International Journal on Advances in Systems and Measurements

2015 vol. 8 nr. 1&2

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

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Systems and Measurements
Volume 8, Number 1 & 2, 2015

Mohamed Eltoweissy, Pacific Northwest National Laboratory / Virginia Tech, USA
Paulo Felisberto, LARSyS, University of Algarve, Portugal
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
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, VTT Technical Research Centre of Finland, 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
Konandur Rajanna, Indian Institute of Science, India
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
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

International Journal on Advances in Systems and Measurements
Volume 8, Numbers 1 & 2, 2015

CONTENTS

A Design Framework for Developing a Reconfigurable Driving Simulator

Bassem Hassan
Project Group Mechatronic Systems Design
Fraunhofer Institute for Production Technology IPT
33102 Paderborn, Germany
Bassem.Hassan@ipt.fraunhofer.de

Jürgen Gausemeier
Heinz Nixdorf Institute
University of Paderborn
33102 Paderborn, Germany
Juergsen.Gausemeier@hni.uni-paderborn.de

Abstract - **Driving simulators have been used successfully in various application fields for decades. They vary widely in their structure, fidelity, complexity and cost. Nowadays, driving simulators are usually custom-developed for a specific task and they typically have a fixed structure. Nevertheless, using the driving simulator in an application field, such as the development of the Advanced Driver Assistance Systems, requires several variants of the driving simulator. Therefore, there is a need to develop a reconfigurable driving simulator, which allows its operator to easily create different variants without in-depth expertise in the system structure. In order to solve this challenge, a design framework for developing a Task–Specific reconfigurable driving simulator has been developed. The design framework consists of a procedure model and a configuration tool. The procedure model describes the required development phases, the entire tasks of each phase and the used methods in the development. The configuration tool organizes the driving simