideal gas law is a good approximation of the behavior of many gases under several conditions, although it has limitations. The most frequently form of a state equation is given by

$$\frac{P}{\rho} = \frac{RT}{M} \tag{31}$$

The acoustic wave velocity c depends of the material properties. Speed of sound is proportional to bulk modulus and inversely proportional to density of the air as dictate in (23), resulting in this case as

$$\frac{\kappa}{\rho} = c^2$$

Where $\kappa = \gamma P$ and γ is the heat capacity ratio. There is a strong relationship between pressure of the air, density of the air, temperature of the air and speed of sound [33]. This relationship is

$$\frac{\gamma P}{\rho} = c^2 = \frac{\gamma RT}{M} \tag{32}$$

Equation (32) implies that temperature can be expressed in terms of speed of sound.

$$T_{air} = \left(\frac{c_{air}}{20.055 \frac{m}{s \cdot K^{1/2}}} \right)^2 \tag{33}$$

Equation (33) is defined using nominal values of air. The density of a substance is the relationship between mass amount and volume occupied. The density can change by changing either the pressure or the temperature. Increasing the temperature generally decreases the density.

Table I shows the density of the air in terms of air temperature.

TABLE I. TEMPERATURE AND DENSITY OF THE AIR

Speed of sound (m/s)	T (K)	T(°C)	ρ [Kg/m^3]
379.18	357.48	84.48	0.9870

V. RESULTS

The one-dimension pressure distribution model on the concerned can-combustor is formulated on Section III, along with the determination of parameters' values.

$$P(x,t) = -2kc^2 \rho\xi \cos(k((\pi/k) + \pi/2k)) \cos(\omega t)$$

Pressure distribution along the can-combustor is described in Fig. 8. This paper becomes useful for several combustor chambers with similar geometries.

Advanced FFT Spectrum Analyzer is used to verify in real-time the signal evolution, as depicted in Fig. 9. Due to expansion and compression ratio, the air will generate high pressure areas whose magnitude is directly proportional to the temperature of the medium. The temperature of the metal gauze oscillates around 600 °C at t=0, convectively transferring the thermal energy and change the properties of the medium. The average temperature of the column was 78 ° C.

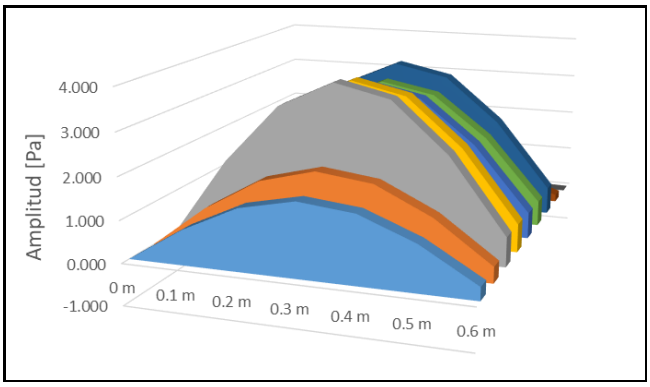


Figure 8. Fundamental mode of vibration in air column.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

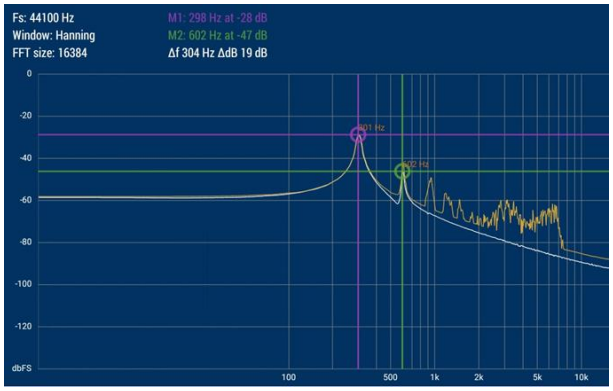


Figure 9. Fundamental frecuency measure on Rijke tube experiment.

It is experimentally proven with the results that the speed of sound has been altered by the temperature change in the medium that propagates the disturbance, which was 379 m/s. Following the relationship on (23), the fundamental frequency of the gaseous fluid contained in a cylindrical geometry is obtained. Thermoacoustic phenomenon presence is used to validate the pressure distribution model for the air oscillation inside the Rijke tube.

VI. CONCLUSION

Taking as reference theoretical results from equations (19) and (28) both predicts a modal node just at the middle of the Rijke tube axial dimension for the air density and pressure, respectively. Also, (19) predicts an antinode for the temperature behavior.

Experimentally, the air density and pressure modal node have been proven by the results reported in Fig. 8; where the thermodynamics relationship between the air flow speed (sound) and the internal temperature measured supports these results from the Eigenvalues gotten in Fig. 9.

So that, at the end, the main conclusion is that the propagation of acoustic oscillation in a compressive medium (air in this case) may be induced by applying thermal energy to the chamber housing surface, so the thermodynamic relationship between pressure, density and temperature has been verified for a single geometry combustor chamber, a Rijke tube in this case.

The pressure distribution model for the dynamic behavior of the pressure oscillation into a Rijke tube synthesized from the wave and Luikov equations, was validated by predicting the first mode frequency of the air column pressure inside, as well as it is shown in Fig. 10, where also the Fourier Analysis results are included.

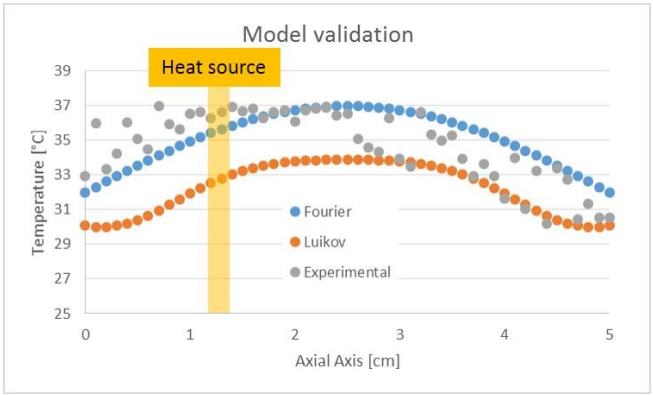


Figure 10. Experimental Model validation.

Experimental results illustrated in Fig. 10 were obtained by using a thermos-camera and a set of thermocouples at the Rijke tube external surface. Both models in the temperature space shows an acceptable prediction, but analysis needs to explain why boundary conditions has change in their average values.

Based on the temperature results, the Luikov equation for pressure can be used with certain reliability to have a model for both variables.

The next step is to use both models to predict the concerned variables behavior but by been taking into account the experimental measurement of their boundary values.

For the National Laboratory of Cooling Technology Research, the results reported here are important in terms of the thermoacoustic technology development to provide an alternative for the cooling systems; because from these fundamentals both models are going to be used to develop control strategies for the cooling generation and ambient conditioning.

For the P02 project from the CEMIE Eolic program, these results take importance because of the presence of the thermoacoustic phenomena in several additive manufacturing process, which disturbs the manufacturing finishing and composition.

ACKNOWLEDGMENT

Authors thanks National Council for Science and Technology (CONACYT) for the scholarships of the postgraduate students involved; also because of the support through the consolidation project for the National Laboratory Research on Cooling Technologies (LaNITeF) Budget No. 293784. These results were gotten by the synergy of both projects the LaNITeF consolidation as well as the P02 of the Mexican Center for Innovation in Eolic Energy (CEMIE Eólico) supported also by the Secretariat of Energy, through the Budget for Sustainable Energy.

REFERENCES

[1] I. Mejia, E.E. Rodriguez, C. Núñez and C. Cruz, “Thermoacoustics Analysis in a Rijke Tube,” The Thirteenth International Conference on Systems (ICONS 2018) IARIA, Apr. 2018, ISBN: 978-1-61208-626-2, ISSN: 2308-4243.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[2] M. Brauer, "Ambient Air polution Expousre Estimation for the Global Burden of Disease 2013," Enrimental Science & Technoloy, No. 50, pp. 79-88, 2016.

[3] GBD 2013 Risk Factors Collaborators, "Global, Regional, and Natural Comparative Risk Assessment of 79 Behavioral, Enviromental and Occupational, and Metabolic Risk or Clusters of Risks in 188 Countries, 1990 – 2013: A Systematic Analysis for the Global Burden of Disease Study 2013," The Lancet 394, pp. 2287 – 2323, 2016.

[4] A. Valera-Medina, S. Morris, J. Runyon, D.G. Pugh, R. Marsh, P. Beasley, T. Hughes, "Ammonia, Methane and Hydrogen for Gas Turbines," Energy Procedia, vol. 75, pp. 118 – 123, 2015.

[5] Y. Okuri, Heat pump type air conditioner, US patent, US5937669A, 1998.

[6] L. E. Bell, "Cooling, Heating, Generating Power, and Recovering Wasted Heat with Thermoelectric Systems," Science, vol. 321, Issue 5895, pp. 1457 – 1461, 2008.

[7] G. W. Swift, "Thermoacoustics; A Unifying Perspective for Some Engines and Refrigerators," Ed. Springer, ISBN: 978-3-319-66933-5, 2017.

[8] B. Schuermans, F. Güthe, D. Pennell, C. O. Paschereit, "Thermoacoustic Modelin of a Gas Turbine Using Transfer Function Measured Unde Full Engine Pressure," Journal of Engineering for Gas Turbine and Power, vol. 132, No. 11, 2010.

[9] J.W.S. B. Rayleigh, "The theory of sound," Book, vol. 2, London: Macmillan and Co. Cambridge 1877.

[10] P. Merkli and H Thomann, "Termoacoustic effects in a resonance tube," Journal of Fluid Mechanics, vol. 70, Cambridge University. July 1975.

[11] T. C. Lieuwen and V. Yang, "Combustion Instabilities in Gas Turbine Engines: Operational Experience, Fundamental Mechanisms and Modeling," American Institute of Aeronautics and Astronautics, 2006.

[12] M. O. Vigueras-Zuñiga, A. Valera-Medina, N. Syred, and P. Bowen, "High Momentum Flow Region and Central Recirculation Zone Interaction in Swirling Flows," Sociedad Mexicana de ingeriera Mecanica. vol. 4, pp. 195-204, 2014.

[13] S. Hochgreb, D. Dennis, and I. Ayranci, "Forced and Self-Excited Instabilities From Lean Premixed, Liquid-Fuelled Aeroengine Injectors at High Pressures and Temperatures" Turbine Technical Conference and Exposition, ASME Turbo Expo 2014.

[14] N. Yadav and A. Kushari. "Effect of swirl on the turbulent behaviour of a dump combustor flow", Journal of Aerospace Engineering, vol. 224, no. 6, pp. 705-717, 2010.

[15] A. Valera, A. Griffiths, and N. Syred, "Analysis of the Impact Caused by Coherent Structures in Swirling Flow Combustion Systems", Ingeniería investigación y Tecnología, FI-UNAM, 2012.

[16] Z. Zhang, D. Zhao, N. Han, S. Wang, J. Li, "Control of combustion instability with a tunable Helmholtz resonator", Aerospace Science and Technology, Elsevier, 2014.

[17] W. Rowen. "Simplified mathematical representations of heavy-duty gas turbines", ASME J. Eng. Power, vol. 105, pp. 865-869, 2013.

[18] J. Rai, N. Hasan, B. Arora, R. Garai, R. Kapoor and Ibraheem. "Performance Analysis of CCGT Power Plant using MATLAB/Simulink Based Simulation", International Journal of Advancements in Research & Technology, vol. 2, no. 5, 2013.

[19] R. Smith, J. Ranasinge, C. Gulen, Combined Cycle Power Plant, US Patent, US20070017207A1, 2005.

[20] T. Samad and A. Annaswamy. "The impact of Control Technology", [Online] Available: ww.ieeecss.org, 2011

[21] A. Banaszuk, "Control of Combustion Inestability: From Passive to Active Control of Combustors", Grand Challenges for Control, IEEE, 2012.

[22] A. V. Luikov, Heat and mass transfer in capillary porous bodies, Pergamon Press, Oxford, U.K., 1966.

[23] P. Contantin, C. Fioas, Navier – Stokes Equations, The University of Chicago Press, ISBN 0-226-11548-8, 1989.

[24] D. De-Vries, Simultaneous transfer of heat and moisture in porous media, Transaction on Am. Geophy, Unio 39, pp. 909-016, 1968.

[25] T.Z. Harmathy, Simultaneous moisture and heat transfer in porous systems with particular reference of drying, I. Ec. Fundamentals, vol. 8, no. 1, pp. 92 – 103, 1969.

[26] K.N. Shukla, Diffusion Process During Dryling of Solid, World Scientific Publishing Co., Singapore, 1993.

[27] C. Dong, K. Jay, and J. Yong. "Analysis of the combustion instability of a model gas turbine combustor by the transfer matrix method", Journal of Mechanical Science and Technology, vol. 23, April, pp. 1602-1612, 2009.

[28] S. Kai-Uwe, K. Rainer and B. Hans-Jörg. "Experimental Characterization of Premixed Flame Instabilities of a Model Gas Turbine Burner". Flow, Turbulence and Combustion, vol. 76, no. 2, pp. 177-197, 2006.

[29] M. Alonso and E. Finn, "Physical: Waves and Fields" Book, vol. II. Fondo Educativo Interamericano, pp 694-712.

[30] Universidad de Valladolid, "Waves and propagation", Curso online:https://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io2/public_html/viento/principios_viento.html. April 2017.

[31] S. Shariati, A. Franca, B. Oezer, R. Noske, D. Abel, A. Brockhinke, "Modeling and Model Predictive Control of Combustion Instabilities in a Multi-section Combustion Chamber Using Two-Port Elements", IEEE Multi-conference on Systems and Control, Antibes, France. 2014

[32] Y. Cengel and M. Boles, "Termodinamica" Book, 7th Edition, Mc Graw Hill, 2012.

[33] J. Mathews and F. Kurtis. "Numerical Methods Using Matlab". Fourth Edition. Prentice Hall New Jersey. Chapter 2, 2004.

[34] E. Gutiérrez, G. Vélez, D. Szwedowicz, J. Bedolla, C. Cortés. "Identification of Close Vibration Modes of a Quasi-Axisymmetric Structure: Complemetary Study," Ingeniería Investigación y Tenología, vol XIV, Apr, pp. 207-222, 2012.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Experimental and Fundamental Analysis of the Dynamical Behavior for the Thermal Distribution in a Cooling Chamber

Pérez Vázquez Jesús Hernán
Engineering Center for Industrial Development (CIDESI)
National Research Laboratory on Cooling Technology (LaNITeF)
Santiago de Querétaro, México
jperez@posgrado.cidesi.edu.mx

Montoya Santiyanes Luis Alvaro
Engineering Center for Industrial Development (CIDESI)
National Research Laboratory on Cooling Technology (LaNITeF)
Santiago de Querétaro, México
lmontoya@cidesi.edu.mx

Nuñez Martín Carlos Alexander
Engineering Center for Industrial Development (CIDESI)
National Research Laboratory on Cooling Technology (LaNITeF)
Santiago de Querétaro, México
cnunez@posgrado.cidesi.edu.mx

Rodriguez Vázquez Eloy Edmundo
Engineering Center for Industrial Development (CIDESI)
National Research Laboratory on Cooling Technology (LaNITeF)
Santiago de Querétaro, México
eloy.rodriguez@cidesi.edu.mx

Mejia Alonso Israel
Engineering Center for Industrial Development (CIDESI)
National Research Laboratory on Cooling Technology (LaNITeF)
Santiago de Querétaro, México
imejia@cidesi.edu.mx

Dominguez Librado Guillermo
Engineering Center for Industrial Development (CIDESI)
National Research Laboratory on Cooling Technology (LaNITeF)
Santiago de Querétaro, México
gdominguez@posgrado.cidesi.edu.mx

***Abstract*— A mathematical model is proposed to describe the dynamics of the heat extraction process in a domestic refrigerator. The concerned model is based on both the Newton's Law of Cooling and the Fourier Heat Equation; where the first one acts as the boundary condition to consider the physical properties of the insulation walls and their interaction with the environment and with the internal energy contained in the cooler chamber. The Fourier Equation has been implemented to describe the energetic behavior in the concerned chamber and it is grounded in the numerical consideration of the finite prismatic volumes concept, which take the respective thermodynamic properties of air or water as well as the thermal load distribution defines. The finite volume concept and the conductive equation consider the null mass exchange between volumes; therefore, heat conduction emulates the finite volumes interaction and convection simulates the cooling chamber boundaries. Codification was performed through Matlab and its experimental validation took places in the National Laboratory for Cooling Technology Research (LaNITeF). Certain finite volumes were located strategically into the cooling chamber to be compared with the sensor's measurements. The model provides useful data to improve the understanding of the temperature behavior in terms of the chamber geometry, cool air flow and thermal load ratio, leading to basics concepts for the future development of control strategies to implement several energetic consumption optimization algorithms.**

Keywords-Heat exchange;thermal diffusivity;finite volume.

I. INTRODUCTION

Nowadays, the implementation of new technologies in the refrigeration industry has increased significantly. The objective of these technological developments has different arguments, depending on the problem to solve, but in the global context, the most important issues are: the energy consumption of the vapor compression cooling systems, the heat extraction from of high magnitudes of thermal loads, the impact of the refrigerant gas leaks to the environment and the performance of the heat interchangers of the vapor compression thermodynamic cycle [1].

There are interesting contributions in the state of the art concerned with theoretical and/or experimental methodologies to extract the thermodynamic parameters of cooling devices or systems from several companies, moreover, from manufacturers of thermoelectric chillers (TEC) and thermoelectric generators (TEG).

The Lineykin et al. model could be useful to analyze the displacement requirements of the TEC and the loading effects of the TEG [2]. The cited review and analysis indicate that greenhouse emissions from conventional vapor compression refrigeration systems driven by diesel engines commonly used for the refrigeration of food transportation can reach 40% of the vehicle's engine emissions [3]

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Thermoacoustic refrigeration is an alternative technology capable to provide cooling levels for several temperature magnitudes, without using any substance harmful to the nature environment. Therefore, thermoacoustic represents a technology niche to increase the cooling market for a bigger production of perishables products [4]. Currently, the most efficient thermoacoustic refrigerators are used in the industrial environments [5], which work by taking advantage of the interdependence between the enthalpy carried through by the Gedeon mechanism and the local temperature; as an example of this fact we have the recently developed four-stage progressive wave thermoacoustic engine [6].

The prime movers and refrigerators based on thermoacoustic have gained considerable importance toward practical applications because of: the absence of moving components, the reasonable efficiency, the use of environmentally friendly working fluids, etc. [7]. The results from the experimental studies were correlated in terms of: the Nusselt number, the Prandtl number and the Reynolds number to estimate their correlation in the heat transfer stage at the heat exchangers. Experimental data show that there are significant errors in the cited heat transfer correlation analysis algorithm. Dynamically, these results also show that there is a relationship between the oscillatory heat transfer coefficient at the heat exchangers and both the mean pressure and the frequency of oscillation [8].

The perishable products conservation has been indispensable in any place that requires a conservation process for food, as well as in: restaurants, large, medium and small commercial stores, that transport or shows different human consumption products which must remain at a minimum temperature level to be conserved in their best conditions, as well as for domestic purposes in daily life [9].

In the other hand, cooling technology is one of the most used and important services for the industry manufacturing process, more over it is critical for many specific sectors as well as the healthcare and feeding; essentially because of the infrastructure of their production and storage processes [10]. Therefore, cold needs are always present in some industries, where the material or product is obtained, cultivated or extracted, also where it is transferred to the plant, factory or warehouse, even though in its storage time, and in its production and conservation processes as well as when it is put on sale in optimal conditions for the final consumer [11] and [12].

This article analyzes the energy exchange mechanism between the thermal load and tend the volume cooled by the evaporator. This analysis scope is extended to the behavior of the temperature distribution in the cooling chamber of a domestic refrigerator, by modifying both the density of the elements (finite volumes) according to their position into the cooling volume and the ratio between its energy consumption and the total thermal load (considering 3 percentages of full load or volume) [13]; with the purpose to focus on the cooling power to get a better conservation performance of the food characteristics for and then to improve the energetic performance of the concerned cooling systems [14].

Several deeper researches work on different areas of refrigeration technologies (i. e., condenser, energy exchange) shows that the environmental conditions change strongly affects these systems performances. In the matter of fact all cooling systems designs needs to take in account this to improve its energetic performance and to reduce its effect in the natural environment health. The improvements can also be developed to decrease the negative environmental impact of refrigerant leaks by prediction and monitoring their flow by using a mathematical model. These mathematical models have been developed by taking considering the different relationships between the thermodynamically variables of interest [15].

One of the most relevant foundation for the dynamic analysis of energy in the state of the art, was achieved and documented on [16], where there is a formulated mathematical model which use also both the Newton law for cooling and the heat transfer Fourier equation; by solving both equations using the finite volume method, without mass flow, considering just the thermodynamic characteristics of the water in different percentages of thermal load [17].

The present work is structured as follows: Section II mentions the basic concepts to detail the proposed theories and the expected effects. Section III presents the mathematical model developed and the results obtained from simulations, in addition to have an early interpretation of the process of energy exchange between mass along the internal volumes. Section IV explains the results generated by the simulations and their possible effect in the refrigeration cycle. Finally, section V presents the conclusions of this first stage of the developed research.

II. BASIC CONCEPTS

The most important assumptions considered in the mathematical model proposed here, are listed as follows:

-The environment temperature and pressure are homogenous and time invariant.

-The thermodynamic characteristics of the involved materials are also time invariant and spatially isotropic.

-There is no mass flow through the finite volume nodes into the refrigerator chamber.

-Energy interchange into the chamber is carried out by conduction between inside nodes and by convection at the boundary conditions.

As it was defined, with these assumptions the dynamic model can be synthetized as a linear relationship for the considered variables through both the Newton law for cooling and the Fourier heat transfer equation.

In the testing procedure to emulate different levels of thermal load some water containers were used and locate at the refrigerator and freezer chambers.

It is important to clarify that the Fourier heat transfer equation was implemented to solve the temperature conduction between the internal nodes conceived as finite volumes into the chamber, so that the mass flow has been considered null [18]. In addition, The Newton law for cooling solves the heat convection between the internal finite volume nodes and the boundary nodes at the isolation walls. Table I

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

provides a useful list of terms used in this work and the mathematical symbols related.

TABLE I. NOMENCLATURE

Symbol	Description	Units SI
α	Thermal diffusivity	K/ρC
K	Thermal conductivity	W/m °C
ρ	Density	Kg/m^3
C_v	Specific Heat	J/Kg °C
$\Delta_{x,y,z}$	Delta	m
Δ_t	Time Differential	s
V_{nodo}	Node volume	m^3
$\dot{Q}$	Generated Heat	W
CFC	Compressor Cooling Capacity	W
CFE	Refrigerator capacity of the evaporator	W
Ta	Room Temperature	°C
R134a	Refrigerant	Gr.
Te	Evaporator temperature	°C
$V_{subnodo}$	internal temperature constant	m^3
Ti	Constant of volumes	°C
τ_0	internal temperature constant	-
τ_0	Constant of volumes	-
τ_2	Evaporator constant	-
%CT	percentage of thermal load	% o lt.

Experimentally, two states of thermal load ratio, empty (0%) and full (100%), were tested to determine the temperature evolution and the energy behavior along the testing time [19]. The main purpose of both tests was to quantify the effect of the thermal momentum into the refrigeration cycle performance [20].

A. Characteristics of the model

The transfer medium into the confined volume of the refrigerator chamber for the empty scenario was considered as a homogeneous air mix with a thermal conductivity of 0.024 w / m ºK, where the initial condition is the environment temperature as 25 °C.

The system dynamics is altered when the thermal load ratio changes its magnitude from empty to full, then the thermodynamic properties were modified (by adding water to the transfer media). In this scenario the heat extraction behavior takes more time from its initial to its final equilibrium. It must take in account that the water as transfer media is considered due to its magnitude into the food composition.

B. Theoretical Proposal

The mathematical fundamentals of the dynamic model to analyze the relationship between the heat extracted from the cooling chambers volume and its temperature distribution, comes from the convection Newton Law [21],

$$\dot{Q}_{conducción} = -kA\frac{dT}{dx} \quad (1)$$

From the Newton law the amount of heat energy is included in the space discretized Fourier equation where it can see that some characteristics can change in the time domain according to the estimated amount of energy that the heat extractor absorbs, therefore the most significant coefficients are the thermal conductivity, the temperature gradient and the volume; but in this proposal to obtain a linear approximation, the properties are considered time invariant throughout the simulation [22].

$$T^n = \alpha \cdot \Delta T\left(\frac{\partial^2 T^n}{\partial x^2} + \frac{\partial^2 T^n}{\partial y^2} + \frac{\partial^2 T^n}{\partial z^2} + \frac{1}{k}\dot{Q}^n\right) + T^{n-1}. \quad (2)$$

C. Volume Analysis

The objective of this proposal is to describe the thermal distribution into the refrigerator chamber; therefore, all real spatial dimensions and thermal conduction properties of the refrigerator components materials were considered. Then, from time and space discretized the Fourier heat transfer equation, the formulation for the next time state of the temperature was synthetized for each i, j, k internal node, in terms the thermal properties of the materials, which occupy the confined volume causing a orthonormal heat transfer flow with its neighbors through the three spatial directions as were.

$$T^n_{i,j,k} = \alpha \cdot \Delta t\left(\frac{T^{n-1}_{i+1,j,k} - 2T^{n-1}_{i,j,k} + T^{n-1}_{i-1,j,k}}{\Delta x^2} + \frac{T^{n-1}_{i,j+1,k} - 2T^{n-1}_{i,j,k} + T^{n-1}_{i,j-1,k}}{\Delta y^2} + \frac{T^{n-1}_{i,j,k+1} - 2T^{n-1}_{i,j,k} + T^{n-1}_{i,j,k-1}}{\Delta z^2} + \frac{1}{k}\dot{Q}^n_{i,j,k}\right) + T^{n-1}_{i,j,k} \quad (3)$$

$$T^n_{i,j,k} = \tau_1\left(T^{n-1}_{i+1,j,k} - 2T^{n-1}_{i,j,k} + T^{n-1}_{i-1,j,k} + T^{n-1}_{i,j+1,k} - 2T^{n-1}_{i,j,k} + T^{n-1}_{i,j-1,k} + T^{n-1}_{i,j,k+1} - 2T^{n-1}_{i,j,k} + T^{n-1}_{i,j,k-1}\right) + \tau_2\left(\dot{Q}^n_{i,j,k}\right) + T^{n-1}_{i,j,k} \quad (4)$$

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

$$\tau_0 = \frac{\left(\frac{K}{\rho C_v}\right)(\Delta t)}{\left(\Delta_{x,y,z}\right)^2} \quad (5)$$

$$\tau_2 = \left(\frac{\alpha * \frac{\Delta t}{V_{subnodo}}}{\frac{k}{\rho C_V}} \right) = \frac{^\circ C}{W} \quad (6)$$

$$\tau_1 = \frac{\tau_0 * V_{nodo}}{V_{Subnodo}} \quad (7)$$

Spatially, some finite volume nodes location was defined in the same places that some sensors were located which were used to validate the numerical prediction. The heat source is the evaporator device, so that its heat input is negative, because it extracts the energy from the confined volume of the refrigerator chamber. The interaction of energy between the evaporator and the outside variables is not explained in this document. Fig. 1 shows the finite volume nodes distribution for the temperature model inside the refrigerator chamber.

III. MATHEMATICAL MODEL AND SIMULATION RESULTS

In this section, the final formulation of the mathematical approach developed in the present work is detailed; adding the data of the real thermal and energetic conditions which are considered in the synthetized model (4). Furthermore, the Matlab codification of (4) is compiled and executed to generate the numerical results shown on Fig. 3.

The data acquisition process is applied to a commercial fridge with a linear compressor and the following features
- 132 watts, 60 Hz, 0% of thermal load ratio.
- 313 watts, 150 Hz, 100% of thermal load ratio.

To clarify the proposal, a computational node was defined for each finite volume; as it is shown in both Fig. 2 and Fig. 4.

The width of the insulating wall (polyurethane) for the refrigerator chamber is 0.05 m, and the energy entering this chamber was quantified as 1.64 W by equation (3); this energy magnitude which is added to the heat extracted by the evaporator, depends of the environment temperature and it is consistent with other authors reports [16] and [17].

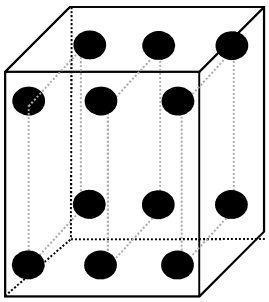


Figure 1. Spatial distribution of the finite volume nodes of the time discrete model for the temperature into the refrigerator chamber.

In the testing scenario the internal temperature of the cooling chamber volume as well as the thermal load energy start at the ambient temperature, so the despondency process is also considered for this prediction; to look for the variation magnitude in the thermal inertial moment of the entire load inside. Then, the entering energy leaks from the ambient were quantified in the same point of the initial thermal equilibrium as

$$\dot{Q} = \frac{\Delta T}{R T} = \frac{Ta - Ti}{R T} \quad (8)$$

$$CFC = (1.05)(CFE) \quad (9)$$

The value of energy that comes from the environment is considered homogeneously for each existing volume, in any of its three directions at the boundary conditions, as it is illustrated in Fig. 2. In the node net shown (Fig. 2) the nodes next to the red wall, represent the ones which utilize the convection mechanism of energy transfer with the environment while the in the remaining nodes interaction is describe by conduction just.

A. Operation of the model

The behavior of the temperature must be radial from the center at the energy outlet location of the evaporator node. Each node will have an energy extraction limit that matches with the phase change energy of each material in the confined volume as it is shown in Fig. 3. Fig. 4 helps to detail the energy transfer mechanism between nodes.

The first simulation was developed proposing an empty stage for the cooling chamber, without: shelves with water inside, any object or internal products. Here the logic predicts that there is a smaller amount of energy to be transported by the refrigeration systems which in consequence takes less time to get an internal thermal equilibrium at the refrigerant boiling point (see Fig. 5).

The empirical behavior is consistent with the theoretical radial propagation, hence the location of the evaporator to show the lowest values of temperature as shown in Fig. 2, depositing the element of higher density to the bottom of this. The empirical behavior is consistent with the theoretical radial propagation, hence the location of the evaporator to show the lowest values of temperature as shown in Fig. 3, depositing the element of higher density in the lower part of it. By combining both proposed equations, depending on where the type of material is located, it should be remembered that the proposed model is not considered as non-linear, all the characteristics developed throughout the phenomenon must be constant for a first approach with the operation of the refrigeration cycle and a greater understanding of which properties may have the greatest impact.

Fig. 4 shows the same behavior as Figure 2, but with a transverse view on the z-axis, divided into 9 different instants. Subfigures 1, 2 and 3, which are the first layers,

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

show a considerable decrease in internal energy. Layers 4, 5, 6 show a higher energy, but a decrease in the lower part of the chamber, due to the density of the air increases when the temperature decreases (energy is extracted from the air by the evaporator).

The red zone in Figure 4 indicates that after working at 100%, with a total time of 12 hours (simulated), the system does not reach the stable state. However, with 0% thermal load, the system stabilizes or reaches the same, because there is a minimum energy exchange that is very difficult to notice experimentally.

The numerical solution of the finite volume nodes for the temperature determines the coldest and hottest zones in the refrigerator chamber, depending on the current thermal load ratio and its distance from the evaporator location, without considering the flow rate of the evaporator existing energy in it.

IV. ANALYSIS OF RESULTS

The Fourier's equation (proposed model) solved in finite differences, holds a dynamic behavior like a first order system for all its nodes, based on the concept of finite volume and the sub-node strategy. The dynamic behavior obtained with the experimental characterization represents a behavior consistent to the empirical results.

The stabilization time of the model is in good agreement with the experimental results, since the model proposed in finite differences, in each iteration, is fed by temperature data at the output of the evaporator (sensor available in the refrigerator). However, the theoretical values of the constants require adjustments; to reduce the deviations in a steady state.

Based on the results obtained from the tests, 4 critical zones were chosen to analyze the dynamic response due to the thermal load, as well as the dynamic response due to the speed of the compressor (Fig. 7). In each zone, the corresponding dynamic poles are calculated from each time constant. Considering the poles obtained for the 4 regions of the load-velocity space, the errors with respect to the theoretical values must be considerably reduced.

V. CONCLUSIONS

The presented article develops a useful mathematical model and the results of a Matlab codification in forms of graphs.

A strong dependence has been observed between the coefficients of the model (thermodynamic properties) and the control variables that are regulated in the refrigerator device.

The speed of the compressor, as well as the distribution and magnitude of the thermal load inside the refrigerator, are variables that contribute to modify the temperature evolution of the system.

The non-linearity of the phenomenon must be considered when modifying the value of the poles in the model and is supported by the improvement of the results.

ACKNOWLEDGMENT

The authors thank the Mexican Council for Science and Technology (CONACYT) for its support of the student's scholarships No. 599179, 492895, 437556, 555423 and 486129 of the graduate students; also because of the support through the consolidation project for the National Research Laboratory on Cooling Technology (LaNITeF) budget 293784. These results were gotten by the synergy of both projects the LaNITeF consolidation as well as the P02 of the Mexican Center for Innovation in Eolic Energy (CEMIE Eolic), supported also by the secretary of Energy, through the budget for Sustainable Energy.

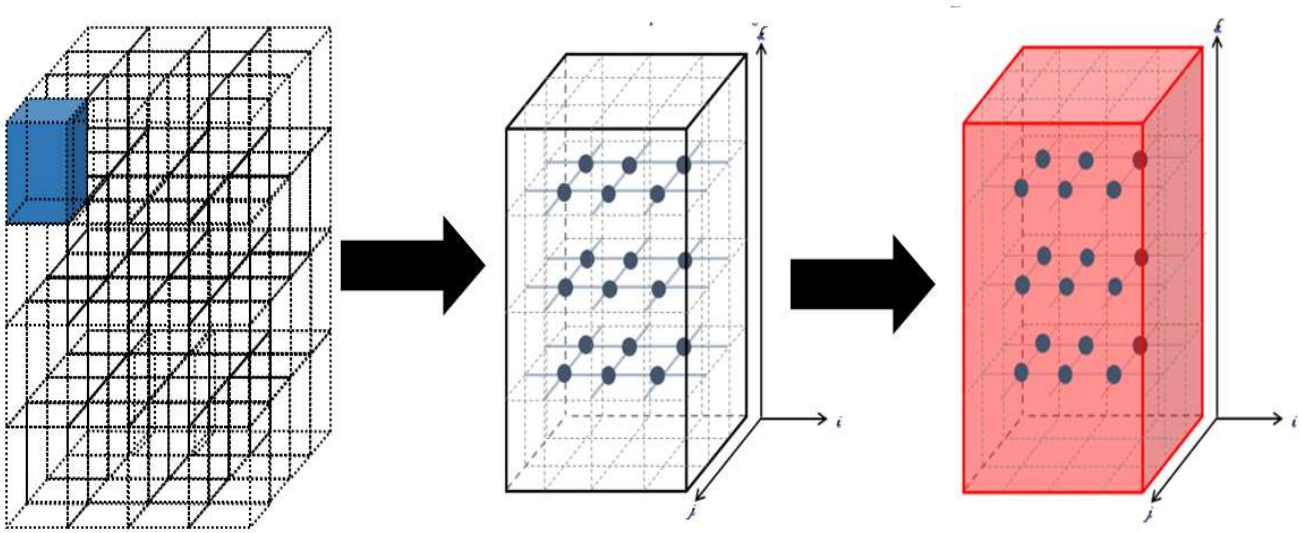


Figure 2. Decomposition in finite volumes

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

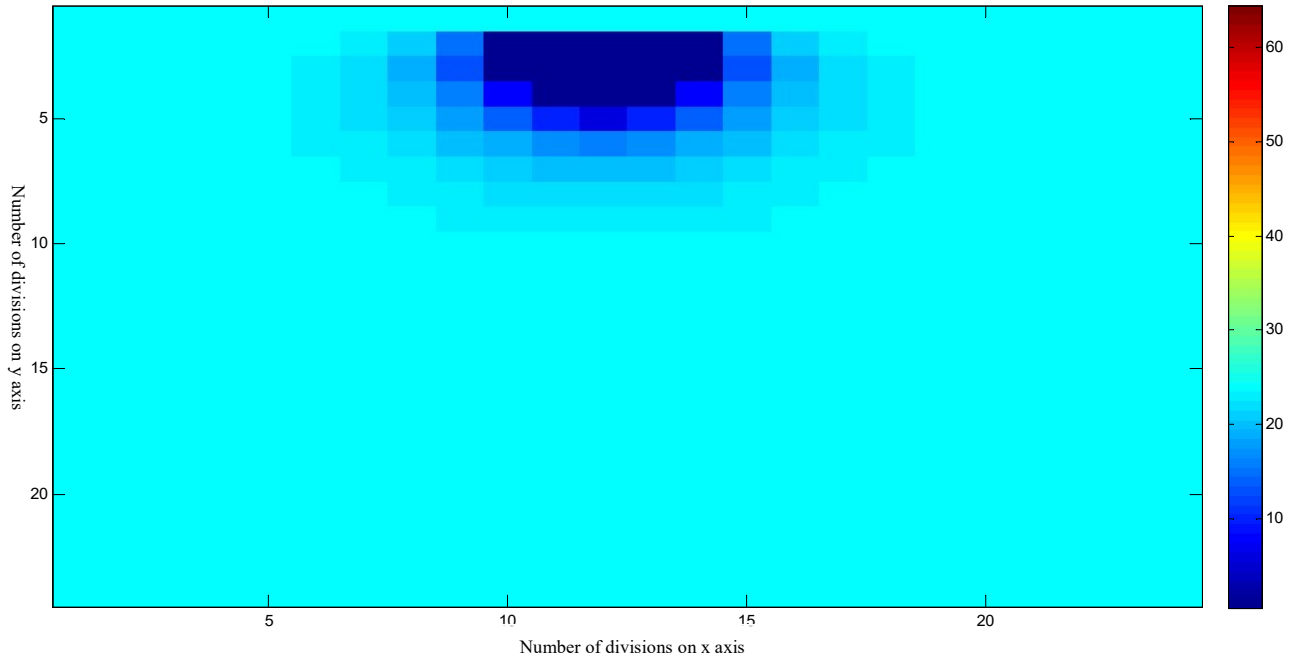


Figure 3. Prediction of the temperature distribution into the domestic refrigerator chamber

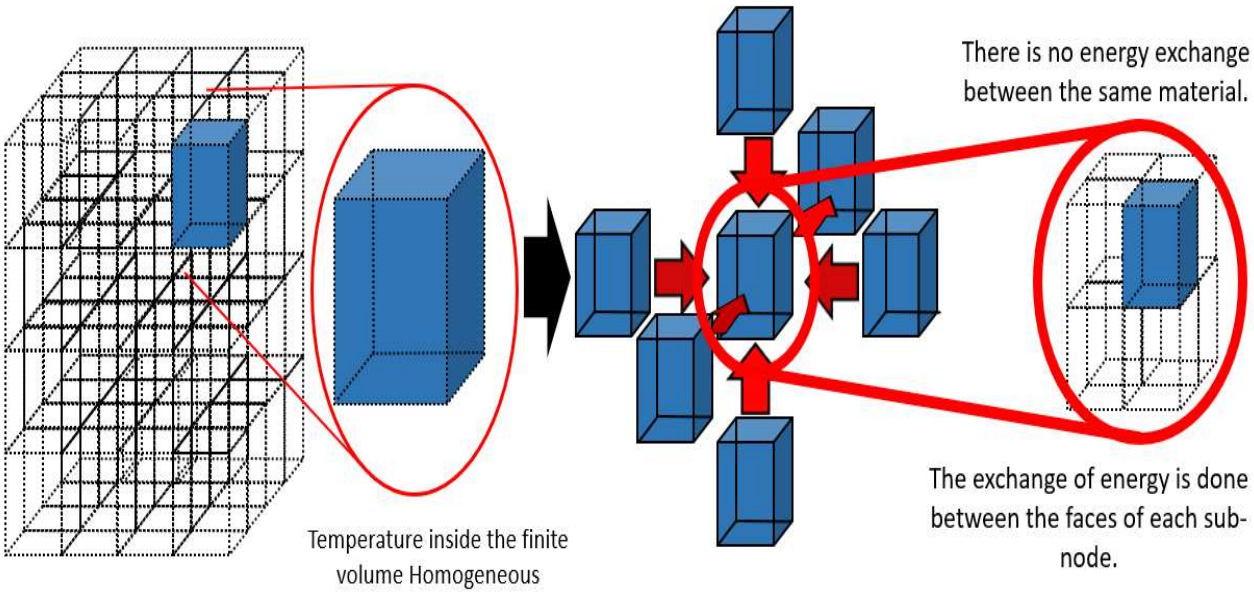


Figure 4. Heat exchange of cold chamber volume by finite volume

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

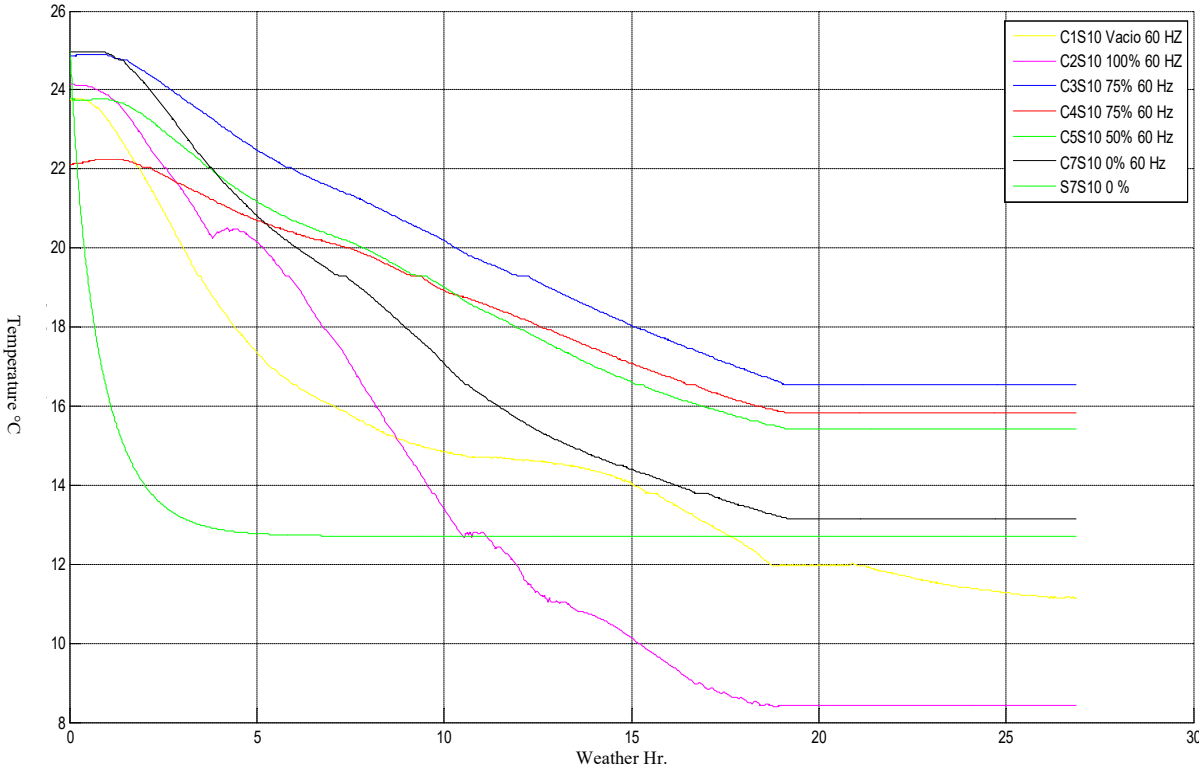


Figure 5. Experimental results of instrumented chiller chamber

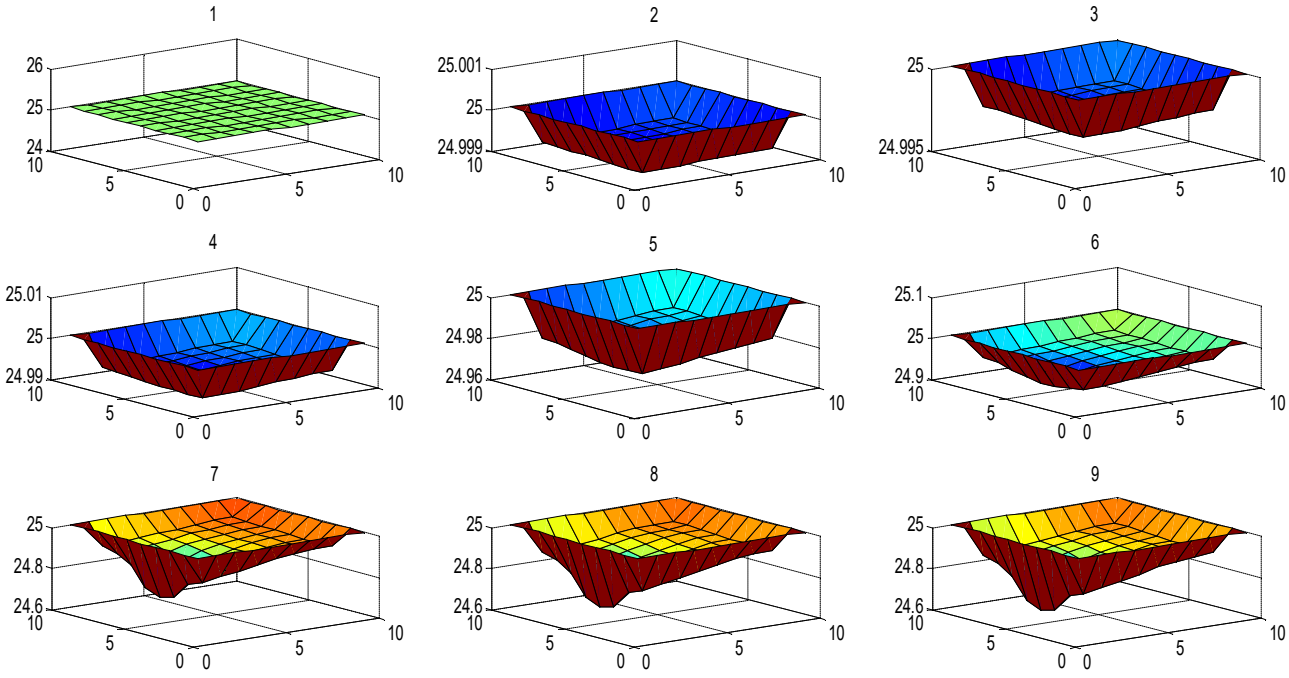


Figure 6. Behavior of the temperatures according to the height of the refrigerator

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

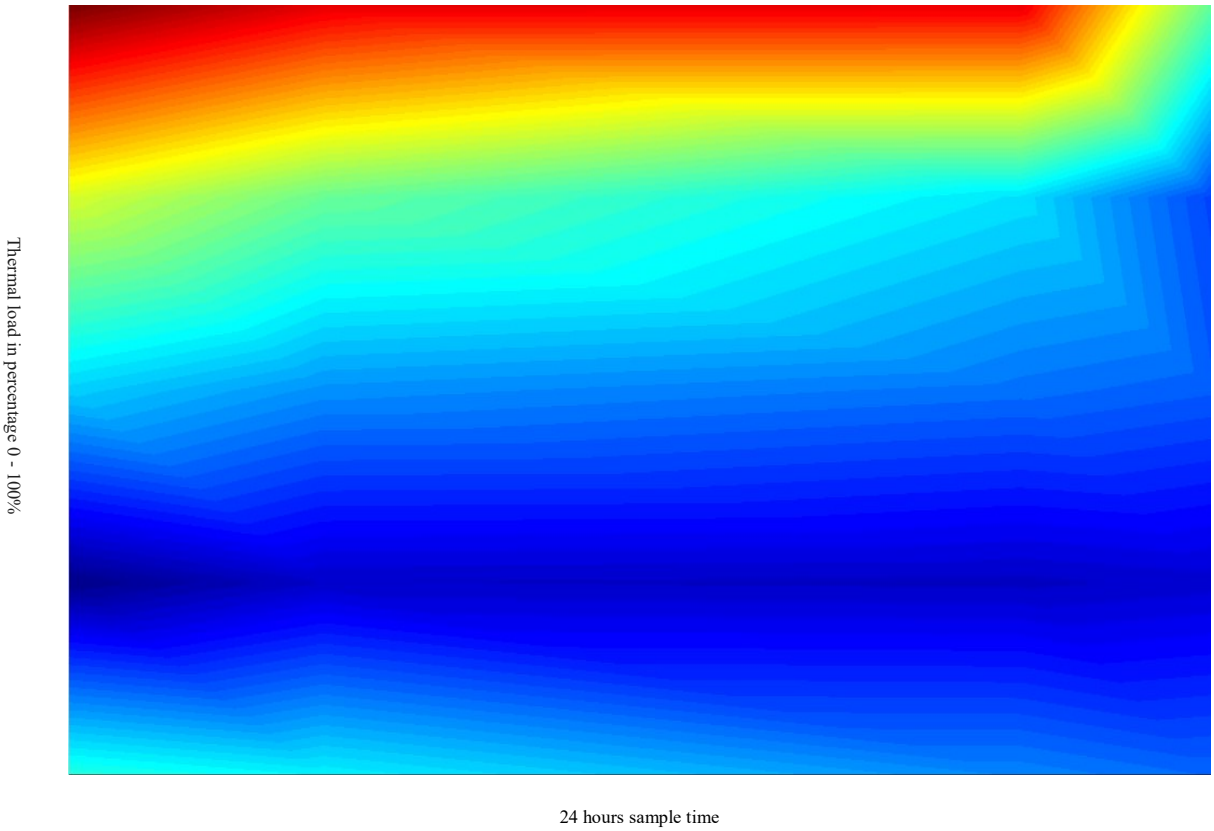


Figure 7. Distribution of the energy inside the cold chamber with different thermal loads.

REFERENCES

[1] Pérez Vázquez Jesús Hernán, Nuñez Martín Carlos Alexander, Zuñiga Osorio Helen Janeth, Uriza Gosebruch Ottmar Rafael and Cruz Gonzales Celso Eduardo, "Analysis of the dynamic behavior of the temperature distribution inside a domestic refrigerator," The Thirteenth International Conference on Systems (ICONS 2018) IARIA, Apr. 2018, ISBN: 978-1-61208-626-2, ISSN: 2308-4243.

[2] S. Lineykin and S. Ben-Yaakov, "Modeling and Analysis of Thermoelectric Modules," in *IEEE Transactions on Industry Applications*, vol. 43, no. 2, pp. 505-512, March-april 2007. doi:10.1109/TIA.2006.889813.

[3] S.A. Tassou, G. De-Lille, Y.T. Ge, "Food transport refrigeration – Approaches to reduce energy consumption and environmental impacts of road transport," Applied Thermal Engineering, Vol. 29, Issues 8–9, 2009, pp. 1467-1477, ISSN1359-4311, doi.org/10.1016/j.applthermaleng.2008.06.027.

[4] Florian Zink, Jeffrey S. Vipperman, Laura A. Schaefer, "Environmental motivation to switch to thermoacoustic refrigeration, Applied Thermal Engineering," Vol. 30, Issues 2–3, 2010, pp. 119-126, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2009.07.008.

[5] Ke Tang, Ye Feng, Tao Jin, Shenghan Jin, Ming Li, Rui Yang, "Effect of Gedeon streaming on thermal efficiency of a travelling-wave thermoacoustic engine, Applied Thermal Engineering," Vol. 115, 2017, pp. 1089-1100, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2017.01.054.

[6] Mathew Skaria, K.K. Abdul Rasheed, K.A. Shafi, S. Kasthurirengan, Upendra Behera, "Simulation studies on the performance of thermoacoustic prime movers and refrigerator," Computers & Fluids, Vol. 111, 2015, pp. 127-136, ISSN 0045-7930, doi.org/10.1016/j.compfluid.2015.01.011.

[7] Emmanuel C. Nsofor, Serdar Celik, Xudong Wang, "Experimental study on the heat transfer at the heat exchanger of the thermoacoustic refrigerating system," Applied Thermal Engineering, Vol. 27, Issues 14–15, 2007, pp. 2435-2442, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2007.03.008.

[8] P. V. Hernán and R. Eloy, "Structural model of the internal temperature distribution in a cold chamber," *Congreso Nacional 2016 PICYT, Leon Guanajuato*.

[9] P. Hernán, R. Eloy y N. Alexander, "Mathematical simplification of the heat extraction of a domestic refrigerator," Congress of Young Researchers, Queretaro 2016.

[10] F. Belmant, et. al., "Analysis of the temperature stratification of a no-frost domestic refrigerator whit bottom mount configuration," Applied Thermal Engineering 65 (2014) pp. 299-307, doi.org/10.1016/j.applthermaleng.2014.01.022

[11] Jara Nelson, et. al., "Dynamic model for the study of the application of the refrigeration renewal plan in Colombia" Universidad Salesiana Ecuador, 2016.

[12] Jara Nelson and Isaza Cesar. "Domestic refrigeration systems - State of the art of improvements in energy efficiency" University Salesiana Ecuador , 2016. http://www.indiaenvironmentportal.org.in/files/file/refrigerators%20energy%20consumption.pdf

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[13] Tassou, S. A., et. al., "A review of emerging technologies for food refrigeration applications," Applied Thermal Engineering 30, 2010,pp. 263,276. doi.org/10.1016/j.applthermaleng.2009.09.001

[14] Vincet CE and Heun MK, "Thermoeconomic analisys & design of domestic refrigeration systems", domestic use energy Conference 2006.

[15] Bromberg A., et. al., "Tips and techniques that do not harm the environment for the technical service of equipment with refrigerators" Mainstream Engineering Corporation, Rockledge,Florida. www.epatest.com/608/manual/Manual_SP.htm

[16] Belman, J., M., "Analysis of the flow and temperature distribution inside the comportment of a small refrigerator", 2016, doi.org/10.1016/j.applthermaleng.2016.06.065

[17] Björk, Erik and Palm, Björn, "Refrigerant mass charge distribution in a domestic refrigerator". Part II: Steady state conditions, 20, doi.org/10.1016/j.applthermaleng.2005.10.004

[18] Dongliang Zhao and Gang Tan, "A review of thermoelectric cooling: Materials, modeling and applications," Applied Thermal Engineering, Vol. 66, Issues 1–2, 2014, pp. 15-24, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2014.01.074.http://www.sciencedirect.com/science/article/pii/S1359431114000854

[19] J. L. Tang and Z. R. Ouyang, "Analysis on the cooling water system operation under the first time test of 40 tesla hybrid management", Cases studies in Termal engineering, Vol. 10, 2017, pp. 95 – 99, doi.org/10.1016/j.csite.2017.05.004

[20] Hongxia Xi, Lingai Luo and Gilles Fraisse, "Development and applications of solar-based thermoelectric technologies," Renewable and Sustainable Energy Reviews, Vol. 11, Issue 5, 2007, pp. 923-936, ISSN 1364-0321, doi.org/10.1016/j.rser.2005.06.008.

[21] Niccolò Aste, Claudio Del Pero and Fabrizio Leonforte, "Active refrigeration technologies for food preservation in humanitarian context – A review," Sustainable Energy Technologies and Assessments, Vol. 22, 2017, pp. 150-160, ISSN 2213-1388, doi.org/10.1016/j.seta.2017.02.014.

[22] Y.T. Ge and S.A. Tassou, "Performance evaluation and optimal design of supermarket refrigeration systems with supermarket model "SuperSim". Part II: Model applications," International Journal of Refrigeration, Vol. 34, Issue 2, 2011, pp. 540-549, ISSN 0140-7007, doi.org/10.1016/j.ijrefrig.2010.11.004.

[23] Kokouvi Edem N'Tsoukpoe, Daniel Yamegueu and Justin Bassole, "Solar sorption refrigeration in Africa, Renewable and Sustainable Energy Reviews," Vol. 35, 2014, pp. 318-335, ISSN 1364-0321, doi.org/10.1016/j.rser.2014.04.030.

[24] Edwin Garces, Kevin van Blommestein, Jamie Anthony, James Hillegas-Elting, Tugrul Daim, Byung-Sung Yoon, "Technology domain analysis: A case of energy-efficient advanced commercial refrigeration technologies,"

Sustainable Production and Consumption, Vol. 12, 2017, pp. 221-233, ISSN 2352-5509, doi.org/10.1016/j.spc.2017.08.002.

[25] Maxime Perier-Muzet, Jean-Pierre Bedecarrats, Pascal Stouffs and Jean Castaing-Lasvignottes, "Design and dynamic behaviour of a cold storage system combined with a solar powered thermoacoustic refrigerator," Applied Thermal Engineering, Vol. 68, Issues 1–2, 2014, pp. 115-124, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2014.03.065.

[26] Ahmed I. Abd El-Rahman, Waleed A. Abdelfattah and Mahmoud A. Fouad, "A 3D investigation of thermoacoustic fields in a square stack," International Journal of Heat and Mass Transfer, Vol. 108, Part A, 2017, pp. 292-300, ISSN 0017-9310, doi.org/10.1016/j.ijheatmasstransfer.2016.12.015.

[27] Konstantin Tourkov, Florian Zink, Laura Schaefer, "Thermoacoustic sound generation under the influence of resonator curvature," International Journal of Thermal Sciences, Vol. 88, 2015, pp. 158-163, ISSN 1290-0729, doi.org/10.1016/j.ijthermalsci.2014.09.016.

[28] Hadi Babaei, Kamran Siddiqui, "Design and optimization of thermoacoustic devices," Energy Conversion and Management, Vol. 49, Issue 12, 2008, pp. 3585-3598, ISSN 0196-8904, doi.org/10.1016/j.enconman.2008.07.002.

[29] A.C. Alcock, L.K. Tartibu and T.C. Jen, "Experimental Investigation of Ceramic Substrates in Standing", Wave Thermoacoustic Refrigerator, Procedia Manufacturing, Vol. 7, 2017, pp. 79-85, ISSN 2351-9789, doi.org/10.1016/j.promfg.2016.12.021.

[30] Matthieu Guédra, Flavio C. Bannwart, Guillaume Penelet and Pierrick Lotton, "Parameter estimation for the characterization of thermoacoustic stacks and regenerators", Applied Thermal Engineering, Vol. 80, 2015, pp. 229-237, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2015.01.058.

[31] Chao Shen, Hong-Xin Li, Dong-Wei Zhang, Peng Yu, Na Pan and Shuang-Feng Wang, "Study on the heat transfer characteristic of solar powered thermoacoustic prime mover at different tilted angles", Applied Thermal Engineering, Vol. 103, 2016, pp. 1126-1134, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2016.04.096.

[32] Kai Wang, Jie Zhang, Ning Zhang, Daming Sun, Kai Luo, Jiang Zou and Limin Qiu, "Acoustic matching of a traveling-wave thermoacoustic electric generator," Applied Thermal Engineering, Vol. 102, 2016, pp. 272-282, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2016.03.106.

[33] K. Tang, G.B. Chen, T. Jin, R. Bao, B. Kong and L.M. Qiu, "Influence of resonance tube length on performance of thermoacoustically driven pulse tube refrigerator," Cryogenics, Volume 45, Issue 3, 2005, pp. 185-191, ISSN 0011-2275, doi.org/10.1016/j.cryogenics.2004.10.002.

[34] Shinya Hasegawa, Tsuyoshi Yamaguchi and Yasuo Oshinoya, "A thermoacoustic refrigerator driven by a low temperature-differential, high-efficiency multistage thermoacoustic engine," Applied Thermal Engineering, Volume 58, Issues 1–2, 2013, pp. 394-399, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2013.04.030.

[35] Ye Feng, Ke Tang, Tao Jin and Kaihao Zhang, "Impact of cross-sectional area ratio on time-averaged pressure drop induced by jet pump for thermoacoustic engine," Energy Procedia, Vol. 142, 2017, pp. 337-342, ISSN 1876-6102, doi.org/10.1016/j.egypro.2017.12.053.

[36] Daming Sun, Kai Wang, Yinan Guo, Jie Zhang, Ya Xu, Jiang Zou and Xiaobin Zhang, "CFD study on Taconis thermoacoustic oscillation with cryogenic hydrogen as working gas", Cryogenics, Vol. 75, 2016, pp. 38-46, ISSN 0011-2275, doi.org/10.1016/j.cryogenics.2016.01.004.

[37] Linyu Li, Zhanghua Wu, Jianying Hu, Guoyao Yu, Ercang Luo and Wei Dai, "A novel heat-driven thermoacoustic natural gas liquefaction system. Part I: Coupling between refrigerator and linear motor, Energy," Vol. 117, Part 2, 2016, pp. 523-529, ISSN 0360-5442, doi.org/10.1016/j.energy.2016.06.022.

[38] Ekhlas Alkhwildy and Issa Morad, "Impact Factor in Thermoacoustic Prime Mover Design, Energy Procedia", Volume 74, 2015, pp. 643-651, ISSN 1876-6102, doi.org/10.1016/j.egypro.2015.07.800.

[39] K. Ghorbanian and M. Karimi, "Design and optimization of a heat driven thermoacoustic refrigerator," Applied Thermal Engineering, Vol. 62, Issue 2, 2014, pp. 653-661, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2013.09.058.

[40] Tao Jin, Rui Yang, Yi Wang, Yuanliang Liu and Ye Feng, "Phase adjustment analysis and performance of a looped thermoacoustic prime mover with compliance/resistance tube," Applied Energy, Vol. 183, 2016, pp. 290-298, ISSN 0306-2619, doi.org/10.1016/j.apenergy.2016.08.182.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[41] Tao Jin, Rui Yang, Yuanliang Liu and Ke Tang, "Thermodynamic characteristics during the onset and damping processes in a looped thermoacoustic prime mover," Applied Thermal Engineering, Vol. 100, 2016, pp. 1169-1172, ISSN 1359-4311, doi.org/10.1016/j.applthermaleng.2016.02.115.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Towards a Knowledge-intensive Framework for Efficient Vaccine Development

Leonard Petnga
Department of Industrial and Systems Engineering
University of Alabama in Huntsville
Huntsville, AL 35899, USA
Email: leonard.petnga@uah.edu

Surangi Jayawardena
Department of Chemistry
University of Alabama in Huntsville
Huntsville, AL 35899, USA
Email: surangi.jayawardena@uah.edu

Abstract—**Modern vaccine research & development efforts are complex, long, costly undertakes with high rate of failure. A major cause of inefficiency can be attributed to the mostly unstructured, unorganized, disconnected and diversity of knowledge spread across the Vaccine Development Life Cycle (VDLC) and honed by number of stakeholders with conflicting interests. State-of-the-art approaches have mostly fostered stove-piping knowledge and information within individual disciplines and separation of concerns, with little interest or appetite for cross-domain knowledge integration. In this research, we build on the ability of systems engineering to bridge the gaps between (and integrate) other disciplines to architect and develop a novel knowledge-intensive framework for efficient vaccine development. We formulate a model-based platform that accounts for the need for, (1) formalisms to support unambiguous and correct knowledge representation and reasoning across the VDLC, (2) capturing stochastic system biology behaviors and integrating with stakeholders' discrete decisions and, (3) models that are formal, reusable, customizable and can be assembled as needed for the purpose of the analysis at hand. Description logic-based formalisms and foundational domain theories support knowledge models of domains tightly coupled with Markov models of biological and chemical processes are the cornerstone of our framework. An example step-by-step implementation procedure illustrates the modularity, flexibility and configuration of the framework for tackling increasingly complex, cross-domain challenges across the VDLC. Vaccine preservation laboratory experiments are conducted to assess some prototype formulations and generate system biology models to be integrated with semantic models in the platform. Results are very encouraging but further work is needed in identifying and mapping all relevant biological system behaviors for the analysis under consideration and improving their characterization and integration in the framework.**

Keywords-Vaccine; Knowledge formalisms; Systems Engineering; Semantic; Ontology; Markov Chain; Multiple Regression Model.

I. INTRODUCTION

This work is concerned with the development and prototyping of a framework based on knowledge description formalisms and stochastic modeling of biological systems for improved efficiency across the Vaccine Development Lifecycle (VDLC). It stems from and, extends previous work on Knowledge-driven Vaccine Systems Engineering [1]. Scientific breakthroughs in biotechnology and genetic decoding as well as advances in information technologies and computation have spurred the acceleration of vaccine development. This can be observed in the wide range of technologies and techniques used to develop modern vaccines, which can now target over 25 infectious and non-communicable diseases. Genome-based approaches have enabled the development of vaccines for Meningococcus B or the development of the first ever therapeutic vaccine (for prostate cancer). Similarly, (conjugate) vaccines with multiple antigens or strains now allow for broadened protection while reducing the required number of injections [2][3]. With more than 2.5 million child deaths/year prevented, billions in healthcare cost savings and multiple outcome-related productivity gains, vaccination has become a cornerstone of modern human being of all ages (and financial) health [4]. Looking at vaccine research & development pipelines in Big Pharmaceutical companies, ongoing efforts aim at developing vaccines for more than 50 bacterial, viral, parasitic, degenerative and addictive diseases. This effort includes vaccines against the top 3 killers in developing countries, i.e., lower respiratory infection, HIV/AIDS, and Diarrheal diseases [5].

While population and health professionals rejoice, researchers are faced with mounting challenges hindering the vaccine development life cycle. Among the challenges are: (1) the knowledge disconnect between the disciplines involved – biology, chemistry, engineering, manufacturing, legal, regulator and healthcare – and between stakeholder's views (see Figure 1), which makes the development process very convoluted; (2) the need for sustained capital investment over a lengthy period – hundreds of millions of dollars and 8 to 10 years from research to market with high failure rates – for vaccine development; (3) the costly and stringent storage and handling conditions to be satisfied in order to reduce the loss in vaccine potency and expand an otherwise very short shelf life span (i.e., a year or less); (4) genetic mutations and constantly evolving environment factors making it difficult for certain vaccines to be produced (e.g., HIV-1) [6][7]. A unified, formal, vaccine knowledge-driven approach for the VDLC is needed to enable stakeholders along the VDLC to answer both domain specific and cross-domain questions, quickly, accurately and cheaply, in the context of highly stochastic and complex biological dynamics.

In this project we take a significant step towards a novel knowledge-intensive framework for efficient vaccine development. Our objective is to develop the foundational semantic infrastructure for knowledge and behavior specification, mod-

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

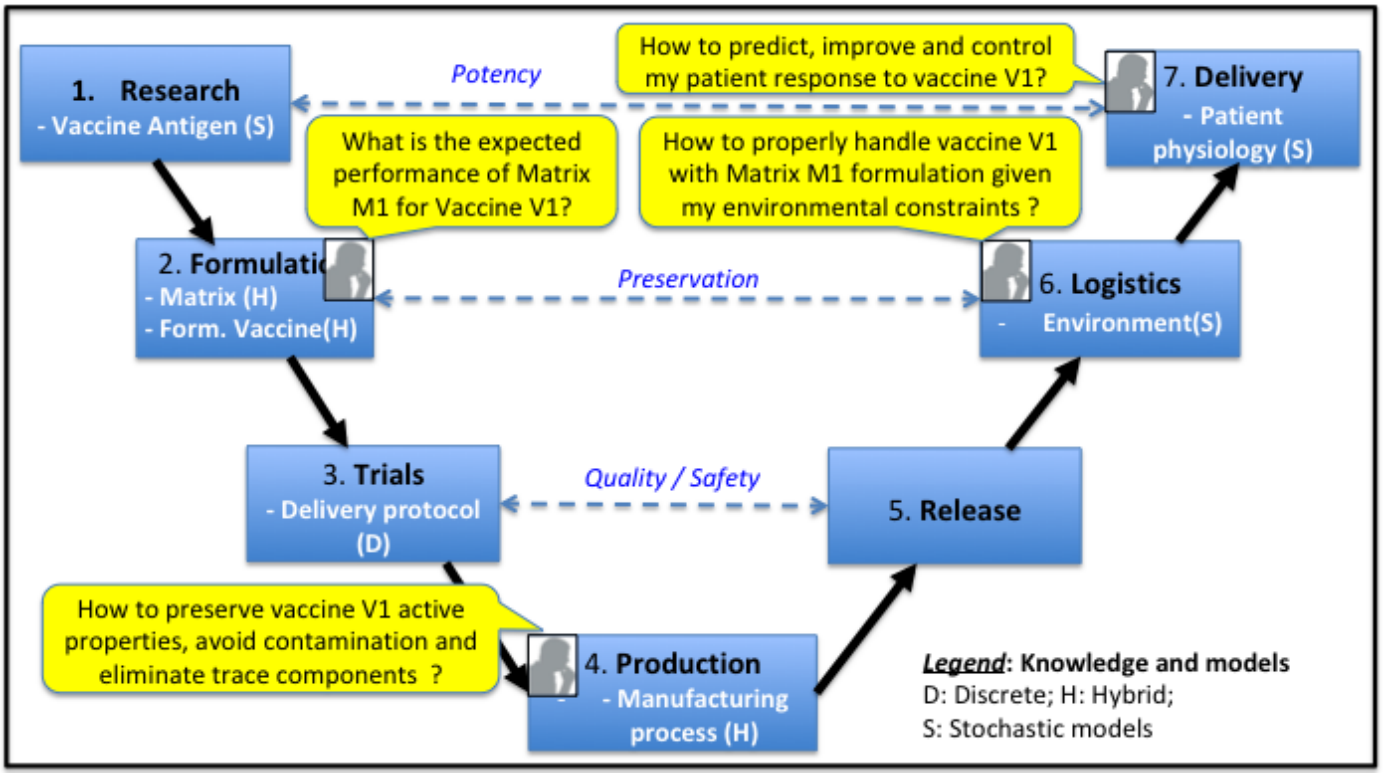


Figure 1. Vee-model of the Vaccine Development Lifecycle and Current Knowledge Gaps

eling and processing across the VDLC as a whole. Therefore, we build on the discipline of systems engineering ability to bridge the gaps between (and integrate) other disciplines, to architect a knowledge-enabled vaccine development platform. The guiding principles of our approach are as follows: (1) formal methods must drive and support the development of vaccine domain models, (2) the latter must properly capture the depth and breadth of stochastic behaviors of biological systems and, (3) models must be reusable, customizable and integrated at will for the purpose of the analysis at hand. The resulting platform supports the integration of biological system dynamic models and, discipline and stakeholder knowledge models thus, enables the emergence of novel architectures, which instantiation can be performed and executed against the requirements of a given application or analysis. Section II is a review of vaccines typology, mechanisms and existing development approaches. Section III introduces mathematical foundations for the formal representation and description of vaccine knowledge and systems biology. Section IV describes the architecture of the framework along with a simplified software implementation infrastructure. An experimental vaccine case study is introduced in Section V to illustrate some of the core capabilities of the framework. The paper concludes with discussions, conclusions and future work.

II. State of the Art: Typology, Mechanisms and Development Lifecycle of Vaccines

Vaccine has been playing a hugely important role in preventing infectious and non-communicating diseases and improving overall quality of living. However, for a vaccine to be successful, (1) its active ingredients should induce an effective and sustained immune response, (2) it must have minimal side effects and, (3) it must be produced cost-effectively at a large scale. Because of the complex nature of vaccine manufacturing it is important to understand and control and or, predict the factors that impacts the efficacy, stability and safety of the vaccine along its process-engineering pathway.

A. Typology and Composition of Vaccines

They are mainly three classes of vaccines. Most conventional or the first generation of vaccines consists of a live, but attenuated form of the pathogen or an inactivated pathogen. *Live, attenuated vaccines* – consists of live viruses that have been extensively passaged through animal hosts until an acceptable balance has been retained between the loss of virulence and retention of immunogenicity. *Inactivated vaccines* – contains microorganisms that have been treated to destroy their infectivity (inactivation). The second generation of vaccines consist only a part of the pathogen – subunit vaccines. *Subunit vaccines* – consists of epitopes around external surface of the pathogen. With recent advances in vaccine science, a third generation vaccines have emerged as DNA and recombinant vector vaccines. *DNA vaccines* consist of non-replicating plasmids, which contain DNA that encodes specific proteins (antigens) from a pathogen. *Recombinant viral vectors vaccine* works by enabling an intracellular antigen expression in the body. Figure 2 illustrates the most common components found in modern vaccines at delivery point. The main components play various functions needed to enable the trigger, execute and maintain host immunization including, (1) elicit and enhance immune response (active ingredients and adjuvants respectively), (2) ensure the stability of various

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

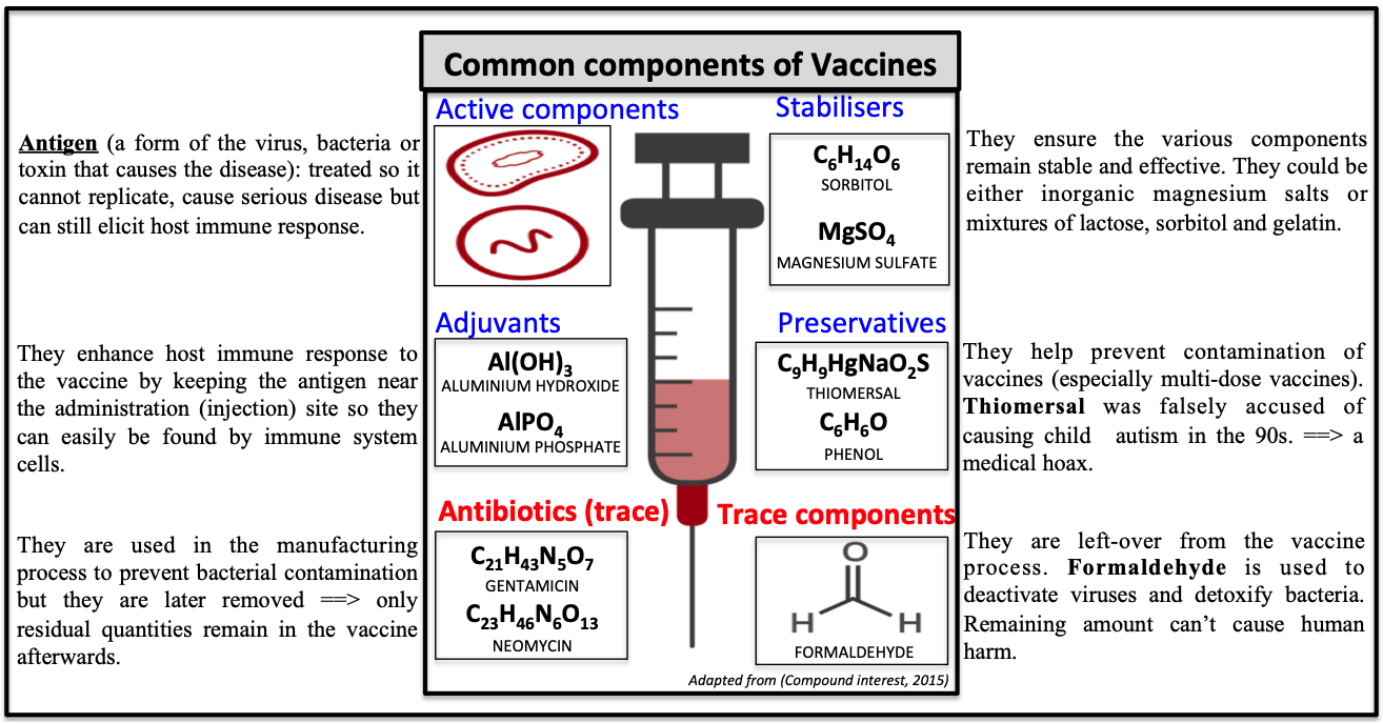


Figure 2. Common components of vaccines(Adapted from [8])

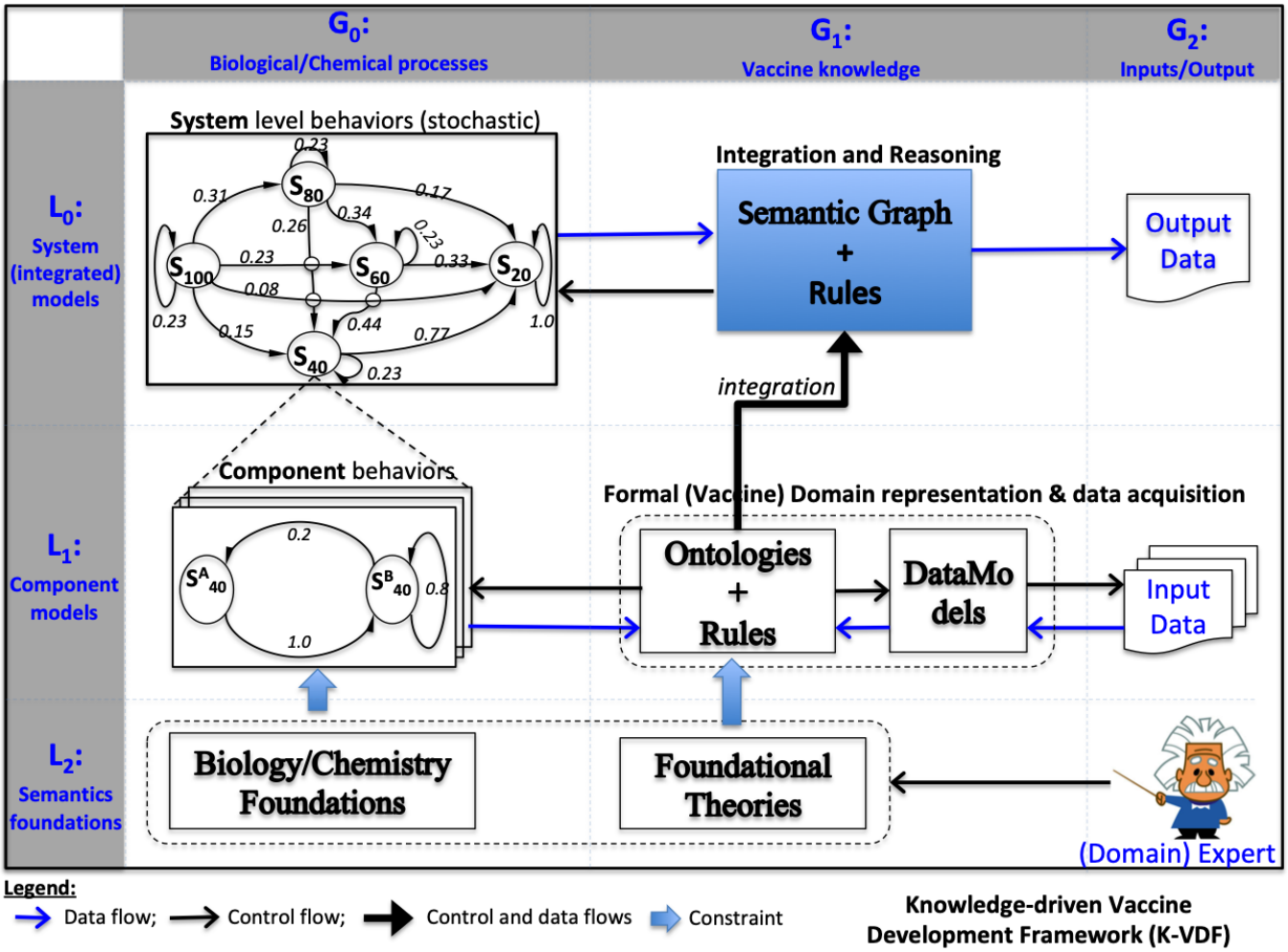


Figure 3. Knowledge-driven vaccine development framework system architecture.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

components (stabilizers) and, (3) protect the vaccine against contamination(preservatives). Other components Other inactive elements (e.g., antibiotics and trace components) are inherited from the development and manufacturing process.

B. Mechanisms of Immunization

A vaccine raises immunity through a complex process – yet to be fully understood – in the body. Vaccine protects by inducing immune mechanisms capable of rapidly controlling replicating pathogens or inactivating their toxic components. One immune mechanism is raising antibodies against the vaccine antigen. For example the antigen in an inactivated viral vaccine is the inactivated virus. Vaccine antigens can stimulate a number of cells in the immune system, which includes macrophages, T cells, and B cells. An immune response begins when macrophages ingest the antigen. Fragments of the digested antigen are displayed on the surface of the macrophage. These displayed antigen fragments are recognized by T lymphocytes helper cells, which stimulate B lymphocyte cells to secrete antigen specific antibodies. T helper cells activate killer T cells to actively bind and destroy the antigen. Long-term protection is given by immune memory cells (B and T memory cells), capable of rapidly and effectively re-activating the production of antibodies and killer T cells. Despite the success of three generations of successful vaccines that have eradicated small pox and nearly eradicated polio, there is still a great need for new vaccines and these are emerging far more slowly than we would wish. Successful immunization is not only influenced by various immunological factors (including host physiology and the type and nature of antigen), but also by formulation and delivery aspects.

C. Vaccines Development

The development of vaccines is a complex, lengthy and extremely expensive process involving public, private and non-profit players. It is a high-risk undertaking as many of vaccine candidates fail in preclinical studies. Also, regulatory, technical and manufacturing hurdles lie in the path that translates a vaccine candidate to the final vaccine available in clinics for administration. There are multiples closely coupled stages in vaccines development. The first stage involves candidate selection (exploratory stage) from a fundamental research laboratory – conducting the discovery of antigens – and testing the candidate among animal models. During the pre-clinical stage, development of small case scale material and formulation – a prototype vaccine "medicine" – is done to make material for phase I, II and III studies (clinical development). The exploratory and preclinical can last 1 to 10 years and cost between $10 and $20 millions. Phase I includes test of safety among a sample of 10-100 human subjects to evaluate clinical responses. Phase II focuses on the evaluation of immune responses in a sample of 100-3000 subjects and large scale studies are conducted in phase III to test vaccine efficacy and tolerance. In a clinical development of material candidate, the candidate is cultured, harvested, inactivated (in certain cases), formulated and filled (free dried or in liquid form) in vials syringes packaged and released for distribution. Clinical trials coupled with regulatory approvals can last between 4 to 7 years and are highly capital extensive, with costs in hundreds of millions of dollars.

III. Mathematical Foundations for Vaccine Knowledge Formalization and Systems Biology

A. Knowledge Representation Formalisms

Knowledge representation formalisms are needed to properly capture and formally represent a domain (e.g., vaccine) knowledge as well as reasoning on it. Over the years, researchers have developed several such formalisms including Semantic Networks [9], Frame Systems [10], Description Graphs [11] and Logic-based formalisms [12]. The declarative part of frame systems – a class of logic-based formalisms – are credited for the rise and development of modern Artificial Intelligence (AI) formalisms. Modal and description logics (DL) are descendants of such systems. DL appears to be the most appealing logic-based formalisms for framework like ours, thanks in part to its flexibility of extension to enable complex domain descriptions and the capability to support multi-values attributes and reasoning (for some subsets). Such features are critical to enable the formal representation of a heterogeneous and intricate domain as the vaccine, at various levels of abstractions. Also, we note that some results for description logics were found by translating results from variations of modal logics (propositional dynamic logics, μ-calculus) into description logics [13].

B. Description Logic Semantics and The Semantic Web

Capturing vaccine knowledge and crossing the divide between disciplines along and across the development life cycle depicted in Figure 1 requires mechanisms not just to represent, but also to integrate, share and reuse knowledge across the various stages of the process. Knowledge must (1) be captured, represented in a clear, unambiguous way with respect to the associated domain and the context of use and, (2) lend itself to automated processing and reasoning by machines. This requires data to be enriched and backed by sound semantics to ensure accuracy of facts and inferencing.

Description Logics (DL) formalisms, as fragment of first order logics, provide the sound mathematical foundations and decidability needed to tackle the first part of this challenge [15]. A brief definition of key DL concepts and its $\mathcal{ALC}$ extension are introduced in the appendix of [14]. The strong mathematical foundations of DLs enable the development of machine and human readable ontological languages, such as the web ontology language (OWL), in a systematic way. OWL is the language of choice for creation of ontologies, which are engineering artifacts specifying the intended meaning of a vocabulary used to describe a given domain (e.g., vaccine). As such, ontologies provide explicit semantic meanings that enrich the way models can be branched and integrated across domains of knowledge automatically. Understanding the intricate relationships spanning the vaccine domain and their ultimate affects on vaccine effectiveness will greatly benefit from these capabilities. In [14], $\mathcal{SHOIN}$ and $\mathcal{SROIQ}$ DLs (respectively mapped to OWL1-DL and OWL2 DL) have been identified as appropriate logic-based formalisms for knowledge-driven frameworks such as the one introduced in this work. The computational decidability of OWL2 DL makes it a suitable language for the development of ontologies in our framework.

The second part of above-mentioned challenge can be addressed using semantic web technologies integrated with

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

reasoner through Application Programming Interfaces or API (e.g., Jena). Semantic web technology resources are organized as a stack, where technologies such as the eXtensible Markup Language (XML), the Resource Description Framework (RDF) and OWL provide the necessary foundations needed by the one on the top, in hierarchical layers. The stack enables the implementation of reasoning that can prove whether or not assertions in the knowledge base are true or false in almost real-time (decidability). Therefore, semantic web technologies are the mean by excellence for automated processing and reasoning over a high variety of distributed and heterogeneous across-domains information such as the ones encountered in vaccine development. Specifically, the various sources of information will be organized, formalized and merged accordingly using semantic models (ontologies, rules and computation extensions) and reasoning will be performed to answer simple and complex biological and/or engineering questions.

C. Stochastic Modeling of System Biology

Beyond the formal description of their structure and properties, the effective capture of the behavior of biological systems (e.g., vaccine antigen, host physiology) across the VDLC is needed to accurately represent and understand the essence of unfolding biological (and underlining) chemical processes at various level of abstractions. Therefore, there is a need for models that can allow for the simulation of the system behavior over time, propagate and predict changes from interactions within systems and with the environment. Researchers have developed and introduced various modeling schemes of biological phenomenons and systems with emphasis on aspects such as body metabolism, neuronal systems, genetic networks or processes (e.g., intracellular processes). Resulting models work fine for cellular level analyses and studies but they are ineffective higher levels of abstractions (e.g., tissue, organs) biological phenomenons [16]. Thus, in order to address those limitations, we opt for a more general formalism – Markov models – in our framework. Such models have been shown effective, in previous work, in modeling and predicting the behavior of highly stochastic biological [17] and biomedical systems [18]. Moreover, they are domain independent and well-suited for integration through segmentation mechanism to domain specific models. In this work, we will use *Markov chain (MC)* formalism to represent actual biological or chemical behavior as a network of states as nodes and directed edges representing allowable transitions between states annotated with their probability of propagation. The graph on the top left corner of Figure 3 – illustrates such MC model. In MCs, feedback and steady-states are allowed as long as all propagation probabilities at each state sum up to 1. A variant of Markov models – *Hidden Markov models* – extend MCs and are suitable for observed system performance (e.g., lab experiments) studies. Markov models, when properly developed and analyzed, are powerful for analysis and prediction of complex system behaviors.

IV. System Architecture and Software Infrastructure

In this section, we introduces and briefly describes the architecture of the proposed framework at the core of effort towards for efficient vaccine development. It is built on top of the mathematical foundations introduced in Section III applied to the vaccine domain knowledge as introduced in Section II. Also, it mirrors a simplified software infrastructure that can enable its deployment at increasingly higher scales and levels of complexity.

A. Overview

The system architecture consists of modules to be assembled as per the needs of the analyses as illustrated in Figure 3. The modules lie at the intersection of three groups of vaccine knowledge categories and three layers of abstractions mirroring various levels of representation of the system. In the first group (G_0), knowledge of component and system biological/chemical dynamics constrained by relevant corresponding (abstract) foundational theories is captured and represented using Markov chains. Knowledge in the second group (G_1) is mostly the formal representation of vaccine, other related domains (e.g., gene, DNA) and foundational fields (e.g., time, space) knowledge as constrained by the corresponding theories. The last knowledge group (G_2) comprises the actual problem input data, the semantically enriched output data resulting from the analysis as well as structured and unstructured (domain) expert knowledge. In the knowledge-intensive framework, not all knowledge types or groups are created equal. They interact with each others – each playing different role – within and across groups to enable the desired functionality of the framework through its analysis-oriented configuration. The main layers of the infrastructure where the various modules are assembled are as follows.

B. Semantic Foundation Layer (L_2)

It provides the mathematical foundations needed by models to ensure effective and unambiguous description of both the domains involved in the analysis and biological/chemical phenomenons. We distinguish foundational theories for known cross-cutting domains (such as time, physical quantities or communication) in module (L_2, G_1) from laws governing biological and chemistry processes in module (L_2, G_0). The Allen Temporal Interval Calculus (ATIC) is a well-suited cross-domain theory that has been shown effective for formal description and reasoning in the temporal domain [14]. In the absence of a valid theory to support the formal description of a domain in the framework, well-accepted domain standards (e.g., CDC Standard for Adult Immunization Practice) as well as heuristics and expert knowledge can be used to fill the void. This offers the possibility for the modelers to inject new theories in the framework for test or evaluation purposes and assess their effectiveness or suitability for given family of problems/analysis. However, the scope and depth of knowledge to be used depends on the application of interest and the goals pursued by the modelers/researchers.

C. Component Layer (L_1)

The component layer enables the modeler to makes use of the formalisms provided by the semantic layer below (i.e., L_2) to create and manage domain knowledge and behavior models that can be reusable across applications. In the context of the VDLC, the knowledge (see module (L_1, G_1)) can be organized and classified in three categories based on their function in a modular way. Core domain (e.g., vaccine antigen, host) knowledge is segregated from cross-cutting domains

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

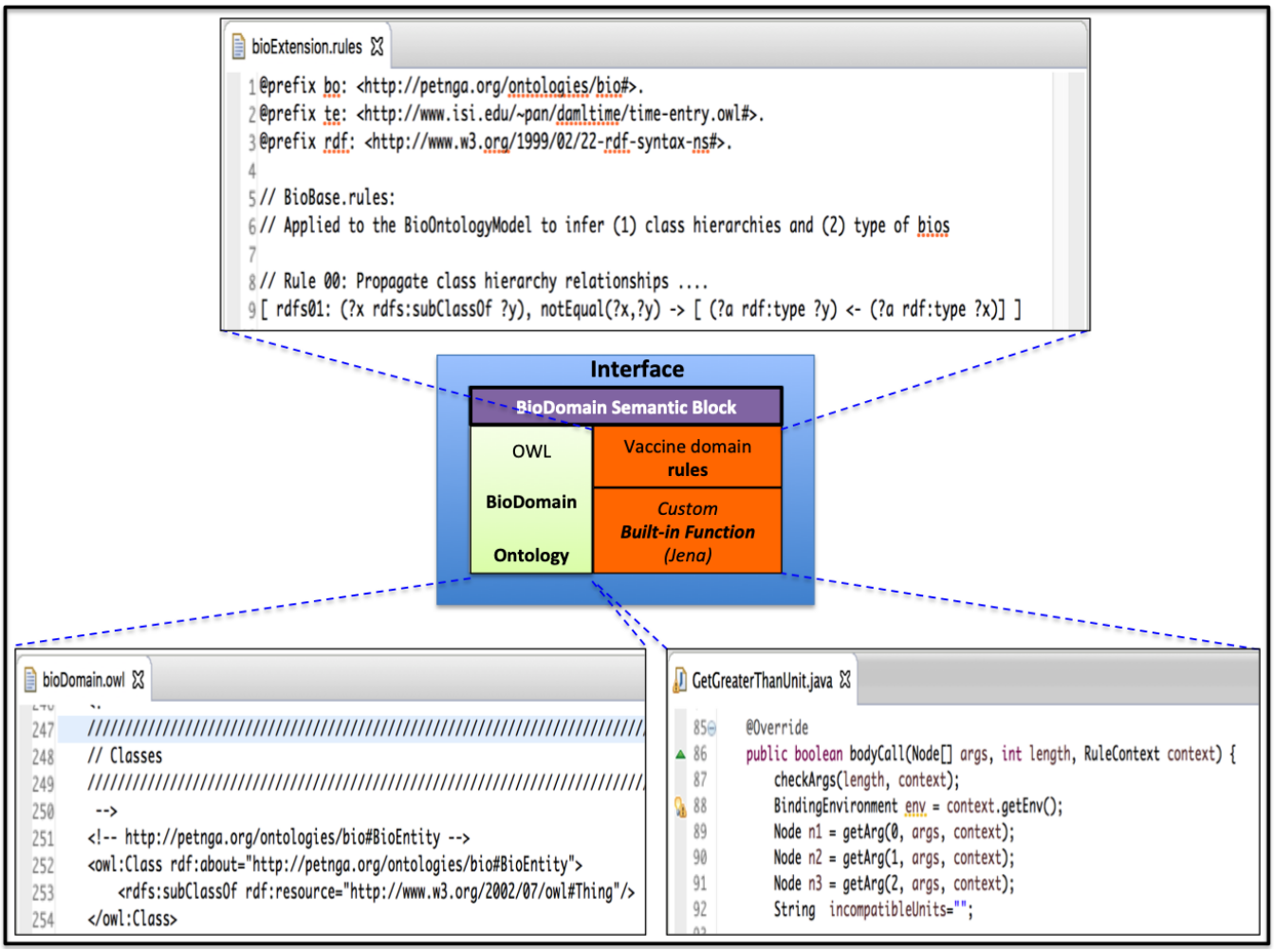


Figure 4. Schematic of a semantic block for the formal description and reasoning about the Vaccine (bioDomain).

(e.g., storage condition, vaccine schedule) knowledge, which in turn, is separated from foundation domain (e.g., time, physical quantity) knowledge. DLs provide the formalisms needed by core domains knowledge while theories such as the ATIC will constraint models of some cross-cutting domains (e.g., vaccine schedule, clinical trial planning). Each domain knowledge in encoded into a "Semantic block" that encapsulates (domain) knowledge in a formal and well-defined manner. Each of the blocks is made of, (1) a domain ontology, (2) set of domain rules, (3) custom computation functions and, (4) interfaces that enable communication between semantic blocks as illustrated in Figure 4. The built-in functions are the glue linking the ontologies to specialized computation platforms and Markov models of system biology (in module (L_1, G_0)) via domain rules as encoded by the reasoner's rules engine. Data-models are templates interfacing input data and ontologies. They enable the modeler to draw from the problem's data stored in input files (module (L_1, G_2)) then, populate the ontology with initial facts in an accurate, systematic and traceable manner. This modular approach adds further rigor and flexibility in the ability of the modeler to build complex applications using reusable semantic blocks (as composite knowledge model).

D. System Layer (L_0)

Leveraging the capabilities of the framework requires bringing together its various modules and pieces in an organized but systematic way. This is needed to close the knowledge gaps between disciplines and stakeholders in along the VDLC as discussed in Section I and answer increasingly complex questions as pictured in Figure 1. Therefore, two tasks need to be performed, i.e., (1) integrate various domain specific knowledge at level L_1 on both the semantic and stochastic behavior sides and, (2) link them and configure the framework accordingly to emulate system level behavior for the application under consideration. Next, the resulting semantic graph (module (L_0, G_1)) is transformed as rules – integrated to stochastic models of the system behavior (module (L_0, G_0)) – are fired. Here, a linkage between the system and component level behaviors to ensure consistency in representations. An "integrator" semantic block can be used as a "semantic controller" that encodes defined system metrics whose instances are checked against system requirements (as constraints). Given the complexity of the integration task, advanced computation capabilities – for controlled and systematic assembly of the models as well as simulation and output generation – such as the ones provided by the Whistle

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

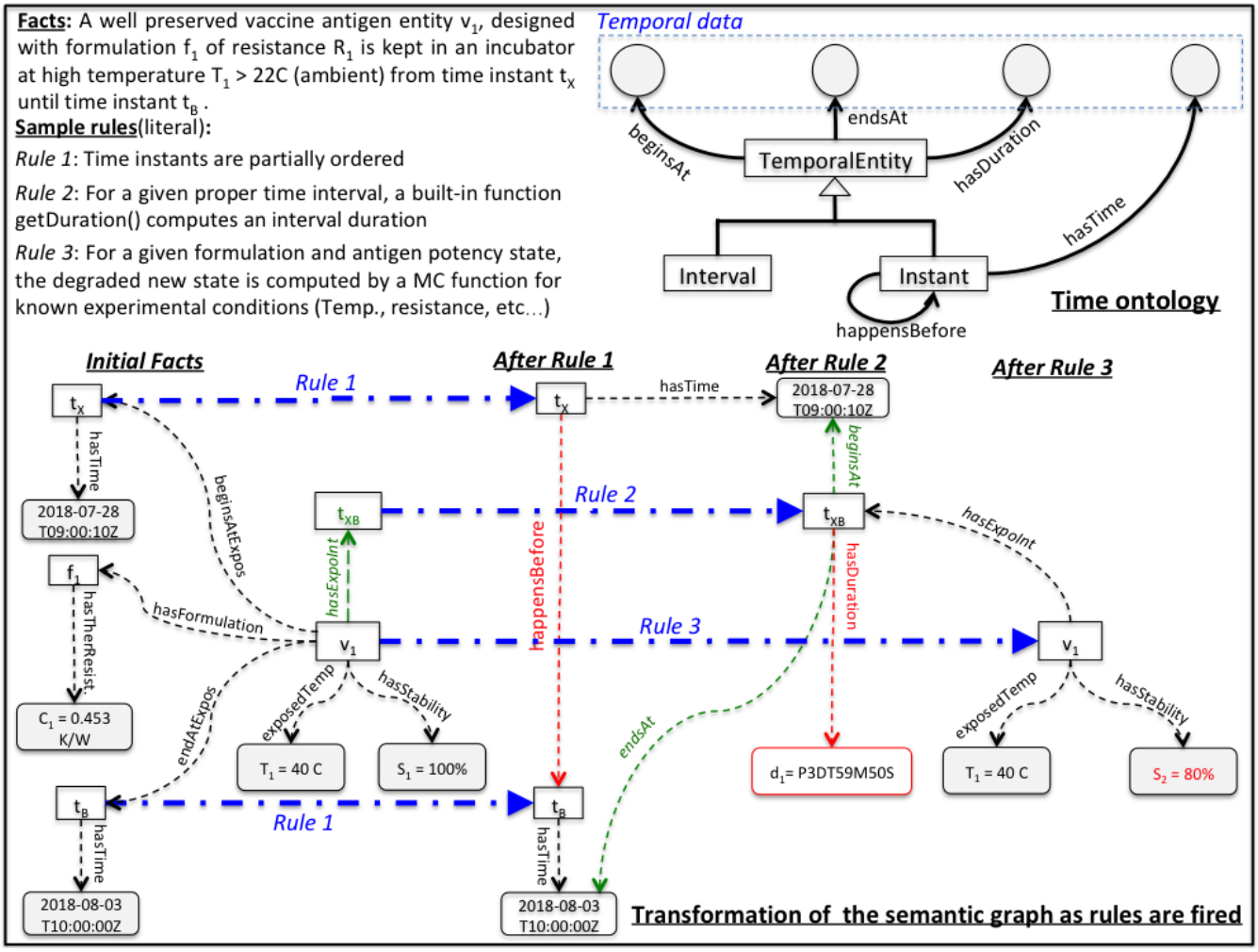


Figure 5. Illustration of rule-based reasoning for vaccine preservation using the temporal domain.

scripting environment are needed. Whistle [19] is 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. It is designed for rapid, high-level solutions to software problems, ease of use, and flexibility in gluing application components together. Computational support is added, enabling the language to handle input and output of model data from/to files in various formats (XML, Java, etc.).

E. Working Example of the Usage of the Framework

We describe in this section a configuration and usage of the framework in a scenario where a researcher investigating formulations at step 2 of the VDLC (see Figure 1) looks ahead in step 6 for an answer to the question: *"What is the expected preservation performance of a Matrix M_1 for a vaccine v_1 currently under study ?"*. To that aim, (s)he must leverage the infrastructure of the framework as pictured in Figure 3, in a step-by step configuration and assembly of the various modules as required by the needs of the study then, perform an analysis of the results. In this case, the query is subject to three simplifying assumptions: (A_1) the matrix M_1 (i.e., mainly stabilizers in the formulation), has been properly characterized and the "degradation resistance" C_{f_1} of the formulation is known, (A_2) there are no significant or unknown biological/chemical phenomenons not captured in the framework, and (A_3) computations and reaction times are negligible. The step-by-step details are as follows.

(i) The researcher prepares the input data (e.g., XML file) of the problem as per the framework predefined DataModel: formulation's unique id ($\mathtt{f}_1$), degradation resistance ($\mathtt{C}_{\mathtt{f}_1}$), expected en-route preservation temperature ($\mathtt{T}_1$), initial known stability level of the formulation ($\mathtt{S}_1$), value of time instants when the exposure to $\mathtt{T}_1$ starts ($\mathtt{t}_\mathtt{X}$), and ends ($\mathtt{t}_\mathtt{B}$).

(ii) The data is loaded into the system and the various ontologies (e.g., Time, BioDomain, etc) are populated with instances, i.e., initial facts, as shown on the left side of Figure 5. These are now statements in individual, separated domain knowledge-based as defined in Section IV-C and in module (L_1, G_1) of Figure 3. Specifically, temporal data (i.e., $\mathtt{t}_\mathtt{X}$ and $\mathtt{t}_\mathtt{B}$) are collected as per the template defined in the DataModel, then are deposited in the time ontology while vaccine and formulation data (e.g., $\mathtt{C}_{\mathtt{f}_1}$, $\mathtt{S}_1$, etc.) populate the BioDomain ontology.

(iii) Rule sets for individual domains and parametrized MC models for bio subdomains (e.g., formulation, vaccine antigen)

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

are provided by the component layer to be used for lower level integration and computation. When and if needed (not in this case), the MC – encoded as built-in function called by the rules engine when the state of the component is needed for inferencing as illustrated in Figure 4 – computes the state of the component and the result is stored in the knowledge base. As indicated in Section IV-B, ontologies and MCs implement foundational theories and system biology/chemistry laws.
(iv) The system integration (layer L_0) is performed by assembling the semantic blocks (ontology + rules + built-ins). The ontologies are integrated and new entity ($\mathtt{t_{XB}}$) and relationship `hasExpoInt` are created to bridge the bio and temporal domains. However, $\mathtt{t_{XB}}$ at this point is a placeholder for a proper time interval in the terminology box (TBox) of the time ontology.
(v) Rule 1 is fired, resulting in the creation of `happensBefore` relationship between $\mathtt{t_X}$ and $\mathtt{t_B}$. This rule belongs solely to the temporal domain and could have been called and executed in step (iii) too.
(vi) Temporal properties `beginsAt` and `endsAt` of $\mathtt{t_{XB}}$ created in (iv) are populated after rule 2 is fired. This is made possible thanks to the fact that the corresponding values are inputs to the built-in function `getDuration()` that computes the duration of temporal intervals (duration of the exposition in this case). The relationship `hasDuration` is created in the temporal domain to store the result of the duration calculated by the built-in function.
(vii) Rule 3 is fired, resulting in the update of the value of property `hasStability` characterizing the stability of the vaccine antigen v_1. For this result to occur, the system level rules engine must pass (via registered built-in) the parameters (i.e.,$\mathtt{C_{f_1}}$, $\mathtt{d_1}$, $\mathtt{S_1}$, $\mathtt{T_1}$) needed by the MC to compute the new state of the system. As in step (iii), the MC model and the ontology are integrated via the built-in function embedded in the rules engine. The new value (S_2) of the property `hasStability` is the answer to the initial question. This, as well as intermediary results, are stored in an output file (e.g., txt format) to be analyzed further by the researcher.

V. Experimental Vaccine Preservation Study

A. Previous Work and Goal of the Study

In [1], we have illustrated the basic implementation and use of our framework in a simplified Oral Polio Vaccine (OPV) formulation under the set of assumptions listed in Section IV-E. An empirical MC model of the degradation of the vaccine stability S_p ($p \in P = \{20, 40, 60, 80, 100\}\%$) – when exposed continuously to temperature T_j for d_k days – was developed with several parameters. The "degradation factor " k_{ijk}^{tf}, which characterizes the ability of the system to maintain itself in a state S_p under the given experimental set up, was found to be given by Equation (1).

$$k_{ijk}^{tf} = \left[\frac{T_{Max} - T_j}{T_{Max} T_j (100 - C_{f_i})}\right]^{\frac{d_k}{d_{Max}}} \quad (1)$$

where T_{max} is the maximum allowable exposure temperature for the experiment and $C_{f_i} \in (0, 100)$ is the "degradation resistance" of a given formulation. Also, the transitions between states (S_p) were computed as per Equation (2).

$$a_{ijk}^{tf}|_{p,q} = (1 - \Delta_{ijk}^{tf}|_{p,q}) k_a k_{ijk}^{tf} \quad (2)$$

where $\Delta_{ijk}^{tf}|_{p,q}$ is the gap of virulence between a state of stability p and one of stability $q < p$ in P and, $k_a > 0$ is a balancing coefficient allowing the probabilities to sum to 1 as per MC modeling rules.

As pointed out in Section II-B, vaccine and vaccination are complex systems and processes not fully understood yet. The current state of vaccine research and development practices does no provide means to characterize key MC model parameters (e.g., C_{f_1}) or ensure that all relevant phenomenons and interactions are captured in models of system biology at the chosen level of abstraction of the representation. Thus, assumptions (A_1) and (A_2) can hardly sustain real-world applications of the framework. Much needed detailed study (outside the scope of this work) is required to address those challenges. Until that becomes a reality, we will develop and use deterministic models to support the computation and predict of the degradation of the vaccine with the lowest possible margins of error. Such models of system biology must be amenable to a smooth integration with the semantic ones in the framework for usage in real-world applications across the VDLC. Therefore, we will conduct laboratory experiments to satisfy those needs.

B. Overview of the Study and Hypotheses

Vaccine antigens are mostly protein. Thus, we use a protein (enzyme) Horse Radish Peroxidase (HRP) as a vaccine model for the preservation study. To perform the preservation studies we use a commercially available HRP. This 40 kDa protein is similar in size to the popular vaccine mimic ovalbumin. The unique structural features of HRP make it a good model protein for analyzing the influence of various excipient properties on protein stability. Because any conformational or structural perturbations of HRP during storage loss of protein activity this is an excellent candidate to study protein stability. The protein also contains four disulfide bonds and numerous metal-binding sites that attract two divalent calcium ions to bind to the protein as enzymatic cofactors. HRP protein is a metalloenzyme that has a noncovalently bound to a heme prosthetic group at the active site. This allows the protein to catalyze the reduction of hydrogen peroxide to water.

C. Laboratory Experiments Setup and Data Collection

The stability of HRP would be tested in three different temperatures at 22^0C, 30^0C and 37^0C in three different excipient formulations containing varying percentages (1-10 w/v%) of a well established excipients used in commercial vaccine formulations. Formulations were also constituted with a constant amount of preservative neomycin (0.01 w/v%), and adjuvant alum (0.02 w/v%) and the dispersant used was phosphate buffer saline (PBS, 0.25 mM) at pH 7.4. Formulations are as follows : (1) F_1 - 1% $MgCl_2$, neomycin (0.01%) and alum (0.02%); (2) F_2 - 5% $sucrose$, neomycin (0.01%) and alum (0.02%); (3) F_3 - 2.5% $trehalose$, neomycin (0.01%) and alum (0.02%). The amount of HRP added to each formulation (F_1-F_3) was 1.33 μg. The stability of the HRP protein at different temperatures in different formulations was tracked using an analytical fluorimetric redox based assay – Ample Red. The stability of the protein HRP was monitored at regular intervals by a redox based fluorimetric assay. A control formulation of HRP formulated in buffer PBS without excipients was

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

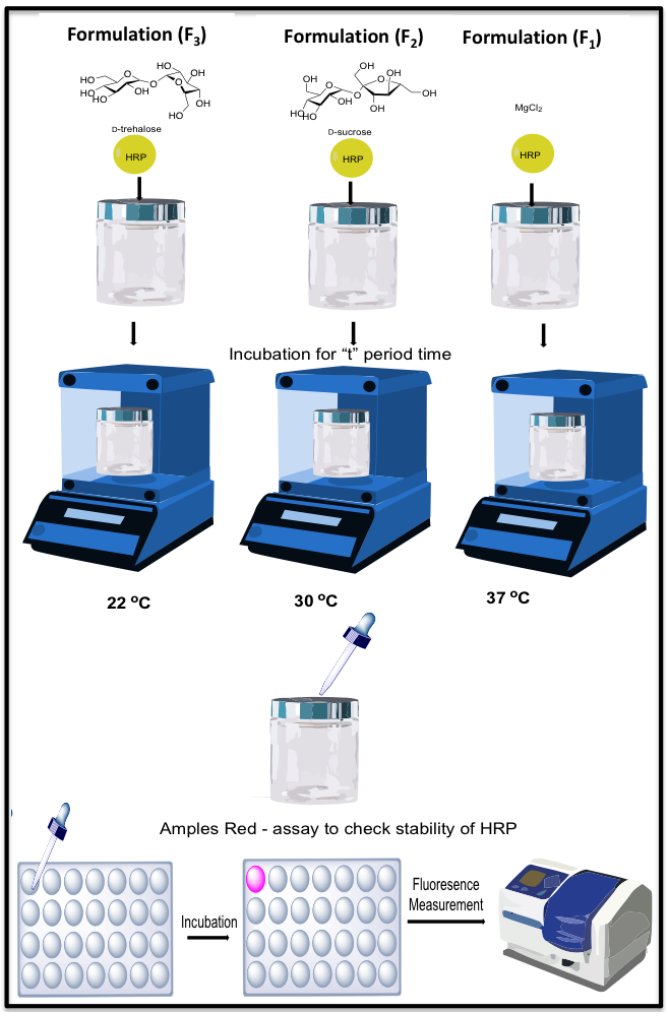


Figure 6. Analysis of the stability of HRP protein in three formulations

kept at -20^0C (FC). Formulations (F_1-F_3) were sealed in 20 mL borosilicate vials kept at the three temperatures mentioned above (22^0C, 30^0C and 37^0C). At regular intervals 50 μL of each formulation with HRP was taken out of each vial added to 50 μL (0.2 mM) of assay reagent Amplex Red in a micro-centrifuge tube. After thoroughly mixing 50 μL was aliquoted and added to 50 μL (4 μM) aliquot of H_2O_2 in a 96 well Costar clear polystyrene plate. The plate was incubated for 1 hour under dark and fluorescence emission was obtained at 590 nm after excitation at 530 nm using a Molecular Devices (M_2) plate reader. The emission intensity of each formulation (F_1-F_3) was recorded and compared to the control (FC) at -20^0C and percent degradation of HRP was calculated for F_1-F_3 at the three different temperatures 22^0C, 30^0C and 37^0C.

D. Experiment Results, Analysis and Limitations

A wide variety of protein stabilizing excipients are used in vaccine development for enhancing the stability of vaccine protein antigens and they are referred to as stabilizing excipients. These excipients have been reported to stabilize the structure of native proteins at moderate (1 w/v%) to high concentrations (30 w/v%). Carbohydrates excipients (i.e., sucrose and trehalose) and polyols (i.e., mannitol, sorbitol) are often used to stabilize protein antigens and protect them from aggregation during lyophilization. Carbohydrates are known to be highly effective in increasing the melting temperature (Tm) of proteins, preventing them from denaturing. Among sugars, sucrose and trehalose have been the most frequently used in thermostabilization. Even though HRP stability data (see Figure 7), at a glance looks erratic, a general trend can be perceived that at higher temperatures (30^0C and 37^0C), $trehalose$ and $MgCl_2$ fail to stabilize the protein HRP. But looking at the stability data of HRP with sucrose (see Figure 7(b)), it is apparent that sucrose could stabilize HRP even at elevated temperatures. Carbohydrates like sucrose and trehalose have high glass transition temperatures (Tg) are known to be more effective in thermostabilizing proteins than other excipients. Salts (i.e., $MgCl_2$) affect in widely different manner when stabilizing proteins. For example, at low concentrations, they could stabilize proteins through non-specific electrostatic interactions while at high concentrations, salting in or salting out would occur. Salting in would preferentially stabilize the protein while salting out would destabilize the protein. At low concentration the hydrated forms of the divalent cation Mg^{2+} has been known to bind to the peptide units through stabilizing hydrogen bonds. Looking at the stability data of HRP protein with $MgCl_2$ compared to the carbohydrate $sucrose$, $MgCl_2$ has provided very little stability.

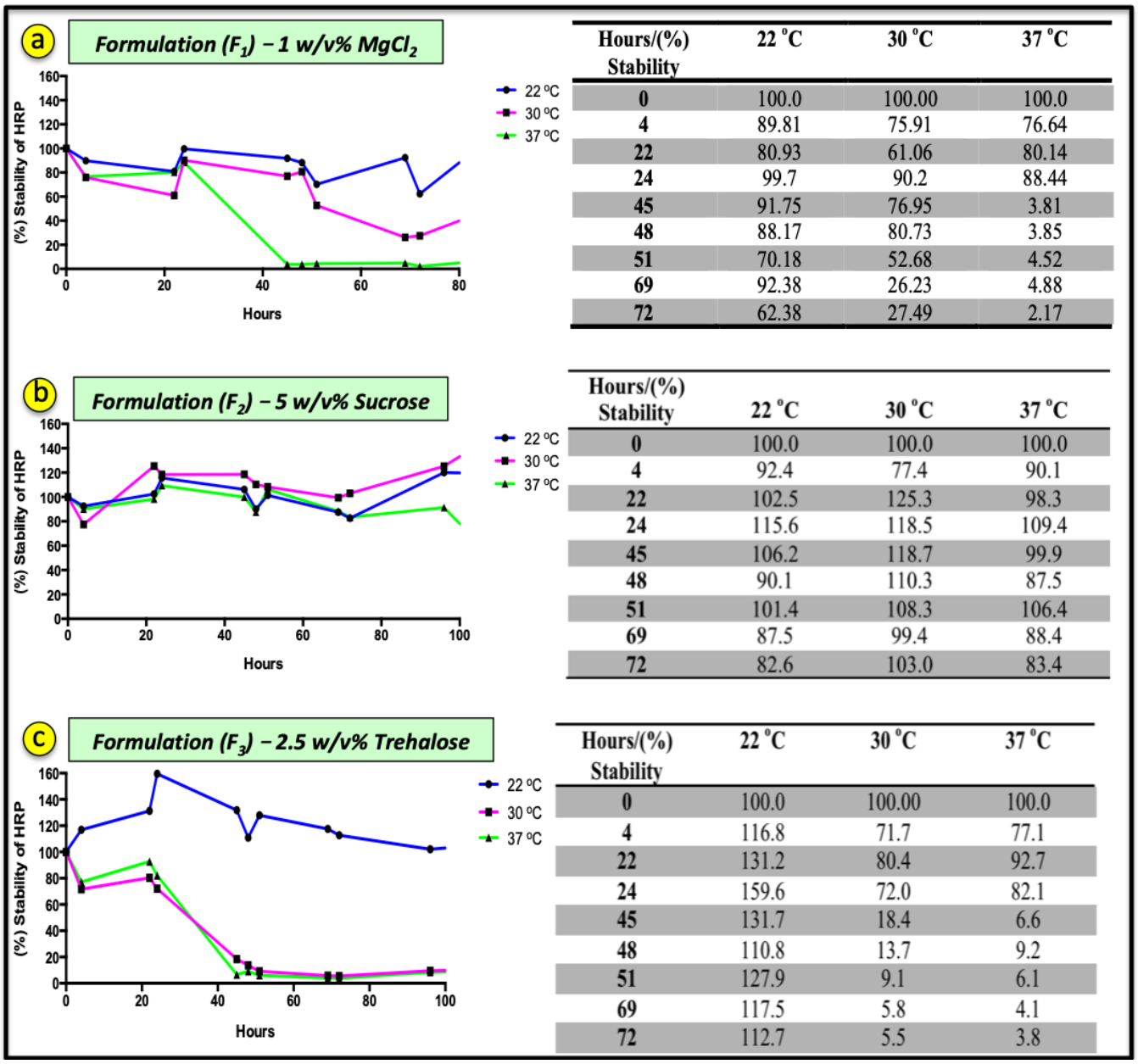
E. Regression Analysis: Procedure and Results

In the absence of means to properly identify parameters characterizing individual formulations as needed by the MC stochastic model described in Equations (1) and (2), we seek to develop indirect means for predicting the degradation of a given formulation. Thus, we formulate and construct regression models correlating (statistically) independent experiment variables (introduced in Section V-C) and the percentage of degradation of the protein (i.e., our surrogate vaccine antigen) as response. The variables considered for this analysis are primary the temperature at which the formulated protein is exposed to (x_2) and the duration of exposition at that temperature (x_1) for formulations $MgCl_2$ and $sucrose$. In the case of $trehalose$, a third variable – the percentage of stabilizer (x_3) in the formulation solution – is added to the mix. We use the data collected in Section V-C to perform the analysis. Simple and multiple regression models accounting for the variables individually or together and their interactions are constructed and identified using the following nomenclature.

$$M_l^{j.k} \models Y = f(x_1, x_2, x_3) \quad (3)$$

where $M \in \{L, Q, P\}$ is the regression model, i.e., Linear(L), Quadratic(Q) or Polynomial(P) of order 3 for the formulation under study; $j \in \{1, 2, 3\}$ is the type of the formulation, i.e., $MgCl_2(1)$, $sucrose$ (2) and $trehalose(3)$; $k \in \{1, 2, 3, 4, 5, 6\}$ is the percentage of the stabilizer in the formulation of interest, i.e., 1%(1), 2.5%(2), 5%(3), 10%(4), 20%(5), and a combination of several percentages(6). Also, $l \in \{1, 2, 3, 4\}$ is the temperature at which the protein is exposed, i.e., average room temperature of $22^0C(1)$, high temperature of $30^0C(2)$, body temperature of $37^0C(3)$ and All temperatures(4). Even though this representation allows us to cover all configurations of regressions, we will be focusing on ones enabling us to capture, represent and identify multiple regressions in a very unique ways. Thus, $l = 4$ in such models.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



Hours/(%) Stability	22 °C	30 °C	37 °C
0	100.0	100.00	100.0
4	89.81	75.91	76.64
22	80.93	61.06	80.14
24	99.7	90.2	88.44
45	91.75	76.95	3.81
48	88.17	80.73	3.85
51	70.18	52.68	4.52
69	92.38	26.23	4.88
72	62.38	27.49	2.17

Hours/(%) Stability	22 °C	30 °C	37 °C
0	100.0	100.0	100.0
4	92.4	77.4	90.1
22	102.5	125.3	98.3
24	115.6	118.5	109.4
45	106.2	118.7	99.9
48	90.1	110.3	87.5
51	101.4	108.3	106.4
69	87.5	99.4	88.4
72	82.6	103.0	83.4

Hours/(%) Stability	22 °C	30 °C	37 °C
0	100.0	100.00	100.0
4	116.8	71.7	77.1
22	131.2	80.4	92.7
24	159.6	72.0	82.1
45	131.7	18.4	6.6
48	110.8	13.7	9.2
51	127.9	9.1	6.1
69	117.5	5.8	4.1
72	112.7	5.5	3.8

Figure 7. Stability of HRP at temperatures 22^0C, 30^0C and 37^0C (a) in Formulation F_1 - 1% $MgCl_2$, (b) in Formulation F_2 5% $sucrose$ and, (c) in Formulation F_3 2.5% $trehalose$ for a period of 72 hours

We follow a rigorous data analysis process to generate and ensure the quality of the resulting regression models for the each of the formulations. First, the data is cleaned from outliers using 95% confidence interval for the response. Second, the data is checked for confounding to make sure the independent variables are independent from one another. This is done by plotting scatter plot for pairs of independent variables and checking for collinearity. In this study, no such relationship were found in any of our formulation data sets. With those foundations in place, the next move is to fit response surface models to individual formulation stability percentage (%HRP), as a function of controllable factors x_i, ($i \in \{1, 2, 3\}$) as defined above. One model of each of the three types, i.e., Linear(L), Quadratic(Q) or Polynomial(P) is created. Plots of residuals versus fitted values are generated and used each time to check for violation of assumptions for error in regression models. This exercise has helped us uncover inconsistencies in the regression models for the $trehalose$ formulation. However, applying a logarithmic transformation to the data has resulted in more normal (randomly distributed points) residual plots, and better regression models. Finally, the best model for each of the formulation was selected by comparing models' coefficient of determinations or goodness-of-fit statistic (i.e., R^2). Table I summarizes our findings.

Polynomial models' coefficients of determination (R^2) are the highest of all model types for all formulations. Thus, they are the best fitted models for the response as shown in Table I. We note here that the response function for the $trehalose$ is

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. Selected multiple regression models for each candidate formulation. The % of trehalose in model $P_4^{3.6}$ comprises 2.5%(2), 10%(4) and 20%(5)

Formulation	Model	Coef. det.(R^2)	Response function: $Y = f(x_1, x_2, x_3)$
$MgCl_2$	$P_4^{1.1}$	82.91 %	$Y = 96 + 1.90x_1 - 0.8x_2 - 0.0499x_1^2 + 0.021x_2^2 + 0.015x_1x_2 + 0.000249x_1^3 + 0.00066x_1^2x_2 - 0.00244x_1x_2^2$
$sucrose$	$P_4^{2.3}$	72.36 %	$Y = -12.9 + 0.46x_1 + 7.88x_2 - 0.0299x_1^2 - 0.135x_2^2 + 0.118x_1x_2 + 0.000115x_1^3 + 0.000206x_1^2x_2 - 0.00219x_1x_2^2$
$trehalose$	$P_4^{3.6}$	88.94 %	$ln(Y) = 9.39 + 0.4388x_1 - 0.337x_2 - 0.254x_3 - 0.003164x_1^2 + 0.00582x_2^2 + 0.01464x_3^2 - 0.02049x_1x_2 - 0.00820x_1x_3 + 0.0074x_2x_3 + 0.000017x_1^3 + 0.000036x_1^2x_2 + 0.000032x_1^2x_3 + 0.000230x_1x_2^2 + 0.000162x_1x_2x_3 + 0.000018x_1x_3^2 - 0.000000x_2^2x_3 - 0.000453x_2x_3^2$

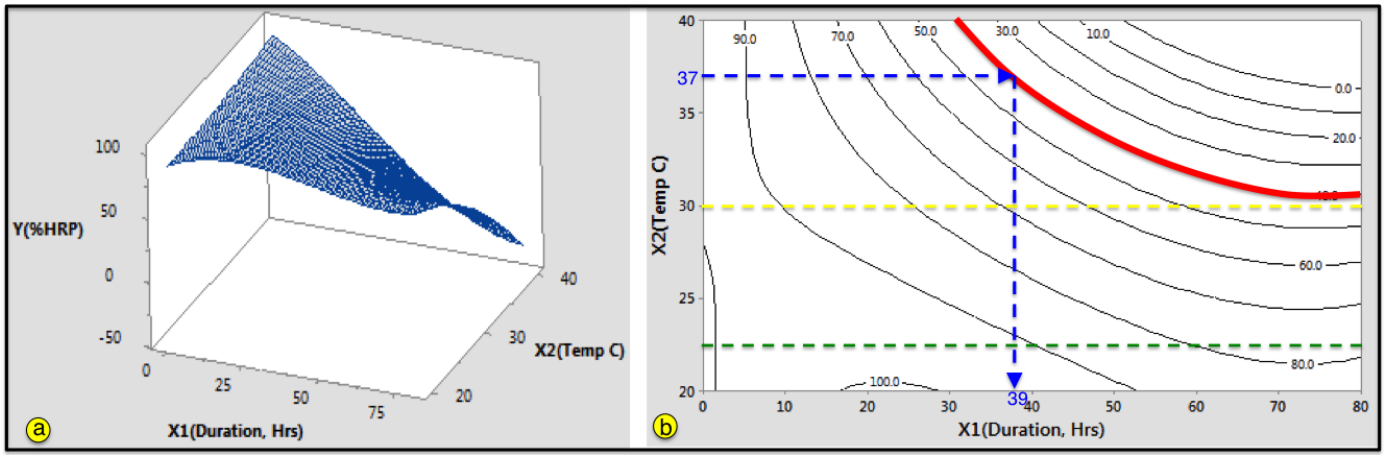


Figure 8. Response surface (a) and contour (b) plots for $MgCl_2$ based on multiple regression model $P_4^{1.1}$

logarithmic thanks to the data transformation and the presence of three – instead of two – independent variables. A plot of the response surface for model $P_4^{1.1}$ ($MgCl_2$ formulation) is shown in Figure 8(a). The hyperplane representing the response, i.e., the percentage of HRP is bended downward as both the duration of exposition and temperature increase. This is consistent with the expected behavior of the system but also as previously found with the empirical model. However, we gain additional insight in the observation that, for this particular formulation, its stability clearly takes a dive as temperature at which the formulated antigen is exposed increases. This suggests that temperature (i.e., intensity of the heat), more than duration of exposure, is the main driver of the breakdown of the stabilization property of the formulated protein. Figure 8(b) is the contour plot for the same response surface. It offers a better view of the devastating effect of temperature on the stability of the formulation. For instance, considering a targeted minimum threshold of 40% stability (as for the Oral Polio Vaccine), one can clearly see that it will take less than 40 hrs (1.6 day) of exposure of the formulated antigen at body temperature (i.e., $x_1 = 37^0C$) to loose 60% of its stability. For the same duration, it looses only 30% of its stability at $x_2 = 30^0C$ and less than 10% stability loss at room temperature ($x_3 = 22^0C$). These results also explain well-established state-of-the-art vaccine preservation practices of keeping vaccines at lower temperatures (i.e., higher logistic costs) to maintain its stability over a long period of time. Finding the formulations that can achieve the same results or better at higher temperatures (i.e., lower logistic costs) remains the holy grail of vaccine preservation research.

VI. Discussions

As of now, it is difficult for real-world applications to sustain assumptions (A_1) and (A_2) stated in the working example of the framework described in Section IV-E. When it comes to assumption (A_1), the case study highlights the challenge of developing accurate and precise system biology models to be integrated with semantic models in our framework as described in Section III and pictured in Figure 3. This is needed to support prediction and reasoning in the framework. In the face of challenges regarding the characterization of formulations to support the full definition of stochastic models (MC), we have developed regression-based models for stability prediction in our prototype implementation on a preservation study. Such models can be used under specific conditions – in lieu of actual MC models – for build-in functions enabling computations such as the ones in rule 3 (see Figure 5). Regression models establish statistical (not causal) correlations between independent variable(s) and a response under specific and well-defined set of conditions. This limits its scope of use and its ability to support the explanation of underlining biological/chemical phenomenon. This will not be resolved until proven and full characterizations of biological agents (e.g., vaccine antigens) are available to be used for models (such as the MC models) used in this framework.

Inconsistencies and out of range results in selected regression models for $sucrose$ ($P_4^{2.2}$) and $trehalose$ ($P_4^{3.6}$) suggests that there are important underlining chemical/biological phenomenons unaccounted for by the model. This is a translation of a clear violation of assumption (A_2). Addressing this problem will require uncovering such phenomenons for the given formulation followed by the identification of explanatory variables to be tracked during experiments and

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

refinement of current regression models to account for the new variable(s). The complexity of the problem significantly increases if we consider that, to date, there are 380 established antigen stabilizing compounds or generally-regarded-as-safe (GRAS) excipients candidates that could possibly be used in a vaccine formulation [20].

The ability of the framework to survive assumption (A_3) in real-world applications depends on the capability of the underlining software infrastructure supporting implementation. Even though one running VDLC-related applications using the framework would not want them to last for ever, they are not safety-critical. Thus, real-time computations are not a "must" but, fast computations – especially when faced with large volume and heterogeneous data – are needed. As pointed out in Section III-B, OWL – the language we use to develop semantic models in this framework – enables both human and machine processing of vaccine and domains knowledge over the World Wide Web (WWW). Proper integration with databases, web-based interfaces, and cloud computing as well as with the appropriate configurations, fast, integrated yet distributed solutions are possible. Therefore, both batch and streaming-based processing of data through the framework are possible.

VII. Conclusion and Future Work

In this paper, we have introduced and described a knowledge-intensive framework for behavior specification, modeling and reasoning for efficient vaccine systems engineering. This research is motivated by limitations of state-of-the-art vaccine development approaches in capturing, representing and reconciling domains and disciplines knowledge and viewpoints across the vaccine development life cycle in an effective manner. The inherent highly stochastic behavioral nature of biological elements such as vaccines coupled with knowledge disconnect between stakeholders (e.g., chemists, biologists, clinicians, the public, big pharma, etc.) with sometimes conflicting interests add to the numerous technological challenges of engineering such complex biological systems. This leads to long, complex, and costly efforts with high failure rates currently observed in vaccine development initiatives. Also, potency of successful vaccines is difficult to predict and very expensive to preserve in the face of changing and challenging environmental conditions as well as limited resources.

The knowledge-intensive framework is shown to be a possible solution to successful vaccine systems engineering moving forward. Description logic semantics provide the necessary formalisms needed to capture, represent and reason about vaccine and foundational domains knowledge in a clear, unambiguous way with respect to the associated domain and the context of use while enabling automated processing and reasoning by machines through semantic web technologies. Library of reusable semantic components – i.e, semantic blocks comprising an ontology, rules, computation and communication interfaces – encapsulate knowledge in a formal and well-defined manner. They are integrated to stochastic models of vaccine system biology – Markov Chains (MC) – of the underlining unfolding biological and/or chemical processes at various level of abstractions throughout the development lifecycle of the vaccine, when needed. This layered and modular structure enables flexibility in the assembly – via integration – of models of various level of complexity and types in support of the investigation of research issues that cut across domains in the vaccine development lifecycle. Thus, this will help bridge the gap between domains and stakeholders along the development lifecycle, with the ripple effect of shortening the development cost, length and complexity.

A step-by-step implementation procedure coupled with a prototype vaccine preservation study have shined some light in the implementation of the framework. Such studies, if successful, can replicate actual preservation conditions in extreme weather (e.g., subsaharan Africa) and guide the design and selection of the most effective formulation able to stabilize the vaccine in given situations. However, limitations in the current state of vaccine research and development practices in, (1) providing means to characterize key MC model parameters and, (2) ensuring that all relevant phenomenons and interactions are properly have appeared to be a challenge to the proper account of system have stood on the way. Thus, we have designed and conducted laboratory experiments, which coupled with regression analysis of stability data has resulted into multiple regression models that were used as an alternative path. The resulting deterministic models are shown to provide statistically significant and satisfactory results under the specific set of experimental conditions. Responses surfaces and contour for on the "on-the-fly" prediction have been produced.

Future work needs to address challenges related to the full and accurate characterization of vaccines properties standing on the way of the creation of stochastic models of biological elements (e.g., antigens) to be used in the framework. Therefore, advanced laboratory experiments are needed to that aim but also to uncover and understand relevant phenomenons of interest contributing to the system response (e.g., stability of the vaccine). The application of the appropriate design of experiments will be needed to ensure cost effectiveness and overall efficiency in studies and analyses. Bringing the benefits of the framework introduced in this work to day-to-day work of stakeholders across the vaccine development lifecycle will also necessitates further work on the refinement and validation of the framework for various vaccine types, analyses and cross-cutting concerns (e.g., potency, preservation, safety) and various environmental conditions. Finally, the collaborative development of domain and discipline knowledge across the development lifecycle – e.g., vaccine ontology as in [21] – is highly suitable to foster dialogue and synergy between stakeholders.

Acknowledgment

The authors would like to thank the students who contributed to this research with data collection and analysis: Kavini Rathnayake, Unnati Patel, Veera Venkata Naga Manohar Devarasetty, Melinda Mustain, James Johnson and Anoop Kumar Reddy Gudipati.

References

[1] L. Petnga and S. Jayawardena, "Knowledge-driven Vaccine Systems Engineering," The Thirteenth International Conference on Systems (ICONS), Athens, Greece, April 22 - 26, 2018.

[2] H. Stevens and K. Debackere, "Vaccines: Accelerating Innovation and Access," *Global Challenges report World Intellectual Property Organization(WIPO)*, 2016.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[3] A. Loharikar, L. Dumolard, S. Chu, T. Hyde, T. Goodman, and C. Mantel, "Status of New Vaccine Introduction Worldwide, September 2016," 65(41);11361140, US. Center for Disease Control(CDC), October, 21, 2016.

[4] World Health Organization(WHO), "Immunization coverage," WHO Factsheets/Detail, July, 18, 2013.

[5] International Federation of Pharmaceutical Manufacturers & Associations(IFPMA), "Vaccine research and development," IFPMA Resources/Graphics, April, 08, 2013.

[6] The United Nation Children's Fund (UNICEF), "Vaccines: Handled with Care," Division of Communication, New York, USA , 2004.

[7] Pharmaceutical research and Manufacturing of America(PhRMA), "Vaccines Factbook," PhRMA, 2013.

[8] Compound Interest, "A Summary of Common Vaccine Components," Accessible at: https://www.compoundchem.com/2015/02/10/vaccines/; Retrieved: November 17, 2018.

[9] J.F. Sowa and A. Borgida, "Principles of semantic networks: explorations in the representation of knowledge," John F. Sowa (Ed.), 1991.

[10] P.J. Hayes, "The logic of frames," Frame Conceptions and Text Understanding, deGruyter, pp. 4661, Berlin, 1980.

[11] M. Pavlic, A. Mestrovic, and A. Jakupovic, "Graph-based formalisms for knowledge representation," 17th World Multi-Conference on Systemics, Cybernetics and Informatics, July 912, Orlando, Florida, USA, 2013.

[12] F. Baader, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, "The Description Logic Handbook: Theory, implementation, and applications," Cambridge, 2003.

[13] K. Schild, "Terminological cycles and the propositional with mu-calculus," J. Doyle, E. Sandewall, P. Torasso (Eds.), 4th Int. Conference on the Principle of Knowledge Representation and Reasoning (KR-94), pp. 509520, 1994.

[14] L. Petnga and M. A. Austin, "An Ontological Framework for Knowledge Modeling and Decision Support in Cyber-Physical Systems," Advanced Engineering Informatics, Vol. 30, No. 1, pp. 77-94, January, 2016.

[15] F. Baader, I. Horrocks, and U. Sattler, "Description Logics," In: Handbook of Knowledge Representation, pp. 135 - 180, Elsevier (US), 2008.

[16] B. Ingalls, "Mathematical Modeling in Systems Biology: An Introduction," Applied Mathematics, University of Waterloo, CA, 2012.

[17] C. J. Tomlin and J. D. Axelrod, "Biology by numbers: mathematical modeling in developmental biology," Nature Reviews; Genetics 8: pp. 331 - 340, 2007.

[18] M. Mosteller, M. A. Austin, R. Ghodssi, and S. Yang "Platforms for engineering experimental biomedical systems," 22th Annual INCOSE International Symposium (IS 2012), Rome, Italy, July 9-12, 2012, 2012.

[19] 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, pp. 223238 (EU), 2014.

[20] U.S. Food and Drug Administration (FDA), "Generally Recognized As Safe (GRAS) food substances," Accessible at: https://www.accessdata.fda.gov/scripts/fdcc/?set=SCOGS; Retrieved: November 17, 2018.

[21] University of Michigan School of Medicine, "Vaccine Ontology," Accessible at : http://www.violinet.org/vaccineontology/introduction.php; Retrieved: November 17, 2018.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The Application of a Radial Basis Function Network to Supervised Terrain Classification

[1]Tiny du Toit and [2]Hennie Kruger
School of Computer Science and Information Systems
North-West University
Potchefstroom, South Africa
e-mail: [1]Tiny.DuToit@nwu.ac.za, [2]Hennie.Kruger@nwu.ac.za

***Abstract*—In this paper, inertial contact sensor-based terrain classification is performed with a Radial basis function network. Compared to the more popular Multilayer perceptrons, Radial basis function networks are also intelligent techniques and universal approximators, but with a much simpler structure and shorter training time. It has been shown that Radial basis function networks are efficient classifiers and, consequently may be used for terrain classification. For the experiments, a mobile robot platform recorded vibration training data with an inertial measurement unit while traversing five different terrains: asphalt, carpet, dirt, paving, and tiles. The composition of these terrains induces specific vibrations in the mobile platform, which are measured by the inertial measurement unit. The vibration signatures comprise the mobile robot's linear acceleration, orientation, and the earth's magnetic field. In contrast to most terrain classification techniques found in literature, no pre-processing of the data is performed. This reduces the computational overhead needed for real-time classification. A Radial basis function network is then trained using a hybrid conjugate gradient descent method and *k*-fold cross-validation. Identification of the terrain is performed in real time. The classification capability is empirically compared to that obtained by a Multilayer perceptron, a Naïve Bayes method and a Support Vector Machine, which have also been successfully applied to terrain classification in literature. It was found that the Radial basis function network outperformed the Support Vector Machine and Naïve Bayes techniques by a relatively large margin. The Multilayer perceptron, although performing slightly better than the Radial basis function network, has some disadvantages compared to the Radial basis function network. Consequently, the Radial basis function network, with no pre-processing of the input data, may be used successfully as an alternative contact sensor-based terrain classification method.**

Keywords–classification; inertial measurement unit; MLP; RBFN; sensor; terrain classification.

I. INTRODUCTION

Mobile robots are employed on various types of terrain [1] in many different operational fields, such as supply and logistics, surveillance, search and rescue missions, agricultural applications, transportation, cleaning, inspection and entertainment [2][3]. For these operations, it may be necessary to traverse some indoor or off-road terrain which might influence the vehicle's performance. The efficiency of these vehicles can be improved by their detection of their environment. This act of identifying the type of terrain being traversed from a list of candidate terrains such as dirt, sand, or gravel, is called terrain classification [4].

Factors, such as friction, cohesion, damping, stiffness and surface irregularity comprise the terrain interface that is presented to the mobile robot [5]. It may be beneficial to identify the current terrain type as the terrain conditions may have an influence on both the planning stages and motion control of the vehicle's trip. Once the mobile robot's control system has knowledge of the surface on which it is travelling, it will be easier to maneuver over uneven terrain or around obstacles, which allows the vehicle to traverse the terrain most effectively. In particular, awareness of the terrain type will enable the vehicle to drive at higher speeds, enable the mobile robot to choose an appropriate driving mode, prevent physical damage, keep wheels from sinking into the ground and obtain an automated driving process which is terrain-dependent.

Research on the identification of terrain types can be divided into two groups: methods relying on noncontact sensors [4] - [8] and methods utilizing contact sensors [9] - [12]. Examples of noncontact sensors are vision sensors and laser scanners. A vision sensor, such as a charge-coupled device (CCD) camera, uses techniques that extract textures and colors from the sensor data to classify this information into variable terrains, like forests and the sky. Unfortunately, the performance of these techniques is highly dependent on environmental factors, such as lighting conditions and climate effects and consequently, the sensor information can be distorted. Laser scanner sensor data that are obtained from a terrain are converted into frequency information. Learning algorithms then use this information to classify the terrain. A disadvantage of such a method is that it needs numerous data points which may hinder real-time classification. As the mobile robot traverses the specific terrain, these terrain properties combined with the robot dynamics produce vibrational signatures in body motion. Methods based on contact sensors, however, classify a terrain using sensor information, such as the vibration frequency or the slope ratio of the mobile robot's body into the terrain type.

The aim of this paper is to perform terrain classification using a Radial basis function network (RBFN) and then to compare the results to a Multilayer perceptron (MLP) neural network [13], the Naïve Bayes method and the Support Vector Machine (SVM) technique, which have also been

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

successfully applied to this problem to provide context. The main focus, however, is on the comparison between the RBFN and the MLP and, consequently only these two methods will be discussed in detail.

Broomhead and Lowe [14] proposed the RBFN in 1988. This type of neural network model forms a unifying link among many different research fields, such as pattern recognition, regularization, function approximation, noisy interpolation, and medicine. The model has become increasingly popular due to its topological structure and neurons that are tuned locally. In addition, it has become a good alternative to the MLP, since it has capabilities equivalent to those of the MLP model, but with a simpler structure and can be trained much faster. Previous studies have shown that RBFNs in general are efficient classifiers [2][15]. In one study in particular [2], a RBF network has been used for terrain classification where a Discrete Fourier transform was implemented to perform feature extraction. Unfortunately, such pre-processing of the data is a time-consuming task, which may prevent the real-time identification of the terrain.

The MLPs that are trained by the backpropagation rule is one of the most important neural network techniques used for nonlinear modeling [16]. Their greatest benefit is that no *a priori* knowledge of the particular functional form is required. Feedforward MLPs are mostly utilized to estimate relationships between input and target variables. They often exhibit superior performance in comparison to more classical methods. In contrast to common belief, they are not a black box tool. The scientific understanding of empirical phenomena subject to neural network modeling can be considerably enhanced. Formal statistical inference can be performed using estimates obtained from neural network learning as the basis. Statistical tests of specific scientific hypotheses that are of interest become possible. The capability of MLPs to extract interactive and complex nonlinear effects extends the power of such tests beyond those possible with more traditional methods, such as linear regression analysis.

Terrain classification will be performed based on real-time vibration data obtained from an inertial measurement unit (IMU) contact sensor. No pre-processing of the data as reported in some previous studies is performed. The assumption is that the output of the IMU sensor is influenced by the vibrations induced in the platform while traversing different terrains. The test vehicle, a Lego Mindstorms EV3 mobile robot, is augmented by an IMU mounted on a Raspberry Pi 2 computer. Data that is collected from the IMU on the moving test vehicle is used as the terrain signature. This signature will then be classified as one of five predetermined terrains - asphalt, carpet, dirt, paving, or tiles.

The remainder of the paper is organized as follows. In Section II, the relatively simple structure and training of the RBFN will be discussed. A variant of the gradient descent method is used for training. The well-known MLP architecture and backpropagation training algorithm are considered in Section III. Specific issues related to artificial neural network model building are examined in Section IV. Experiments performed to determine the accuracy of terrain classification using a RBFN, an MLP, the Naïve Bayes method and an SVM model will be considered in Section V. The results that were obtained will be examined in Section VI. Finally, some concluding remarks and future work will be presented in Section VII.

II. RADIAL BASIS FUNCTION NETWORKS

In this section, the RBFN architecture and training of the model will be considered.

A. Architecture

A RBFN is a feedforward neural network with three layers $(J_1 - J_2 - J_3)$ [15][17][18] as shown in Figure 1. In the input, hidden and output layers there are J_1, J_2 and J_3 neurons, respectively. The bias in the output layer is denoted by $\phi_0(\vec{x}) = 1$ while the nonlinearity at the hidden nodes is denoted by the $\phi_k(\vec{x})$'s. Each hidden layer node uses a Radial basis function (RBF), denoted by $\phi(r)$ for its nonlinear activation function. The hidden layer performs a nonlinear transformation of the input. This nonlinearity is then mapped into a new space by the output layer which acts as a linear combiner. Normally, all hidden nodes utilize the same RBF; the RBF nodes have the nonlinearity $\phi_k(\vec{x}) = \phi(\vec{x} - \vec{c}_k), k = 1, \dots, J_2$, where $\vec{c}_k$ denotes the center or prototype of the *k*th node and $\phi(\vec{x})$ is an RBF. An extra neuron in the hidden layer can model the biases of the output layer neurons. This neuron has a constant activation function $\phi_0(r) = 1$. The RBFN determines a global optimal solution for the adjustable weights in the minimum mean square error (MSE) sense by using the method of linear optimization. The output of the RBF network, provided by input $\vec{x}$, is given by

$$y_i(\vec{x}) = \sum_{k=1}^{J_2} w_{ki}\phi(\|\vec{x} - \vec{c}_k\|), i = 1, \dots, J_3, \qquad (1)$$

where $y_i(\vec{x})$ is the *i*th output, w_{ki} denotes the connection weight from the *k*th hidden neuron to the *i*th output unit, and $\|\cdot\|$ is the Euclidian norm. The RBF usually utilizes the Gaussian function $\phi(\cdot)$ and such a model is normally called the Gaussian RBF network.

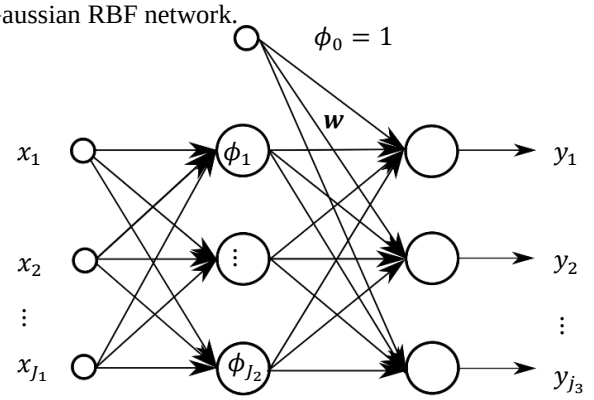


Figure 1. RBF network architecture [16]

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Given a set of N pattern pairs $\{(\vec{x}_p, \vec{y}_p)|p = 1, \dots, N\}$, (1) can be expressed in matrix form as

$$\boldsymbol{Y} = W^T \Phi \tag{2}$$

where $\boldsymbol{W} = [w_1, \dots, w_{J_3}]$ is a $J_2 \times J_3$ matrix, $\vec{w}_i = (w_{1i}, \dots, w_{J_2 i})^T$, $\Phi = [\vec{\phi}_1, \dots, \vec{\phi}_N]$ is a $J_2 \times N$ matrix, $\vec{\phi}_p = (\phi_{p,1}, \dots, \phi_{p,J_2})^T$ is the hidden layer output for the pth sample, specifically, $\phi_{p,k} = \phi(\|\vec{x}_p - \vec{c}_k\|)$, $\boldsymbol{Y} = [y_1\ y_2\ \dots y_N]$ is a $J_3 \times N$ matrix, and $\vec{y}_p = (y_{p,1}, \dots, y_{p,J_3})^T$.

The RBFN is a universal approximator [17]. If the RBF is appropriately chosen, the RBF network can theoretically approximate any continuous function arbitrarily well. The Gaussian RBF is expressed as $\phi(r) = \exp(-r^2/2\sigma^2)$ where $r > 0$ represents the distance from a data point $\vec{x}$ to a center $\vec{c}$ and σ is utilized to control the smoothness of the interpolating function. The Gaussian RBF is a localized RBF with the property that $\phi(r) \to 0$ as $r \to \infty$.

Training of a RBFN is usually performed by a two-phase strategy. During the first phase, suitable centers $\vec{c}_k$ and their corresponding standard deviations, σ_k, also known as widths or radii are determined. The network weights $\boldsymbol{W}$ are adjusted in the second phase. The training approach that is followed in this research is the supervised learning of all the parameters by the relatively simple gradient descent method.

B. Training

There is one output unit for each of the five terrain class values (asphalt, carpet, dirt, paving, and tiles). The model trained for the ith output unit (class value) is given by:

$$y_i(x_1, x_2, \dots, x_m) = g\left(w_{i,0} + \sum_{k=1}^{b} w_{i,k} \exp\left(-\sum_{j=1}^{m} \frac{(x_j - c_k)^2}{2\sigma_{global}^2}\right)\right), \tag{3}$$

where the activation function $g(\cdot)$ is a logistic function [19]. A Gaussian RBF network with the same global variance parameter σ_{global} for all RBF centers still has universal approximation capability [17]. The appropriate parameter values for $w_{i,k}$ and σ_{global} are found by identifying a local minimum of the penalized squared error on the training data. Given p classes, the error function can be expressed as

$$L_{SSE} = \left(\frac{1}{2}\sum_{k=1}^{n}\sum_{i=1}^{p}\left(y_{k,i} - f_i(\vec{x}_k)\right)^2\right) + \left(\lambda \sum_{i=1}^{p}\sum_{k=1}^{b} w_{i,k}^2\right), \tag{4}$$

where $y_{k,i} = 0.99$ if data point $\vec{x}_i$ has the ith class value, and $y_{k,i} = 0.01$ otherwise. Instead of using 1.0 and 0.0, the values 0.99 and 0.01 are used to aid the optimization process. Additionally, in (4), L_{SSE}, is divided by n, the number of training data points, as this was determined through empirical observation to improve convergence with the optimization methods used [20]. Standard calculus is utilized to find the corresponding partial derivatives, which consist of the gradients of the error function with respect to the network parameters. Backpropagation is employed to calculate the partial derivatives in the same manner as in Multilayer perceptrons. The hybrid conjugate gradient descent method specified by [21] is used for training.

Initialization of the network parameters is another important aspect of the training procedure. The initial weights of the output layer are sampled from $\mathcal{N}(0, 0.1)$. This strategy was empirically determined based on the familiar heuristic of choosing small, randomly distributed initial weights [20].

As the k-means algorithm is often used to train the hidden layer of the RBFN in an unsupervised process, it is utilized to determine the initial hidden unit centers c_k. Furthermore, the initial value of the variance parameter σ_{global} is set to the maximum squared Euclidian distance between any pair of cluster centers. This ensures that the initial value of the variance parameter is not too small. The learning process is accelerated on a multi-core computer by parallelizing the calculation of the error function and its gradient on a user-specified number of threads.

Artificial neural networks (ANNs) such as RBFNs and MLPs can be considered as techniques that lie in machine learning middle ground, somewhere between artificial intelligence and engineering [22]. They use heuristic methods, because very often there is no theoretical basis to support the decisions about the ANN implementation, as well as mathematical techniques, such as mean-square error minimization. ANNs are comprised of a large class of various architectures. The RBFN and MLP are two of the most widely used neural network architectures in literature for regression and classification problems [23]. To put the application of the RBFN on terrain classification in context, an MLP constructed for the same purpose is also examined. Both types of neural network structures are good in pattern classification problems and also robust classifiers with the ability to generalize for imprecise input data. A general difference between the RBFN and MLP is that the RBFN performs a local type of learning, which is responsive only to a limited section of the input space. In contrast, the MLP is a more distributed type of approach. The output of an RBFN is produced by mapping distances between the input vectors and center vectors to outputs through a radial function, whereas the MLP output is produced by linear combinations of the outputs of hidden layer nodes in which a weighted average of the inputs is mapped by every neuron through a sigmoid function. In the next section, the MLP architecture and training procedure are considered.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

III. MULTILAYER PERCEPTRONS

Similar to a RBFN, the MLP neural network is capable of arbitrary input-output mapping [24]. With its powerful universal approximation capability, it has been shown that MLPs with an appropriate number of hidden neurons can implement any continuous function. The MLP is extensively used in classification, regression, prediction, system identification, control, feature extraction, and associative memory. An MLP, like a RBFN, is estimated by a supervised procedure where the network constructs the model based on examples in the data with known outputs.

A. Architecture

In most cases, an MLP has several layers of nodes. External information is received at the first or lowest layer. The problem solution is obtained at the highest layer which is an output layer. Between the input layer and output layer there are one or more intermediate layers called the hidden layers. The number of hidden layers is a very important parameter in the network. Bordering nodes are normally fully connected from a lower layer to a higher layer. No lateral connection between neurons in the same layer, or feedback connection is possible. The MLP estimates a functional relationship, which can be written as $y = f(x_1, x_2, \dots, x_m)$, where $x_1, x_2, \dots, x_m$ are m independent variables and y is the dependent variable. Functionally, the MLP in this sense is equivalent to a nonlinear multiple regression model.

A single hidden layer MLP network with h neurons (Figure 2) and c outputs has the following form:

$$y_c(x_1, x_2, \dots, x_m) = g\left(w_0 + \sum_{k=1}^{h} w_k \tanh\left(w_{0k} + \sum_{j=1}^{m} w_{jk} x_j\right)\right), \quad (5)$$

where $g(\cdot)$ is the activation function, and w_i, w_{jk} the weights.

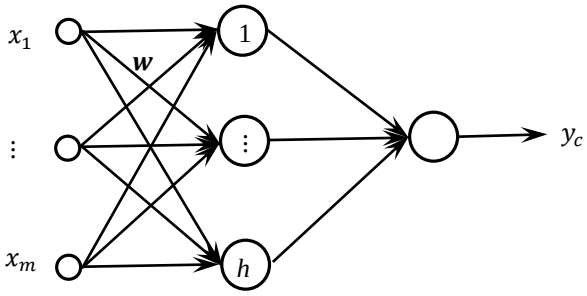


Figure 2. MLP network architecture

The model in (5) can be expressed in matrix form as $y_c = g(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$, where y_c is the output, $g(\cdot)$ the activation function, $\boldsymbol{W} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,m} \\ \cdots & \cdots & \cdots \\ w_{h,1} & \cdots & w_{h,m} \end{bmatrix}$ a $[H \times M]$ weight matrix, $\boldsymbol{x} = [x_1, x_2, \dots, x_m]$ the input vector, and $\boldsymbol{b} = [b_1, b_2, \dots, b_h]$ the bias vector [18].

B. Training

The backpropagation algorithm used to train an MLP was first discovered by [25] and later popularized by [26]. During the training phase, a set of input-output pairs is utilized for training and is repeatedly presented to the network. When training is stopped, the performance of the network is tested. The learning algorithm includes a forward-propagating step, followed by a backward-propagating step. On the whole, the algorithm is as follows:

***input**: training set, weight vector w*
***output**: optimal weight vector w**
repeat
  repeat
    repeat
      Initialize the weights w to small random values.
      Select an instance t, which is a data point from the training set.
      Apply the network input vector to the network.
      Calculate the network output vector z.
      For each of the outputs c, calculate the errors, which is the difference (δ) between the target output and the network output.
      Minimize this error by calculating the necessary updates for the weights (Δw).
      Add the calculated weights' updates (Δw) to the accumulated total updates (ΔW).
    ***until** number of instances comprises an epoch*
    Adjust the weights (w) of the network by ΔW.
  ***until** all instances in the training set are considered. This forms one iteration.*
***until** the error for the entire system (error δ or cross-validation set error) is satisfactorily low, or a pre-defined number of iterations is completed.*

Algorithm 1. Backpropagation algorithm

During training, the backpropagation algorithm performs gradient descent on the error surface by adjusting each weight in proportion to the gradient of the error surface at its location. It is well known that gradient descent can sometimes cause networks to get stuck in a local minimum in the error surface should such a local minimum exist. These local minima correspond to a partial solution for the network given the training data. At best, a global minimum is desired (the lowest error value possible), however, the local minima are surrounded by higher error values and the network usually does not escape these local minima by employing the standard algorithm. To get out of a local minimum, special techniques should be used. These include

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

varying the number of hidden units, changing the learning parameter (α), but especially by using the momentum term (η) in the algorithm. This term is generally chosen between 0 and 1. Taking into account the momentum term, the formula for modifications of weights at epoch $t + 1$ is given by

$$\Delta w_{kj}(t+1) = \eta \delta_k x_m + \alpha \Delta w_{kj}(t), \quad (6)$$

where j denotes the specific neuron. The network can oscillate, or more seriously, get stuck in a local minimum with incorrect values of these parameters.

Regardless of the many favorable characteristics of ANNs, constructing a neural network model for a particular problem is a nontrivial task [24]. Modeling issues that have an effect on the performance of an ANN must be carefully taken into account to ensure the successful application of the ANN. These issues are briefly examined next.

IV. ARTIFICIAL NEURAL NETWORK MODELING ISSUES

One of the critical decisions that must be made when building an ANN model is to determine a suitable architecture, specifically the number of layers, the number of nodes in each of the layers, and the number of connections that join the nodes. Additional network design decisions comprise the choice of activation functions for the hidden and output nodes, the training algorithm, data normalization or transformation methods, training and test data sets, and performance metrics.

A. Network architecture

An ANN is normally formed by layers of nodes. All the input nodes are grouped in the input layer, all the output nodes are in the output layer and the hidden nodes are allocated in one or more hidden layers in the middle. When constructing the ANN, the following variables must be determined:

- the number of input nodes;
- the number of hidden layers and hidden nodes; and
- the number of output nodes.

Selection of these parameters is inherently dependent on the problem. Many different methods to determine the optimal architecture of an ANN exist, but many of these methods are relatively complex in nature and difficult to implement. Examples include the network information criterion [27], the polynomial time algorithm [28], the canonical decomposition technique [29] and the pruning algorithm [30][31]. In addition, none of these methods is able to guarantee the optimal solution for all problems. Currently, there is no simple explicit method to choose these parameters. The guidelines are either based on simulations obtained from limited experiments or heuristic in nature. Therefore, the design of an ANN can be considered more of an art than a science.

A.1 Number of input nodes

The number of input nodes coincide with the number of variables in the input vector used to model target values. Given a specific problem, the number of inputs is usually transparent and relatively easy to choose.

A.2 Number of hidden layers and nodes

Many successful applications of neural networks are highly dependent on the hidden layer(s) and nodes. The hidden nodes in the hidden layer(s) enable a neural network to detect features, capture patterns in the data and to perform complex nonlinear mappings between input and output variables. It is evident that without hidden nodes, simple perceptrons with linear output nodes are equivalent to linear statistical forecasting models. Since theoretical works show that a single layer is sufficient for ANNs to approximate any complex nonlinear function to any desired accuracy [32], a single hidden layer is often used for modeling purposes. Unfortunately, one hidden layer networks may involve a very large number of hidden nodes, which is undesirable in that the network generalization ability and training time will get worse. Two or more hidden layer MLPs may provide more benefits for some types of problem [33][34]. Many authors focus on this problem by considering more than one hidden layer.

Determining the optimal number of hidden nodes is a crucial yet complicated issue. In most cases, networks with fewer hidden nodes are favored as they overfit less and usually have a better generalization ability. However, networks with too few hidden nodes may not have enough power to model and learn the data. There is no theoretical principle for choosing this parameter though a number of systematic approaches exist. Methods for increasing hidden nodes and pruning out unwanted hidden nodes have been proposed. A grid search method used to determine the optimal number of hidden nodes was put forward by [35]. The most common way to establish the number of hidden nodes is by means of experiments or trial-and-error. Various rules of thumb have also been suggested such as each weight should have at least ten input data points (referring to the sample size), and the number of hidden nodes should be determined by the number of input patterns. Some researchers have presented empirical rules to assist in avoiding the overfitting problem by restricting the number of hidden nodes [24]. Additionally, the number of hidden nodes was limited by a heuristic constraint by [36]. A number of practical guidelines exist in the case of the common one hidden layer networks, which include $\frac{n}{2}$ [37], $2n$ [38] and $2n + 1$ [39], where n denotes the number of input nodes. Nevertheless, none of these heuristic choices works well for all problem contexts.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A.3 Number of output nodes

As in the case of the number of input nodes, the number of output nodes is relatively easy to determine as it is directly related to the problem being modeled.

B. Interconnection of nodes

The behavior of a network is essentially determined by the connections between nodes. In most applications, the networks are fully connected with all nodes in one layer being connected to all the nodes in the next, higher layer, excluding the output layer. Sparsely connected networks [40] or direct connections between input nodes and output nodes [41] are, however, possible. The latter may be beneficial to predictive accuracy since it can be utilized to model the linear structure in the data and might increase the recognition power of the network.

C. Activation function

The activation function determines the relationship between the inputs and outputs of a neuron and the rest of a network. This function establishes a degree of nonlinearity that is valuable for most ANN applications. In theory, any differentiable function can be used as an activation function, but in practice, only a small number of activation functions are used. Some heuristic rules exist for the selection of the activation function. When learning about average behavior such as terrain classification, [42] suggests logistic activation functions.

D. Training algorithm

Training of a neural network is an unconstrained nonlinear minimization problem where weights of a network are iteratively adjusted to minimize the overall squared error or mean between the actual and desired output values for all the output nodes over all inputs patterns. Many different optimization methods to use for neural network training exist. Currently, there is no algorithm available to guarantee the global optimal solution for a general nonlinear optimization problem in a reasonable amount of time. In practice, all optimization algorithms suffer from the local optima problem. A solution to this problem is to use the available optimization method, which produces the "best" local optima if the true global solution is not available. The backpropagation algorithm is the most widely used training method.

E. Data normalization

Nonlinear activation functions such as the hyperbolic tangent function usually have a squashing role in restricting or compressing the possible output from a node to typically (0,1) or (-1, 1). Often, data normalization is performed before the training process begins. When nonlinear activation functions are used at the output nodes, the desired output values must be transformed to the range of the actual network outputs. Even when a simple linear output transfer function is utilized, it may still be beneficial to standardize the outputs together with the inputs to facilitate network learning, meet algorithm requirements and to avoid computational problems. Four methods to normalize inputs are presented by [43]: along channel (independent input variable) normalization, across channel (each input vector independently) normalization, mixed channel (combinations of along and across) normalization, and external normalization where all the training data are normalized into an explicit range.

F. Training and test samples

A training and test sample are typically involved when building an ANN model. The training sample is used for developing the model and the test sample for evaluating the predictive ability of the model. At times a set called the validation sample is also put to use to avoid the overfitting problem or to determine a stopping point for the training process. An important issue is the division of the data into the training and validation sets. One common approach is to use k-fold cross-validation [44] where a data set ($\mathcal{D}$) is randomly split into k mutually exclusive subsets (the folds) $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$. A model is then trained and tested k times; for each time $t \in \{1, 2, \ldots, k\}$, it is trained on $\mathcal{D} \setminus D_t$ and tested on $\mathcal{D}_t$. The cross-validation estimate of accuracy is the overall number of correct classifications divided by the number of instances in the data set.

G. Performance measures

In spite of many performance measures for an ANN model, such as training time and modeling time, the most important measure of performance is the prediction accuracy the model can produce beyond training data. Nevertheless, academics and practitioners do not universally accept a suitable measure of accuracy for a given problem. An accuracy measure is frequently defined in terms of the prediction error, which is the difference between the desired (actual) and the predicted value. There are a number of accuracy measures in the prediction literature and each has its advantages and limitations [45].

In the next section, the experiments that are performed to determine the RBFN terrain classification accuracy and comparison with an MLP, Naïve Bayes method and SVM will be discussed. Modeling issues in Section IV will be taken into account to construct the best RBFN and MLP architectures.

V. EXPERIMENTAL DESIGN

The aim of the experiments is to identify the type of terrain being traveled on by a mobile robot from a list of candidate terrains. Figure 3 shows the Lego Mindstorms EV3 experimental platform used in the investigation. The mobile robot has a Raspberry Pi 2 computer attached to the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

front with a Sense HAT inertial measurement unit (IMU) in turn connected to the Raspberry Pi. The Sense HAT is readily available and includes the following sensors: a gyroscope, an accelerometer, and a magnetometer. The mobile robot platform is battery powered and moves on rubber treads. An additional battery pack (not shown) is mounted on top and powers the Raspberry Pi computer. The five terrain types used in the study are displayed in Figures 4 to 8.

Figure 3. Lego Mindstorms EV 3 mobile robot

The terrain (asphalt, carpet, dirt, paving, or tiles) on which the mobile robot is currently travelling is identified in real time. The assumption is that the vibrations induced in the test vehicle and measured by the output of the IMU sensor represent a signature which can be used to accurately classify a terrain. The data for each terrain is sampled at an irregular rate of $\approx 16\frac{2}{3}$ Hz for a 600-second duration. The RBFN is then trained offline using the RBFN training scheme discussed in Section II (B) and the MLP by the backpropagation algorithm discussed in Section III (B). Three outdoor terrains (asphalt, dirt, and paving) and two indoor terrains (carpet and tiles) were analyzed.

Figure 4. Asphalt

Figure 5. Carpet

The RBFN architecture for this specific problem has five outputs that serve to identify the terrain type. Each of the output values $y_i \in [0,1]$ denotes the likelihood that a given signal presented as an input to the RBFN matches one of the five candidate terrains. In addition, the RBFN architecture has twelve inputs, which correspond to the dimension of the input signal data point. Each of these input signal data points received from the Sense HAT IMU can be denoted as:

$$[p\ r\ y\ a_x\ a_y\ a_z\ g_x\ g_y\ g_z\ m_x\ m_y\ m_z],$$

where $p, r,$ and y denote the pitch, roll and yaw (measured in degrees), a is the linear acceleration (m/s^2) measured in three dimensions (a_x, a_y and a_z), g is the rate of turn (degrees/seconds), also measured in three dimensions (g_x, g_y and g_z) and m denotes the earth's magnetic field

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

(gauss), measured in three dimensions (m_x, m_y and m_z) of the mobile robot, respectively.

Figure 6. Dirt

Figure 7. Paving

Figure 8. Tiles

The Weka system [20] was used for data processing, presentation, classifier training and testing. The terrain classification training data set contained twelve inputs, five outputs and a total of 49993 IMU sensor samples.

Before training started, all inputs in the data were normalized to the [0, 1] interval. This data was transformed back into the original space when predictions were produced. These same transformations were performed for new inputs when the predictions were made.

For the experiments, 10-fold cross-validation was performed. Results obtained by the RBFN were compared to those found by the MLP model, and default SVM and Naïve Bayes techniques. The latter are two popular methods found in the literature used for supervised terrain classification [11][12]. Sigmoid activation functions were utilized for the hidden and output nodes of the MLP architecture. A grid search was applied to obtain the best number of RBFN and MLP hidden nodes and number of hidden layers for the MLP. Coincidentally, the best results were obtained with 120 hidden nodes for both the RBFN and single hidden layer fully connected MLP. In the following section, the results will be discussed.

VI. DISCUSSION

The classification accuracy results obtained by the experiments are shown in Figure 9.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

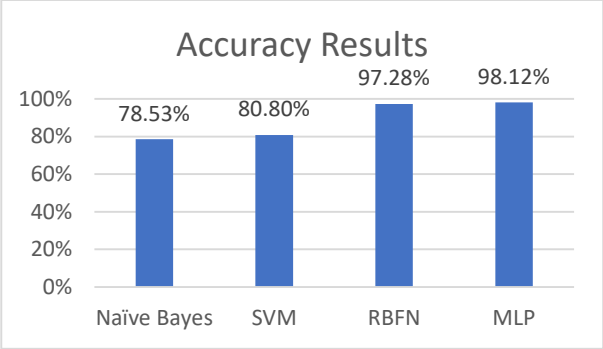


Figure 9. Terrain classification results

From the figure it can be observed that the machine learning algorithms, ordered from worst classification accuracy to best, are the Naïve Bayes, SVM, RBFN and MLP. The latter two techniques produced nearly the same classification accuracy with the MLP slightly outperforming the RBFN. The results presented in Figure 9 show that the RBFN is a feasible terrain classification technique compared to the former two models. Although the MLP exhibited a slightly higher classification performance than the RBFN, the results indicate the good predictive capability of the RBFN. In addition, the RBFN applied to terrain classification has the following advantages and disadvantages:

- Compared to the MLP, the RBFN has less model complexity, exhibits better comprehensibility, is easier to construct due to its simpler fixed three-layer structure and has a fast learning algorithm.
- Special techniques exist to increase the interpretability of RBFNs, thereby reducing the black box effect of neural networks in general [46].
- With regards to generalization, RBFNs can respond well for patterns which are not used for training [47].
- The stability of the designed RBFN model is enhanced by its strong tolerance to input noise [47].
- An ensemble of RBFN models can be constructed to increase the accuracy of a RBFN model. In some cases, this ensemble model can surpass an MLP model [48] in terms of classification accuracy.
- No pre-processing of the input sensor data is performed as in some previous studies.
- Classification of the terrain can be performed in real time because of the onboard IMU contact sensor.
- In terms of predictive accuracy, the RBFN outperformed the Naïve Bayes technique and the SVM model by a relatively large margin.
- A limitation of the RBFN model, however, is that it has greater difficulties if the number of hidden units is large and it is more sensitive to dimensionality [23].

Based on the small difference in classification accuracy between the RBFN and the MLP and the advantages of the RBFN, it can be concluded that it is reasonable to consider the RBFN as a competitive method for supervised terrain classification.

VII. Conclusion and Future Work

In this paper, real-time classification of five given terrains was performed with a RBFN. In contrast to some other techniques found in the literature, no pre-processing of the mobile robot platform's IMU vibration sensor data was performed. Eliminating feature extraction reduces the computational overhead needed to identify the terrain in real-time. The results have shown that even without feature extraction, the RBFN is still a feasible model for contact sensor-based terrain classification compared to other popular models used for this task. It can be used as an alternative to the MLP model due to its simpler structure and shorter training times. The RBFN has the capability to accurately recognize complex vibration signature patterns and can easily adapt to new terrain signatures by providing the model with new training examples. Unfortunately, compared to the other techniques, offline training of the model can be time consuming.

Future work includes a more detailed comparison with the existing methods. Metrics, such as latency (velocity) can be included in the results. Finally, an investigation into the feasibility of the RBFN model applied to other types of robots and how they must be adapted for this task can be performed.

Acknowledgment

The authors would like to thank Mr. Ryno Marx for assembling the mobile robot platform and for acquiring the vibration sensor data for the five terrains.

References

[1] J. V. du Toit and H. A. Kruger, "Terrain Classification Using a Radial Basis Function Network," in Proceedings of the Twelfth International Multi-Conference on Computing in the Global Information Technology (ICCGI), Nice, France, 2017, pp. 11-16.

[2] T. Kurban and E. Besdok, "A Comparison of RBF neural network training algorithms for inertial sensor-based terrain classification," Sensors, vol. 9, pp. 6312-6329, 2009.

[3] D. Sadhukhan, "Autonomous ground vehicle terrain classification using internal sensors," Florida State University, Master's thesis, 2004.

[4] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," Journal of Field Robotics, vol. 23(2), pp. 103-122, 2006.

[5] F. L. Garcia Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel, and R. S. Fearing, "Performance analysis and terrain classification for a legged robot over rough terrain," "IEEE/RSJ International Conference on Intelligent Robots and Systems", Vilamoura, Algarve, Portugal, October 7-12, 2012.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[6] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, MN, USA, 2007, pp. 1-8.

[7] A. Talukder et al., "Autonomous terrain characterization and modelling for dynamic control of unmanned vehicles," in Proceedings of the IEEE Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, 2002.

[8] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," Autonomous Robots, vol. 18, pp. 81-102, 2005.

[9] B. Park, J. Kim, and J. Lee, "Terrain feature extraction and classification for mobile robots utilizing contact sensors on rough terrain," Procedia Engineering, vol. 41, pp. 846-853, 2012.

[10] R. Jitpakdee and T. Maneewarn, "Neural networks terrain classification using inertial measurement unit for an autonomous vehicle," SICE Annual Conference, The University Electro-Communications, Japan, 2008.

[11] C. C. Ward and K. Iagnemma, "Speed-independent vibration-based terrain classification for passenger vehicles," Vehicle System Dynamics, vol. 47, no. 9, pp. 1095-1113, 2009.

[12] M. Happold, M. Ollis, and N. Johnson, "Enhancing supervised terrain classification with predictive unsupervised learning," Robotics: Science and Systems II, University of Pennsylvania, Philadelphia, 2006.

[13] T. Y. Kim, G. Y. Sung, and J. Lyou, "Robust terrain classification by introducing environmental sensors," IEEE International Workshop on Safety Security and Rescue Robotics (SSRR), 2010.

[14] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," Complex Systems, vol. 2, no. 3, pp. 321-355, 1988.

[15] C. S. K. Dash, A. K. Behera, S. Dehuri, and S.-B. Cho, "Radial basis function neural networks: a topical state-of-the-art survey," Open Computer Science, vol. 6, no. 1, pp. 33-63, 2016.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, D. E. Rumelhart and J. L. McClelland, Eds., vol. 1, pp. 318-362, MIT Press, Cambridge, Mass, USA, 1986.

[17] Y. Wu, H. Wang, B. Zhang, and K.-L. Du, "Using radial basis function networks for function approximation and classification," International Scholarly Research Network, Applied Mathematics, Volume 2012, doi:10.5402/2012/324194.

[18] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, "Neural network design," 2nd edition, Martin Hagan, USA, 2014.

[19] E. Frank, "Fully supervised training of Gaussian Radial basis function networks in WEKA," [Online]. http://www.cs.waikato.ac.nz/~ml/publications/2014/rbf_networks_in_weka_description.pdf 2018.11.10.

[20] E. Frank, M. A. Hall, and I. H. Witten, "The WEKA workbench. Online appendix for 'Data mining: Practical machine learning tools and techniques'," Morgan Kaufmann, Fourth Edition, 2016.

[21] Y. H. Dai and Y. Yuan, "An efficient hybrid conjugate gradient method for unconstrained optimization," Annals of Operations Research, 103, pp. 33-47, 2001.

[22] S. E. Jørgensen and B. D. Fath, "Multilayer Perceptron," Encyclopedia of Ecology, Academic Press, pp. 2455-2462, 2008.

[23] I. Yilmaz and O. Kaynar, "Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils," Expert Systems with Applications, 38, pp. 5958-5966, 2011.

[24] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," International Journal of Forecasting, vol. 14, pp. 35-62, 1998.

[25] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," PhD thesis, Harvard University, 1974.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", Nature, 323 (6088):533–536, 1986.

[27] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion determining the number of hidden units for an artificial neural network model," IEEE Transactions on Neural Networks, 5 (6), pp. 865-872, 1994.

[28] A. Roy, L. S. Kim, and S. Mukhopadhyay, "A polynomial time algorithm for the construction and training of a class of multilayer perceptrons," Neural Networks, 6, pp. 535-545, 1993.

[29] Z. Wang, C. D. Massimo, M. T. Tham, and A. J. Morris, "A procedure for determining the topology of multilayer feedforward neural networks," Neural Networks, 7 (2), pp. 291-300, 1994.

[30] J. Sietsma and R. Dow, "Neural net pruning–Why and how?," In: Proceedings of the IEEE International Conference on Neural Networks, 1, pp. 325-333, 1988.

[31] M. Cottrell, B. Girard, Y. Girard, M. Mangeas, and C. Muller, "Neural modeling for time series: a statistical stepwise method for weight elimination," IEEE Transactions on Neural Networks, 6 (6), pp. 1355-1364, 1995.

[32] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Mathematical Control Signals Systems, 2, pp. 303–314, 1989.

[33] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.

[34] A. R. Barron, "A comment on Neural networks: A review from a statistical perspective," Statistical Science, 9 (1), pp. 33-35, 1994.

[35] W. L. Gorr and D. Nagin, J. Szczypula, "Comparative study of artificial neural network and statistical models for predicting student grade point averages," International Journal of Forecasting, 10, pp. 17-34, 1994.

[36] G. Lachtermacher and J. D. Fuller, "Backpropagation in time-series forecasting," Journal of Forecasting, 14, pp. 381-393, 1995.

[37] S. Kang, "An Investigation of the Use of Feedforward Neural Networks for Forecasting," PhD Thesis, Kent State University, 1991.

[38] F. S. Wong, "Time series forecasting using backpropagation neural networks," Neurocomputing, 2, pp. 147–159, 1991.

[39] R. Hecht-Nielsen, "Neurocomputing," Addison-Wesley, Menlo Park, CA, 1990.

[40] S. T. Chen, D. C. Yu, and A. R. Moghaddamjo, "Weather sensitive short-term load forecasting using nonfully

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

connected artificial neural network," In: Proceedings of the IEEE/ Power Engineering Society Summer Meeting, 91, SM 449-8 PWRS, 1991.

[41] K. A. Duliba, "Contrasting neural nets with regression in predicting performance in the transportation industry," In: Proceedings of the Annual IEEE International Conference on Systems Sciences, 25, 1991, pp. 163-170.

[42] C. C. Klimasauskas, "Applying neural networks. Part 3: Training a neural network," PC-AI, May/June, 20-24, 1991.

[43] E. M. Azoff, "Neural Network Time Series Forecasting of Financial Markets," John Wiley and Sons, Chichester, 1994.

[44] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," International Joint Conference on Artificial Intelligence (IJCAI), 1995.

[45] S. Makridakis, S. C. Wheelwright, and V. E. McGee, "Forecasting: Methods and Applications," 2nd ed. John Wiley, New York, 1983.

[46] L. Wang and X. Fu, "A simple rule extraction method using a compact RBF neural network", Advances in Neural Networks (ISNN): Second International Symposium on Neural Networks, Lecture notes in Computer Science, 3496, pp. 682-687, 2005.

[47] H. Yu, T. Xie, S. Paszczyñski, and B. M. Wilamowski, "Advantages of Radial Basis Function Networks for Dynamic System Design", IEEE Transactions on Industrial Electronics, vol. 58, no. 12, 2011.

[48] B. T. Pham, A. Shirzadi, D. T. Bui, I. Prakash, and M.B. Dholakia, "A hybrid machine learning ensemble approach based on a Radial Basis Function neural network and Rotation Forest for landslide susceptibility modeling: A case study in the Himalayan area, India", International Journal of Sediment Research, vol. 33, no. 2, pp. 157-170, 2018.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Newly Developed Low Power Tristate Buffers for Low-power and High Performance Applications with Limited Energy Resources

Karol Niewiadomski and Dietmar Tutsch
Chair of Automation and Computer Science - University of Wuppertal
Wuppertal, Germany
Email: {karol.niewiadomski, tutsch}@uni-wuppertal.de

***Abstract*—The adaptiveness of integrated electronics is a key feature in current and future mobile applications. Despite the continuous improvement of battery capacity, reconfiguration capabilities of integrated electronics are an inevitable step to cover the rising demand for processing power. Without any further adjustments for efficiency, power consumption becomes a limiting factor for runtime performance. Field programmable gate arrays provide suitable configuration capabilities, but lack of efficient power saving design measures. To overcome this challenge, different approaches were proposed in recent research activities. A substantial contributor to battery load are general purpose input output circuits, which serve the purpose of connectivity. In this paper, we present a modified tristate buffer, which is a key component in a typical general purpose input output design. Modifications for active power reduction and standby leakage current suppression are applied at circuit level to achieve a better energy efficiency. This new tristate buffer design is compared to already existing designs. Furthermore, this newly developed design is modified for either higher output impedance or faster switching frequency. All modifications done are explained and discussed by putting a focus on their pros and cons.**

Keywords–Tristate buffer; Battery lifetime; Power reduction; Leakage suppression; Output impedance.

I. INTRODUCTION

Mobile computing is the driving factor for innovations on the field of instant availability of information. It is expected to have the same computing performance in mobile devices like smartphones, tablets and even sometimes in vehicles, as known from high performance computers in very demandable applications [1]. Vehicles exhibit a rising degree of functions supporting autonomous driving which require fast evaluation of different driving situations in real time. This comes along with a urgent demand for continuously rising computing power, whilst batteries do not experience comparable proceedings in terms of higher capacities. Field programmable gate arrays (FPGAs) offer vast reconfiguration capabilities way beyond their earlier use case as glue logic [2]. Depending on the size of the FPGA in terms of number of configurable logic blocks (CLBs), different designs can be loaded into the FPGA and, therefore, synthesized by an intelligent routing of CLBs. However, in most cases implemented designs do not use all resources of reconfigurability, leading to a waste of energy due to leakage currents flowing through blocks in standby mode. Unused blocks inside an integrated circuit, which is intended to be used in mobile applications, should be switched off and only turned on again, if more resources are needed. Different design methodologies can be used on different hierarchical levels to realize a fine-grain and coarse-grain approach for reduction of consumed power. This can be achieved by an efficient combination of design decisions at circuit level, e.g., power gating, clock gating, dynamic voltage scaling, etc. [3]. A breakdown of a CLB into its single blocks reveals further possibilities to modify the schematics towards the intended low power purpose, e.g., optimization of configuration random access memory (CRAM) [4] and data flip-flops (D-FFs) [5]. In addition to that, investigations have shown that a noticeable amount of power is dissipated by the general purpose input outputs (GPIOs), which serve as a generic input / output device for integrated circuits [6]. As the number of reconfigurable / adaptive electronics in mobile applications is expected to grow continuously, we believe that special attention in terms of improvements or redesign should be allocated to these special circuitry, which can not be neglected for the sake of well interconnectivity in integrated circuits.

In this paper, we investigate a standard tristate buffer design on its most significant characteristics, which are dynamic power consumption, standby leakage current and $highZ$ capabilities. In Section II, we give an overview about related work and key aspects of dependencies between performance and power consumption. In Section III, we introduce a reference design of a tristate buffer and discuss typical characteristics in operation and standby. In Section IV, a newly implemented tristate buffer is presented and its benefits for energy sensitive usage are introduced. In Section V, we compare the simulation results of the different investigated designs. In Section VI, all previous discussions are summarized and concluded.

II. RELATED WORK

GPIOs are used in almost every integrated circuit as an interface to communicate with peripheral circuitry. These elements are designed for receiving data as inputs and to transmit data as output to other connected devices. Therefore, tristate buffers are bidirectional circuits with the ability to receive and to transmit logic signals by the same input/output pin. Due to this important functions, GPIO play a major role in consumed area of a chip and power consumption in each complex design [7]. Despite the importance of these crucial parts in each chip, GPIOs were optimized either for high speed or considerable driver capabilities in the past. Concepts for designing special function registers (SFRs) inside a GPIO have been presented [8] for handling difficulties of the rising design complexity of these blocks. These concepts are innovative and certainly of considerable value for future GPIO design, however, power aspects are not covered. Further research work focussed on the improvement of GPIO configurability by adding event-capture modules to the standard GPIO design [9]. This feature provides options to configure some registers by the user whilst power aspects are not in scope. IO transistors are used to

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

connect each chip to peripheral circuitry and should be taken into consideration when leakage currents shall be reduced. Thus, leakage optimization of IO transistors was handled by research activities in the past [10]. This is an important step ahead but a GPIO consists of further parts, which should be also carefully analyzed when overall power improvements are desired. Additional research activities have been focussing on maximizing circuit speed while still keeping the amount of dissipated power at an comparably acceptable level [11]. The results of this work were further elaborated and extended in advanced work, resulting in the implementation of low-power transmitter and receivers [12]. These implementations achieve good results for both, speed and low-power, but can not be categorized as real GPIOs. For that reason, the research activities of the work described in this paper were focusing on the development of a low-power GPIO. Operating frequency was not of first priority, but was also considered in alternative design, which also will be introduced in this paper. Figure 1 illustrates a simplified block diagram of a FPGA without any additional hard processing cores.

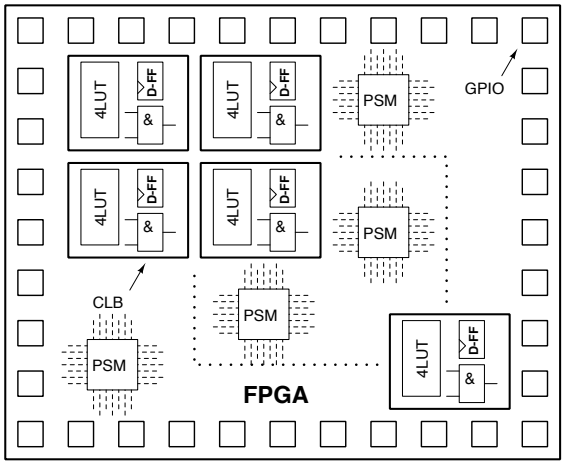


Figure 1. Simplified FPGA structure

As illustrated in Figure 1, all CLBs of this simplified internal hierarchy are surrounded by GPIO blocks. For the sake of simplicity, all further blocks, e.g., switching matrices (PSM), are displayed in a simplified way. In complex systems, several FPGAs may drive an internal bus for different purposes, e.g., data exchange, leading to potential conflicts when different circuits try to write different logic values to the same bus line.

Figure 2 highlights the described conflict and depicts a situation, in which two different FPGAs, connected to the same 4 bit data bus, drive the same line with different values: whilst FPGA1 drives one signal line of the bus with a logic 1 or V_{dd}, FPGA2 tries to do same but with a logic 0. The consequence is a floating voltage on the interconnection signal line, which is difficult to predict and an undefined state. For this reason tristate buffers play an important role inside each GPIO, since they offer one special output state beside their functionality to pass a logic value from the input to the output node: $highZ$, also called high impedance. By enabling this state, a tristate buffer cuts off the connection between its input and output node and, therefore, prevents an undesired throughput from the inputs of the GPIO inside an FPGA to the interconnection bus. So in general, we can identify three aspects to be of relevance

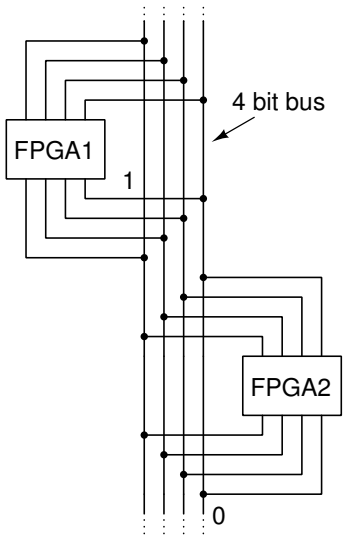


Figure 2. Interconnection bus

for optimization in terms of energy efficiency:

- Subthreshold / standby leakage
- Active power consumption
- $highZ$ behavior

Each of these bullet points has to be addressed by a careful analysis of parameters, which are responsible for different behavior and, therefore, also different results or performance of the circuit in scope. Subthreshold leakage current can be characterized by the following equations [13][14]:

$$J_{DT} \propto A(\frac{V_{ox}}{T_{ox}})^2 \tag{1}$$

$$I_{leak} \propto \frac{W}{L_{eff}} e^{(V_{GS}-V_{t0}-\gamma V_{SB}+\eta V_{DS})/nV_t)(1-e^{-\frac{V_{DS}}{V_t}})} \tag{2}$$

Equation (1) explains that a higher oxide thickness T_{ox} will subsequently lead to a lower current density J_{DT}, which is a favored effect for our purposes as we intent to limit undesired current flows as good as possible. On the other hand, (2) highlights the dependency of a subthreshold current I_{leak} to different factors, e.g., the transistor length L_{eff}, the gate-source voltage V_{GS} and the source-body voltage V_{SB}. On the other hand, active power consumption P_{dyn} depends on various factors showed in the following equation:

$$P_{dyn} = \alpha C_{load} V_{dd}^2 f_{Clk} \tag{3}$$

Equation (3) [15] shows that for significant reduction of consumed battery power several factors, e.g., the switching activity α, the load capacitance C_{load}, the supply voltage V_{dd} and the operating frequency f_{Clk} have to be designed in a way to keep P_{dyn} as low as possible. Some factors like C_{load} can not be easily controlled, however other factors can be adapted in a better way directly at circuit level. Last but not least, the $highZ$ attributes of a tristate buffer play an important role due to their ability to decouple this buffer from the remaining

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

signal chain. A careful design of the output transistors inside a tristate buffer offers heavy impact on this ability. Nevertheless, it should be stated here, that priority was put on low power characteristics of our newly implemented design. Measurement of the $highZ$ state with different output voltages was done after evaluating power consumption of all investigated designs. Furthermore, all measurements were compared against each other to figure out, which design performs best in general.

III. TRISTATE BUFFER CELL DESIGN

The basic purpose of a buffer circuit is to forward the input value with a certain delay to the output node. Some applications might require the addition of a delay time for synchronizing different data paths. The easiest way to understand the basic function of a buffer is to imagine the logic function of two inverter in series. A tristate buffer adds a third, important feature to this functionality: the $highZ$ state. For a better understanding of the circuit's function, a tristate inverter is shown in Figure 3.

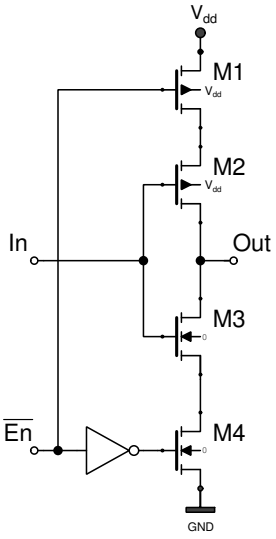


Figure 3. Tristate inverter

As long as $\overline{En}$ provides a logic *LOW* at the respective input node, the transistors *M1* and *M4* are turned on and subsequently provide a direct path to the voltage source V_{dd} and GND. As a consequence, the transistors *M2* and *M3* work as an inverter and, therefore, invert all signals applied to In. On the other hand, if $\overline{En}$ turns to *HIGH*, *M1* and *M4* are turned off and cut-off the internal transistors *M2* and *M3* from the supply voltage and ground path. In this special case, the voltage at the output node *Out* is floating and undefined. This means that in dependence of this floating voltage, only a very small current will flow either as leakage current from the tristate inverter into the circuitry connected to *Out* or from the load into the tristate inverter to GND. By adding one additional *nMOS* and *pMOS* transistor, the discussed tristate inverter can be modified to a tristate buffer, which is shown in Figure 4.

Different aspects of this tristate buffer's behavior have been investigated during simulations by a $90nm$ TSMC (Taiwan Semiconductor Manufacturing Company) technology and a Cadence toolchain (INCISIVE 6.1.5). All simulations, serving the purpose investigating the circuit's dynamic performance,

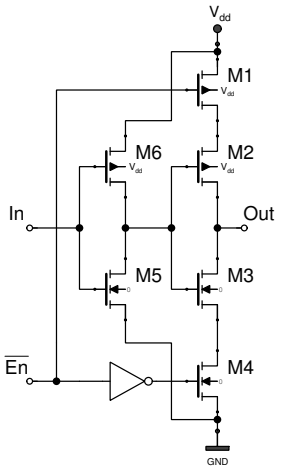


Figure 4. Standard design of a tristate buffer

were done at an operating frequency of $200MHz$ and with standard settings for all transistors' dimensions ($120nm$). Since all analyzed designs are not dynamic logic inheriting a dedicated Clk input, the operating frequency was modulated into the switching events of $\overline{En}$. The results of the first simulation run with active inputs are shown in Figure 5 and also displayed in Table I and Table II. This simulation was followed by further tests for alternative circuit states with the intention to build up a baseline database for further comparisons.

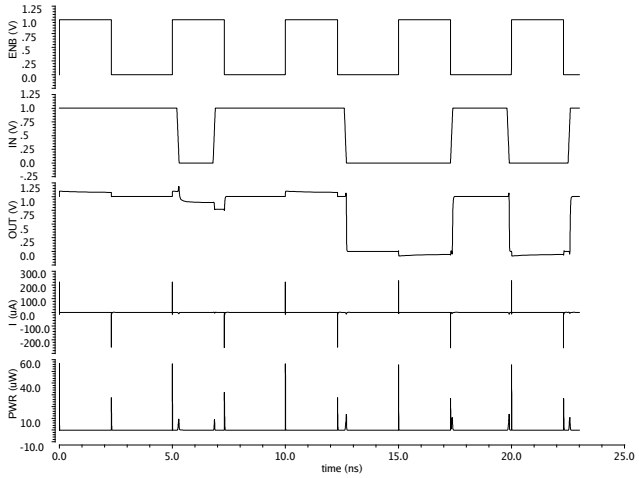


Figure 5. Simulation results of dynamic behavior of a standard tristate buffer

TABLE I. SIMULATION RESULTS (PWR)

Design type	*Average Power nW*	*Max. Power uW*	*Min. Power pW*
Reference	245	56.75	103.8

TABLE II. SIMULATION RESULTS I_{Vdd}

Design type	*Avg. Current nA*	*Max. Current uA*	*Min. Current nA*
Reference	215.2	230.5	261.4

The simulation results display the correct function of this tristate buffer, which directly passes the input value to *Out*

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

whenever $\overline{En}$ is set to *LOW*. Once $\overline{En}$ applies a logic *HIGH* to the cutoff transistors, the voltage level at *Out* starts to float and swings between voltage levels above V_{dd} and below $0V$ (GND). These floating voltages are not defined and also indicate that the whole circuit is in *highZ* mode. Active power dissipation is of high importance for the estimation of required energy resources, but regardless of these results it is also obligatory to have a closer look on the standby power characteristics when the circuit is lead into an idle phase or put completely into standby mode. This means that the data input is inactive and $\overline{En}$ active. The simulation results are shown in Figure 6.

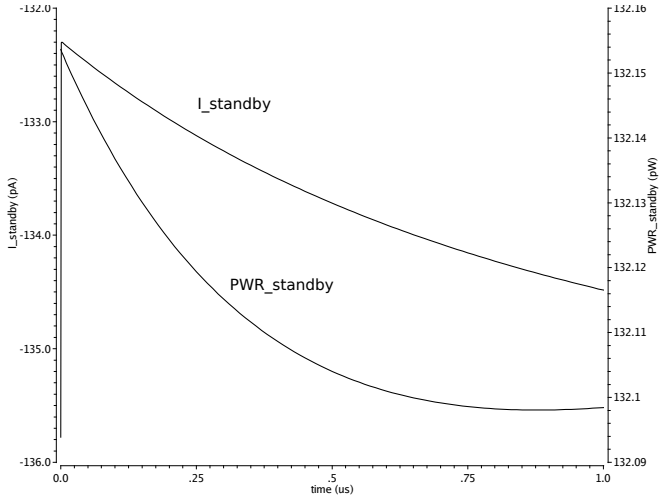


Figure 6. Simulation results of idle standby results

For this analysis and for a better observation of the standby current, the simulation runtime was set to $1\mu s$. The simulation curves of both, standby current and standby power dissipation, show the discharging process of all internal parasitic capacitances after powering on the circuit at the very beginning of the simulation process. Both, standby current and the allocated dissipated power, continuously decrease over time, resulting in an average leakage current I_{leak} of $133.6pA$ and a related average power dissipation of $132.1pW$.

The remaining aspect to be considered at this point is the behavior of the reference tristate buffer in *highZ* mode after setting $\overline{En}$ to *HIGH*. First of all, it should be stated here that there is no unambiguous answer on this question, since this depends on the voltage, which will be applied by the load to the output node *Out* of the tristate buffer. In addition to that, there is always a small throughput from the input node on the output in case that the tristate buffer in *highZ* is still stimulated with input date, which might be a realistic situation when the related control logic fails. Thus, two different situations, active and inactive inputs, must be considered. Based on the assumption that the voltage applied to *Out* may vary from $0V$ to$1V$, a dc sweep simulation was done. The results of both test runs are displayed in Figure 7.

Active input data has a remarkable influence on the circuit's capabilities to decouple its internal switching events to the output node. In case of setting the input node to a $0V$ and, therefore, making it 'inactive', Figure 7 reveals a sweetspot in terms of output impedance and which is closely allocated

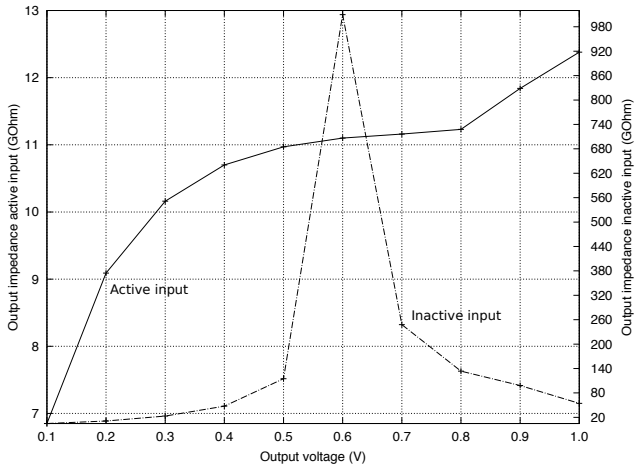


Figure 7. Simulation results of output impedance in *highZ* state

to almost $V_{dd}/2$. Having this striking high output impedance ($\approx$ $1.01T\Omega$) at this voltage range is a desirable situation, since this implements an almost perfect balance between current source and current sink. If input data are applied to *In*, a drop the output impedance can be observed after simulation. Even with turned of decoupling transistors *M1* and *M4*, the throughput originating from the buffer's input is strong enough to lower the impedance at *Out*. Therefore, a stronger decoupling mechanism would probably lead to better results.

IV. MODIFICATIONS

A careful analysis of the reference tristate buffer pointed out that there is still room left for different improvements. Thus, a noticeable adaption of circuits for sensitive low-power application can only be achieved by a synergy of different power savings measures for imaginable operating states.

A. Power Gating

On our way to develop a low-power tristate buffer, the implementation of a *'hold'*-mechanism for standby-phases was an inevitable step. The difficulty here was the fact, that this design does not imply clocked inputs, which could have been gated. Instead of this, a more stringent design technique was applied: power-gating. This modification can be applied in different ways, by adding a gating transistor between the supply voltage and the circuit or by inserting a transistor between *GND* and all internal nodes. A third alternative comes along with a combination of both design modifications and can be found in Figure 8 (transistors *M5* and *M10*).

The addition of two transistors to a design with a total number of eight transistors before this modification means an increase 20% and a high probability for penalty in terms of area consumption. Regardless of the chosen technology for synthesis and allocated continuous proceedings in technology node shrinking, a higher number of transistors is always considered as a drawback. On the other hand, this modification is responsible for a noteworthy limitation of leakage current running through the design under test (DUT), since we achieve a complete decoupling of the tristate buffer vor V_{dd} and *GND* respectively.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

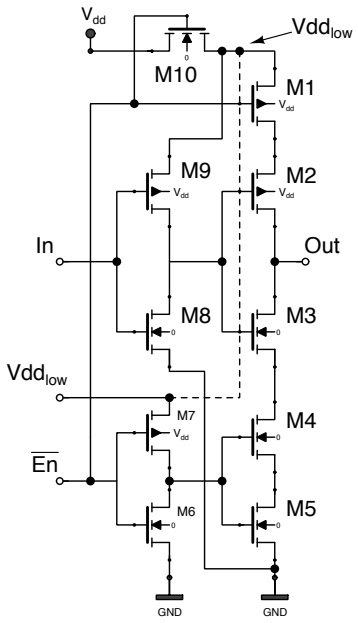


Figure 8. Low-power tristate buffer

B. Leakage Current Reduction

A reasonable extension of power gating is the use of special transistors with a higher oxide thickness T_{ox}, which can be also seen in (1), where T_{ox} is in the denominator and, therefore, has the ability to limit the electrons tunneling through the transistor's gate connector. This results in a reduction of the leakage current in idle / standby state. Despite these benefits it should be mentioned that high T_{ox} transistors have a slower switching frequency than standard T_{ox} do. Hence, adding these transistors should be carefully waived taking a decision upon it. In our case, *M5* and *M10* have an increased T_{ox} than the remaining transistors have for keeping the penalty in performance degradation as low as possible.

C. Subthreshold Current Reduction

Whilst power gating is an effective method for a total shutdown of a circuit, there should be an alternative for measurable reduction of a current flowing through a transistor with an applied gate-source voltage V_{gs} below the threshold. voltage V_{th}. This led to the decision to apply high V_{th} transistors, which have the ability to cut off subthreshold electron tunneling. This method might have a negative impact on the maximum operating frequency and should be carefully applied. Nevertheless, these special transistors can not be neglected during the design of low power designs. All internal transistors, except the gating transistors, have been replaced by their hight V_{th} counterparts and simulated.

D. Multi Supply Voltage

An operating circuit in low power applications should not only be optimized for static power reduction but also for energy efficiency in active mode. As shown in Equation 3, the supply voltage has a vast influence on the overall dissipated power. It's obvious that the best approach would be to decrease the global supply voltage V_{dd}, but might lead to the necessity of additional level restorers for a smooth signal transmission to other circuitry. An alternative is the careful partial supply voltage reduction within a design after analyzing certain parts of a design, which could be powered by a lower V_{dd}. On the other hand, lowering V_{dd} comes along with a slower computation time of the input values, therefore, a smaller supply voltage $V_{dd_{low}}$ was only applied to the internal inverter *M6* and *M7*. In principle, there are two different ways how to generate $V_{dd_{low}}$: this can be realized by an external voltage source (illustraed by the additional voltage source V_{ddlow} in Figure 8) or by exploiting internal voltage nodes (illustrated by the dashed line in Figure 8). The second option shows its beauty by an inherent voltage reduction automatism. Once $\overline{En}$ goes *HIGH M10* is turned off and, therefore cutting off *M7* from V_{dd}, but keeps the internal inverter still working. Minor adaptions to the width of *M7* have to made due to the decreased internal supply voltage. Nevertheless, both options work well with the low power tristate buffer.

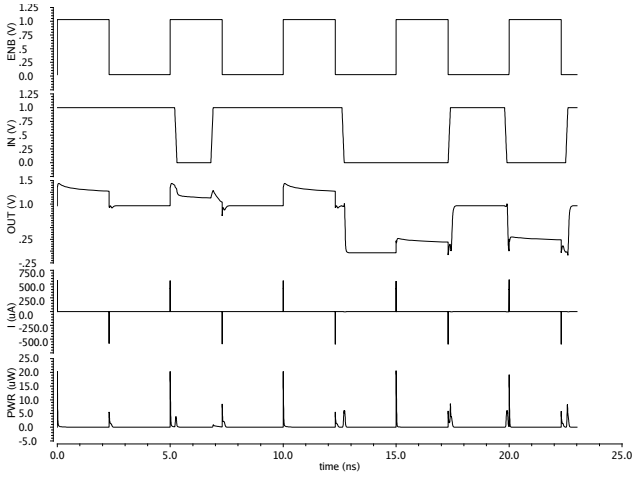


Figure 9. Simulation results of dynamic behavior of the low-power tristate buffer

Figure 9 shows the dynamic behavior of the low power tristate buffer. Compared to the simulation curves shown in Figure 5, it can be seen that the low power tristate buffer is superior in terms of dissipated power during active runtime. The related simulation results are summarized in Table III and Table IV.

TABLE III. SIMULATION RESULTS (PWR)

Design type	Average Power (*nW*)	Max. Power (μW)	Min. Power (*pW*)
LP tristate	191.3	29.72	22.36

TABLE IV. SIMULATION RESULTS I_{Vdd}

Design type	Avg. Current (*nA*)	Max. Current (μA)	Min. Current (*nA*)
LP tristate	225.8	194.8	206.9

Furthermore, an analysis of the standby behavior revealed an improved average leakage current I_{leak} of $24.1pA$ and a related average power dissipation of $22.04pW$. As a final step, the *highZ* characteristic was investigated for having a better comparison to the reference design. The simulation results are displayed in Figure 10.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

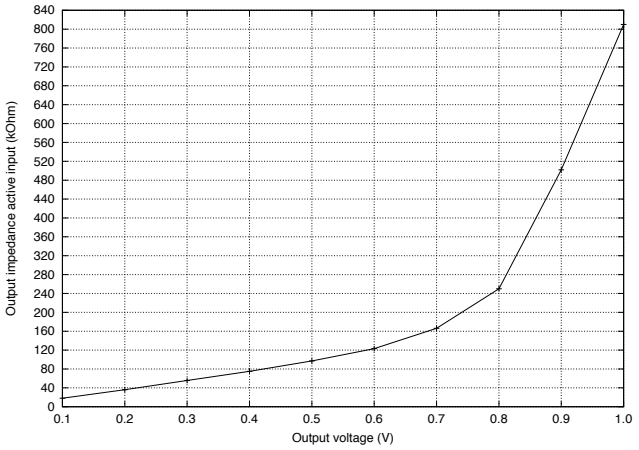


Figure 10. Output impedance in *highZ* state (low power tristate buffer)

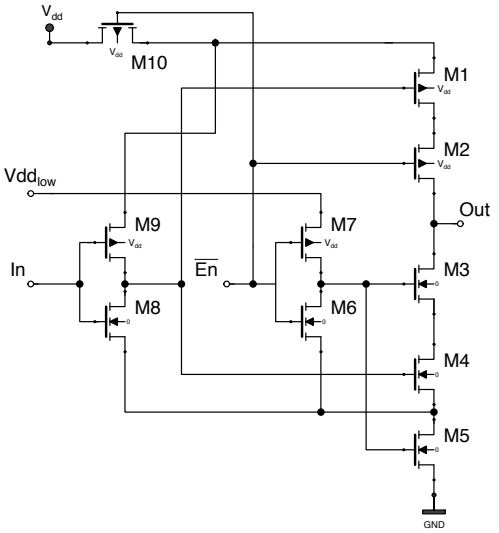


Figure 11. Low-power tristate buffer with swapped output transistors

In contrast to the reference tristate buffer, the newly implemented tristate buffer shows a different behavior. First of all, the output impedance shows a smaller order of magnitude ($G\Omega \rightarrow k\Omega$) and in addition to this, the output curve strongly depends on the voltage at the output node and reveals a proportional dependency. The higher the voltage at *Out* is, the higher the output impedance will be. Despite the fact that the low power tristate buffer's active *highZ* curve has a smaller order of magnitude, the evaluated results are still acceptable and give an evidence about the appropriateness for the usage as an connecting element in complex designs. These results could be improved by modifying the gate lengths of the output transistors *M2* and *M3*. The downside of this modification would lead to necessary modifications of the manufacturing process, but which can be easily handled by modern technology nodes. By picking up this thought, the question for an alternative design may arise. The internal buffer of the low power tristate buffer, consisting of transistors $M2, M3, M8$ and $M9$, provides a direct signal propagation path from *In* to *Out*, regardless of the built-in decoupling measures from the supply voltage and *GND* and, in the worst case, vice versa. Like mentioned before, tweaking the gate length of these transistors, especially of $M8$ and $M9$ would be a suitable way to raise resistance of each MOSFET, but this should be rather treated like the last option. Hence, anticipating any external throughput on internal nodes can be achieved by swapping the transistors of the internal (and simple) core logic with the cut-off switches, shown in Figure 11.

Now, the circuit is still similar to its predecessor shown in Figure 8 but features some mandatory adaptions. First of all, the internal buffer, which was placed right in the centre of the predecessor's circuit before, is now split up into the transistors $M1, M4, M8$ and $M9$. So, the idea here is that the second inverter of the internal buffer, implemented of $M1$ and $M4$, is now 'pulled apart' and flanked by transistors ($M2, M3, M5$ and $M10$), which serve as cut-off switches to V_{dd} and GND. All previously described energy saving measures have been applied here in the same way and simulated, respectively, to all tests done before. The simulation results can be seen in Figure 12.

As expected, the modified low power tristate buffer works fine and transmits correctly all input data in normal operation

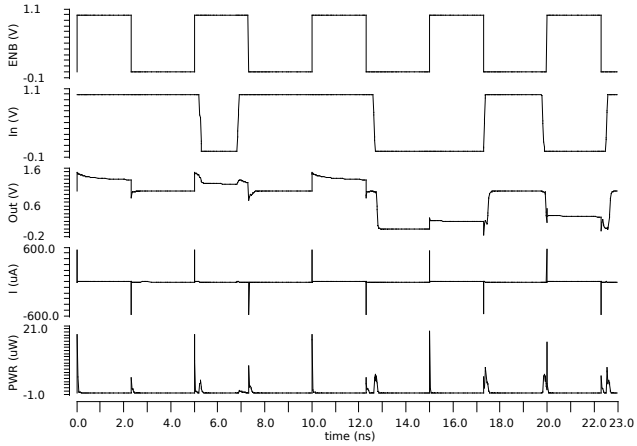


Figure 12. Transient analysis of modified low power tristate buffer

mode. Activation of the *highZ* mode works fine as well, by driving the output voltage to a not specified, floating voltage value. The 'kink' of *Out* during the falling edge if *In* goes back to the fact that the input signal switches *after* right after normale operation mode was activated by *ENB* switched from $1 \rightarrow 0$. Transient analysis in standby mode is pictured in Figure 15. An extended simulation with a transient analysis of $1ms$ is shown in Figure 14. After the verification of the correct function, the next step was to evaluate the circuit's power loss and to compare it with the previous design.

Compared to the results displayed in Table III and Table IV, it can be seen that the average power consumption is around 28.6% higher than before. But it should be taken into consideration that the internal inverter ($M6$ and $M7$) is still powered by V_{dd} and not by a lower, internal supply voltage. So, a few adjustments needs to be done here to figure out how this can be improved towards a lower power loss. Investigations of this circuits behavior have been made by scaling Vdd_{low} from the original $1V$ down to $500mV$ and are shown in Figure 16. This picture is a snapshot from the full simulation and all curve progressions in this figure depict a $1 \rightarrow 0$ transition while

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

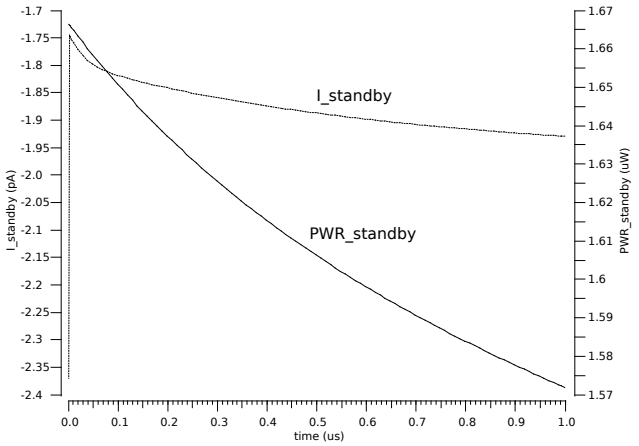


Figure 13. Transient analysis of modified low power tristate buffer in standby mode

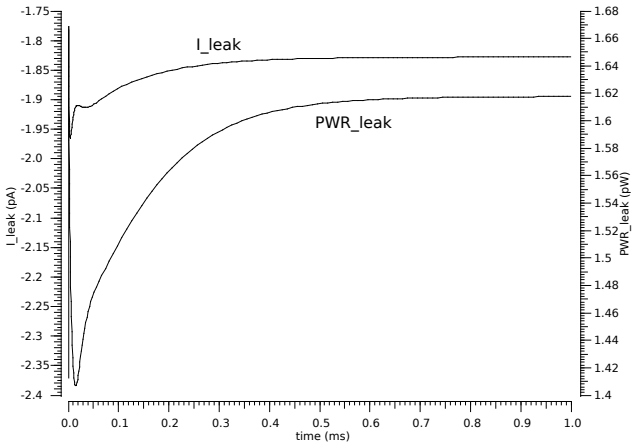


Figure 14. Extended simulation time of power dissipation and leakage current in power-off mode of the modified low power tristate buffer

leaving *highZ* mode and entering normal operation mode. It can be clearly seen how a lower internal supply voltage of the embedded inverter impacts signal propagation. Whilst lowering the supply voltage from $1mV$ to $800mV$, the penalty in signal delay is approximately $77ps$, further decreasing of Vdd_{low} down to $600mV$ leads to a higher penalty of approximaltely $622ps$. Going further down to 50% ($500mV$) of the original supply voltage, the correct function is not provided any more as a reliable inversion of *IN* gets disrupted.

TABLE V. SIMULATION RESULTS (PWR)

Design type	Av. PWR (*nW*)	Max. PWR (μW)	Min. PWR (*pW*)
LP tristate	268	19.81	17.69

TABLE VI. SIMULATION RESULTS I_{Vdd}

Design type	Av. Current (*nA*)	Max. Current (μA)	Min. Current (*nA*)
LP tristate	133.2	563.8	585.8

As expected, the modified low power tristate buffer works fine and transmits correctly all input data in normal operation mode. Activation of the *high Z* mode works fine as well, by driving the output voltage to a not specified, floating voltage value. The 'kink' of *Out* during the falling edge if *In* goes back to the fact that the input signal switches *after* right after normale operation mode was activated by *ENB* switched from $1 \rightarrow 0$. Transient analysis in standby mode is pictured in Figure 15. An extended simulation with a transient analysis of $1ms$ is shown in Figure 14. After the verification of the correct function, the next step was to evaluate the circuit's power loss and to compare it with the previous design.

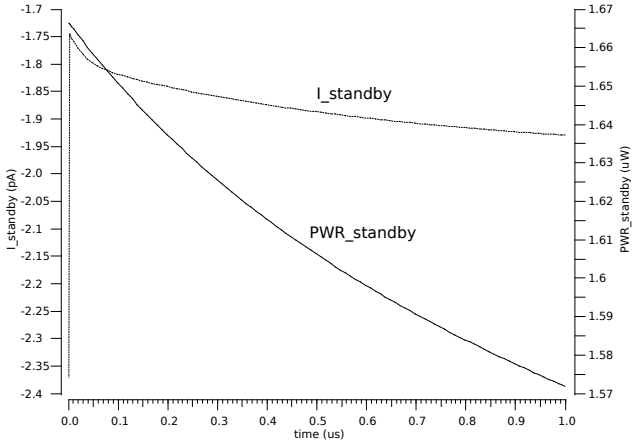


Figure 15. Transient analysis of modified low power tristate buffer in standby mode

Compared to the results displayed in Table III and Table IV, it can be seen that the average power consumption is around 28.6% higher than before. But it should be taken into consideration that the internal inverter ($M6$ and $M7$) is still powered by V_{dd} and not by a lower, internal supply voltage. So, a few adjustments needs to be done here to figure out how this can be improved towards a lower power loss. Investigations of this circuits behavior have been made by scaling $V_{dd_{low}}$ from the original $1V$ down to $500mV$ and are shown in Figure 16. This picture is a snapshot from the full simulation and all curve progressions in this figure depict a $1 \rightarrow 0$ transition while leaving *high Z* mode and entering normal operation mode. It can be clearly seen how a lower internal supply voltage of the embedded inverter impacts signal propagation. Whilst lowering the supply voltage from $1mV$ to $800mV$, the penalty in signal delay is approximately $77ps$, further decreasing of Vdd_{low} down to $600mV$ leads to a higher penalty of approximately $622ps$. Going further down to 50% ($500mV$) of the original supply voltage, the correct function is not provided any more as a reliable inversion of *IN* gets disrupted.

However, what is encountered at this point is a typical trade-off between power consumption and operating speed. Thus, the important aspect here is to make a distinction between thinkable target application of the device during development and then to put an appropriate nucleus on one these applications. Since energy efficiency comes first in the scope of this research work, the decision made was to set Vdd_{low} to $600mV$ and to continue further investigation about the circuit's characteristics. So, the next question to look up for was to see how the output impedance in *high Z* will look like. Similar to earlier investigations, simulation were carried out to check the modified low power tristate buffers capabilities in terms of decoupling itself from a bus. The outcome of

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

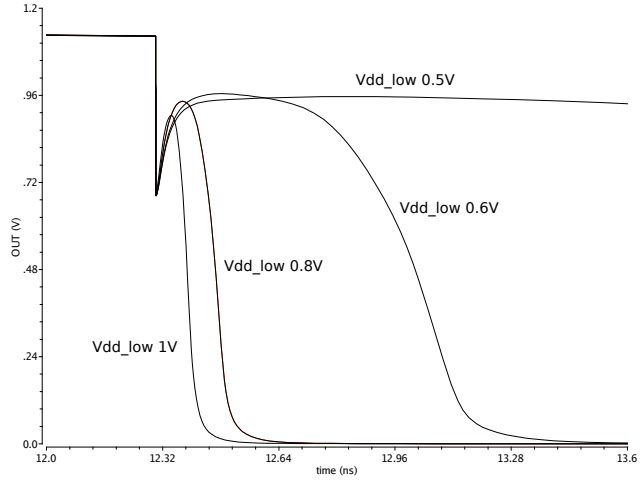


Figure 16. Transient analysis of modified low power tristate buffer with scaled Vdd_{low}

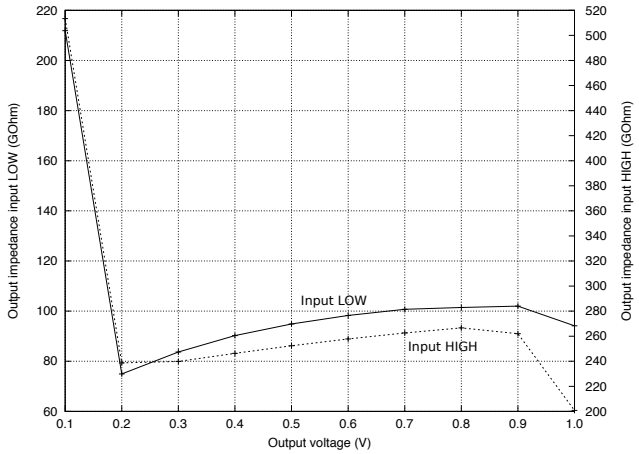


Figure 18. Output impedance in highZ state (modified low power tristate buffer)

these simulations are shown in Figure 18. In direct comparison to Figure 17 the output curves of the modified low power tristate buffer show an obviously more synchronized tracing, regardless of the applied input data. Since this is a feature of an improved decoupling mechanism from the output node(s), it can be considered as beneficial characteristic. For the next step, it was interesting to see how the circuits behave and compete in worst case scenarios. This is summarized in Table VII and in Table VIII.

TABLE VII. OUTPUT IMPEDANCE ($G\Omega$)

Type \In/Out	$0V$ \ $0V$	$0V$ \ $1V$
Ref	6.85	12.38
LP	189	95.69
Mod. LP	211.9	94.16

TABLE VIII. OUTPUT IMPEDANCE ($G\Omega$)

Type \In/Out	$1V$ \ $0V$	$1V$ \ $1V$
Ref	4.8	54.3
LP	189	197.4
Mod. LP	513.4	200.78

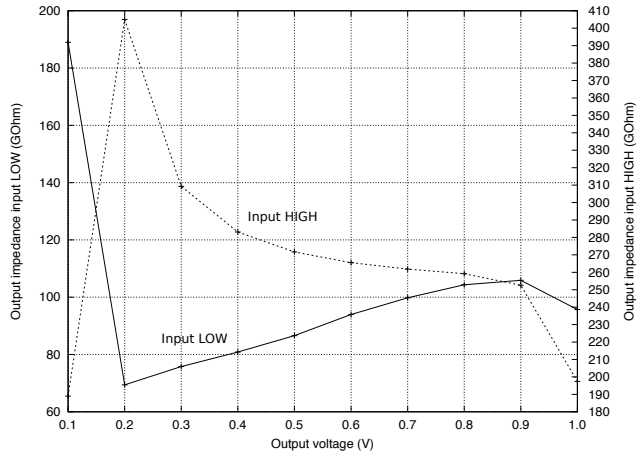


Figure 17. Output impedance in *highZ* state (low power tristate buffer)

After completing the low-power and *highZ* evaluation of the introduced circuits, the question came up whether it is still possible to achieve further improvements with a complete redesign of the common architecture. The circuit of this new buffer is illustrated in Figure 19.

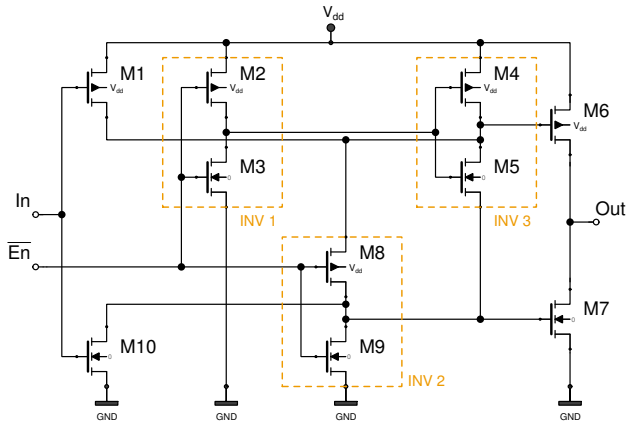


Figure 19. Alternative Tristate Buffer

The basic idea of this new, alternative (AT) tristate buffer is to have a separate decoupling of each of the four transistors ($M1$, $M6$, $M7$ and $M10$), which form the core buffer by additional inverters ($INV1$, $INV2$ and $INV3$). Buffer functionality is ensured by two inverters, consisting of $M1$ and $M10$ (input inverter) as well as $M6$ and $M7$ (output inverter). For normal operation mode $\overline{En}$ is turned to *LOW*, which leads to turning on the pMOS transistors $M2$ of $INV1$ and also $M8$ of $INV2$. So, $M2$ pulls the output of $INV1$ to V_{dd} and, therefore, turns on $M5$ of the next inverter ($INV3$). In parallel to this, $\overline{En}$ also turns on $M8$ of $INV2$, which short-circuits the drain node of $M1$ to the drain of $M10$. The short-circuit loop, which ensures this functionality is displayed in Figure 20.

By closing the internal shirt-circuit loop the output node of the input inverter is connected to input node of the output inverter, which leads to the desired buffer function in case that no *highZ* mode is demanded. If $\overline{En}$ is turned to *HIGH*,

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

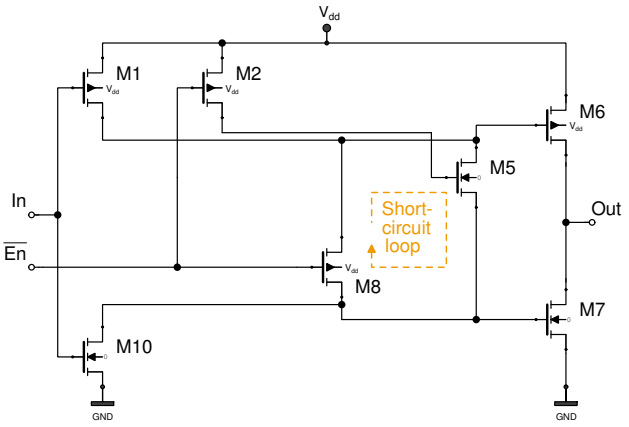


Figure 20. Short-circuit loop in normal operation mode

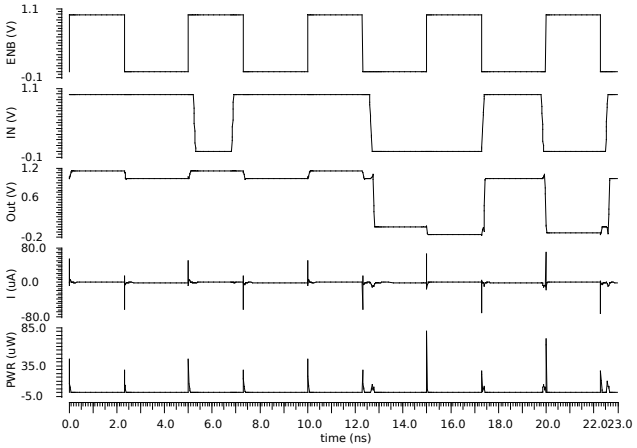


Figure 22. Transient analysis of alternative tristate buffer

transistor $M3$ of $INV1$ is activated and subsequently pulls the input node of $INV3$ to *GND*. As consequence $M4$ is turned on, pulling the output of $INV3$ to V_{dd}, which then turns off $M6$. At the same time, $\overline{En}$ also turns on $M9$ of $INV2$, which leads to turning off $M7$. So, the outcome of this procedure is that each of the output inverters' transistors is switched off separately and *Out* successfully decoupled from the tristate buffer. This is also illustrated in Figure 21 where all related, active transistors are displayed and all turned off transistors are faded out. These measures shall ensure a better, well balanced output impedance regardless of the input signal, which might be applied even when no data shall be transferred to the output. In addition to that, low power consumption and a fast operating frequency are also points of interest, although it is clear that a trade-off between these factors is not avoidable.

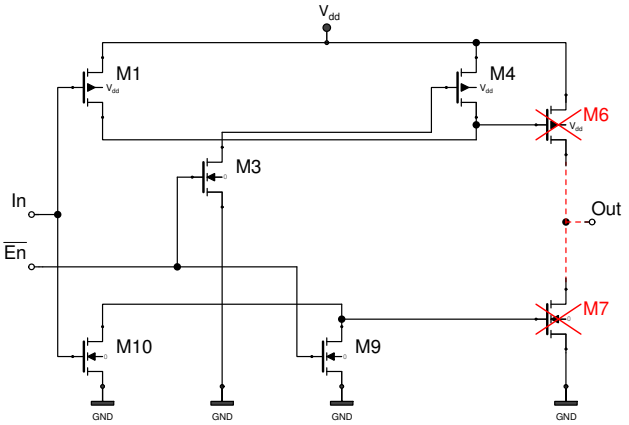


Figure 21. Decoupling of the output inverter in *highZ* mode

In the next step, the alternative tristate buffer was simulated under the same conditions like the previous circuits and the simulation curves are shown in Figure 22.

Figure 22 proves the correct function as the circuit reacts correctly on the applied input data. Similar to the previous investigations, the average, maximum and minimum power dissipation was extracted from the output curves and is summarized in Table IX.

In comparison to the results of the modified LP tristate buffer shown in Table V and Table VI, the newly implemented

TABLE IX. SIMULATION RESULTS (PWR and (I_{Vdd}))

Design type	Av. PWR (μW)	Max. PWR (μW)	Min. PWR pW
AT tristate	449.1	80.69	26.74

AT tristate buffer shows a poorer performance, at least when talking about a preferably low-power operation mode. The reason for this drawback is the lack of additional power reduction measures, which have been applied to the predecessors. Of course, all of these measures could have been added here as well but the basic intention was to design a completely new tristate buffer, which does not exceed a comparable number of transistors. The transient behavior of the AT tristate buffer is shown in Figure 23 and Figure 24, which depicts the outputs curves with an extended duration of the simulation time. The extracted numbers are shown in Table X and underline the previous statements about the low-power capabilities of this design. A better mitigation of dissipated power and leakage current could be achieved by several add-on measures like power gating, multi-V_{th}, multi-V_{dd} and multi T_{ox}. If desired these modifications can be implemented into the AT buffer by accepting the drawbacks described earlier.

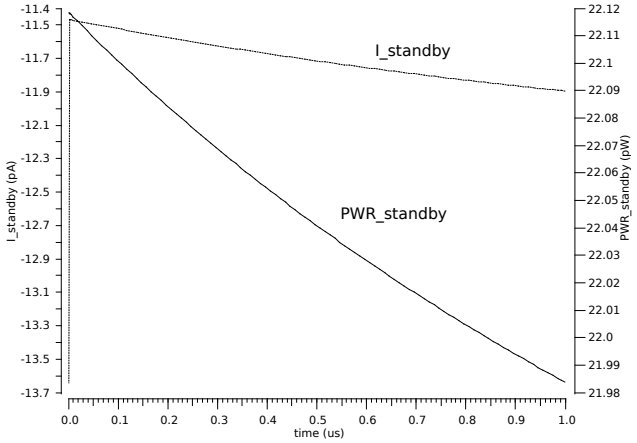


Figure 23. Transient analysis of the AT tristate buffer

After finishing the evaluation of the AT tristate buffer's energy consumption, the focus was put on investigating the

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

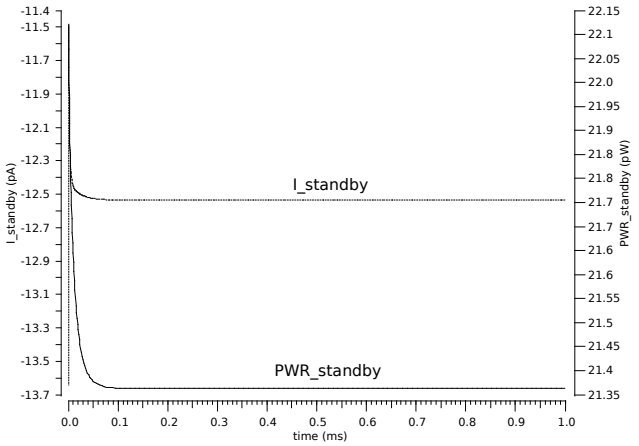


Figure 24. Extended simulation time of power dissipation and leakage current in power-off mode of AT tristate buffer

TABLE X. STANDBY RESULTS COMPARISON

Buffer type	Mod. LP	AT
Av. PWR (*pW*)	1.61	22.04
Av. Current (*pA*)	1.877	11.7

highZ behavior. For this purpose, the circuit was simulated with an active and inactive input in order to check how this might impact the output impedance. The related output curves are shown in Figure 25.

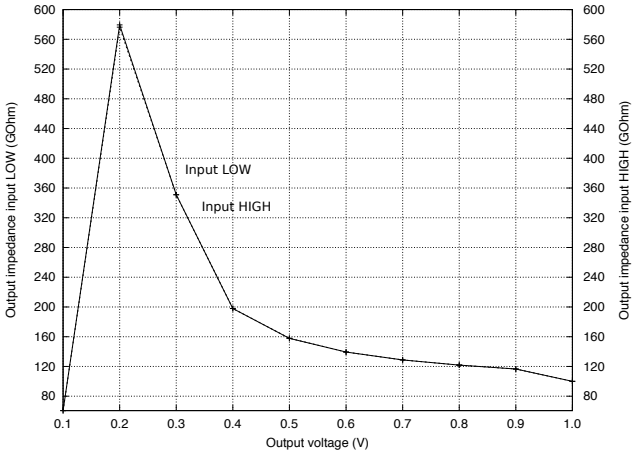


Figure 25. Output impedance in highZ state (AT tristate buffer)

On the first glance, Figure 25 shows that the output curves almost match perfectly on each other. This is a desirable effect as the same output impedance is provided by the circuit regardless of the applied input. Strong, separate driving forces to cut-off each of the output transistors $M6$ and $M7$ are shown in Figure 21. This effect could be even reinforced by using deep cut-off measures when applying a negative V_{GS} voltage at the gates. However, the out-of-the-box result is still considerable.

V. Results comparison

For a better comparison of the investigations, which have been done, all results were summarized in Table XI. The low power tristate buffer outperforms in almost each aspect the reference design, which highlights its appropriateness for use in applications with limited energy resources. Results of dynamic behavior show that power dissipation is reduced significantly, no matter whether the average, maximum or minimum power consumption is in focus of discussion. The most remarkable reduction is allocated to static behavior of both circuits. Here, the standby leakage current and the dissipated power in idle mode are lowered by over 80%, which emphasizes the effect of implemented low power measures.

TABLE XI. SIMULATION RESULTS I_{Vdd}

Design type	*Reference buffer*	*Low power buffer*	$\Delta\%$
Av. PWR (nW)	245	191.3	22 $\downarrow$
Max. PWR (uW)	56.75	29.72	47.63 $\downarrow$
Min. PWR (pW)	103.8	22.36	78.46 $\downarrow$
Av. Current (nA)	215.2	225.8	5 $\uparrow$
Max. Current (uA)	230.5	194.8	15.49 $\downarrow$
Av. Leak. Current (pA)	133.6	24.1	82 $\downarrow$
Av. Standby PWR (pW)	132.1	22.04	83.32 $\downarrow$
highZ max.	12.38 $G\Omega$	81 $G\Omega$	$\uparrow\uparrow$
highZ min.	6.85 $G\Omega$	18 $k\Omega$	$\uparrow\uparrow$

The appropriate choice of process technology due to the multi-oxide requirements as well as careful layout of transistor parameters requires special attention and allows additional improvements. However, the low power tristate buffer delivers remarkable out of the box performance without further detailed optimization. These adaptions are achieved with a small penalty in terms of transistor count and area. Xilinx provides 372 *Maximum User I/O* and 165 *Maximum Differential I/O Pairs* [7], which could be realized in 537 GPIOs. Implementing a new FPGA design by usage of the low power tristate buffer requires 1074 additional transistors. Here it comes to the point where an efficient layout of the overall chip could be a measures to catch up this drawback. In addition to that, the achieved minimum *highZ* state of the new design of about $14k\Omega$ does not perform as good as the result of the reference tristate buffer ($6.85G\Omega$). This could be improved by a further optimization of the transistor parameters in terms of length and width. However, this might lead to a higher energy consumption and should be carefully decided case by case, depending on which characteristic is of higher importance for the respective application. Despite the additional parasitic capacitances, which come along by adding transistors, nearly all measured insights does not weaken the positive overall print.

VI. Conclusion

We analyzed an existing design of a tristate buffer, which was baselined as a reference design serving for further comparisons. During the analysis we did a deep dive into the characteristics of this design for the evaluation of its active and standby performance in terms of dissipated power, average current consumption and the special ability to enter a *highZ* mode. The outcome of these activities was that we wanted to develop a tristate buffer, which is superior in terms of energy savings during runtime and idle state. Due to the lack of a clock signal and, therefore, the impossibility to apply clock gating, we implemented power gating by choosing special high T_{ox} transistors. These transistors have the ability to decouple the tristate buffer from V_{dd} and *GND* as well as the function of gate

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

tunneling mitigation. For subthreshold current reduction we decided to use high V_{th} transistors, being aware of accepting a penalty in the maximum operating frequency, which was not in the focus of our work though. Simulations have shown that the low power tristate buffer delivers outstanding performance in terms of, e.g., average power consumption in active mode, which is decreased by 22% compared to the reference design. This is a noticeable improvement, since it shrinks the losses of energy in active mode of almost a quarter compared to the reference design. In standby mode, our design outperforms the legacy design by 82% related to average I_{leak}, which is a remarkable result. This low power design features the ability to provide the generation of an internal, smaller supply voltage without any extra *enable* signal from external circuitry. The *highZ* mode abilities of the legacy design are better by a higher order of magnitude, nevertheless, we consider the achieved results of the new tristate buffer as acceptable. These results come at the cost of a higher transistor count and an additional input for an internal, decreased supply voltage I_{Vddlow} as an option. Several possibilities exist for future investigations and improvements. A very simple but effective method for achieving remarkable power savings would be the choice of a technology library with shorter channel lengths, e.g., $28nm$. A technology shrink usually leads to a measurable reduction of consumed power, however, this comes along with some drawbacks like the short-channel effect. Applying negative V_{GS} voltages is an effective way to suppress subthreshold leakage currents after turning a transistor off. Of course, this requires auxiliary logic for generation of negative V_{GS} gate voltages, but this should be a small amount of additional transistors. Further supporting measures can be applied at a higher hierarchical layer, e.g., at architectural level. Controlled dynamic voltage scaling offers the potential to drive the whole. circuit into a deep sleep mode if a standby mode is not crucial for operation of the whole logic. These suggested measures will be starting points for further elaboration of enhanced energy balance with strong focus on a extended battery lifetime in mobile applications.

ACKNOWLEDGMENT

The authors thank Pierre Mayr, from Ruhr University of Bochum, for giving advice on the aspects of correct signal transmission for bus architectures. We are grateful to Andreas Ullrich, from University of Wuppertal, for his continuous support in tool maintenance.

REFERENCES

[1] K. Niewiadomski and D. Tutsch, “Low power tristate buffer for mobile applications,” in Proceedings of the 10th International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2018), February 2018, pp. 1–6.

[2] C. Maxfield, The Design Warrior’s Guide to FPGAs: Devices, Tools and Flows, 1st ed. Newton, MA, USA: Newnes, 2004.

[3] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital integrated circuits- A design perspective, 2nd ed. Prentice Hall, 2004.

[4] K. Niewiadomski, C. Gremzow, and D. Tutsch, “4t loadless srams for low power fpga lut optimization,” in Proceedings of the 9th International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2017), February 2017, pp. 1–7.

[5] K. Niewiadomski and D. Tutsch, “Low power charge recycling d-ff,” in The Tenth International Conference on Advances in Circuits, Electronics and Micro-electronics (CENICS 2017), September 2017, pp. 1–6.

[6] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, “A 90nm low-power fpga for battery-powered applications,” in Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays. ACM, 2006, pp. 3–11.

[7] XA Spartan-3A Automotive FPGA Family Data Sheet, Xilinx, 04 2011, rev. 2.0.

[8] S. Kim, K. Cho, and B. Min, “An efficient gpio block design methodology using formalized sfr description,” in 2011 International SoC Design Conference, Nov 2011, pp. 84–87.

[9] Y. Jianfei, L. Zhaolin, W. Chipin, Z. Qingwei, and C. Jiajia, “Design of a configurable general-purpose input/output with event-capture,” in 2010 Second Pacific-Asia Conference on Circuits, Communications and System, vol. 1, Aug 2010, pp. 335–338.

[10] C. Panigrahi, M. Rastogi, and K. Gopal, “Leakage optimization of thick oxide io/esd transistors in 40nm global foundry process,” in 2017 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Oct 2017, pp. 84–87.

[11] K. Otsuka, F. Fujii, Y. Akiyama, and K. Hashimoto, “Io interface for over 25gbps operation with low power,” in IEEE CPMT Symposium Japan 2014, Nov 2014, pp. 83–86.

[12] ——, “Low power transmitter/receiver circuit for over 25gbps operation,” in 2014 9th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT), Oct 2014, pp. 185–188.

[13] C. Piguet, Low-power processors and systems on chips. CRC Press, 2005.

[14] C. Piguet, J.-M. Masgonty, V. von Kaenel, and T. Schneider, “Logic design for low-voltage/low-power cmos circuits,” in Proceedings of the 1995 International Symposium on Low Power Design, ser. ISLPED ’95. New York, NY, USA: ACM, 1995, pp. 117–122. [Online]. Available: http://doi.acm.org/10.1145/224081.224102

[15] A. Bellaouar and M. I. Elmasry, Low-Power Digital VLSI Design Circuits and Systems, 1st ed., J. Allen, Ed. Norwell, MA, USA: Kluwer Academic Publishers, 1995.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

www.iariajournals.org

International Journal On Advances in Intelligent Systems
issn: 1942-2679

International Journal On Advances in Internet Technology
issn: 1942-2652

International Journal On Advances in Life Sciences
issn: 1942-2660

International Journal On Advances in Networks and Services
issn: 1942-2644

International Journal On Advances in Security
issn: 1942-2636

International Journal On Advances in Software
issn: 1942-2628

International Journal On Advances in Systems and Measurements
issn: 1942-261x

International Journal On Advances in Telecommunications
issn: 1942-2601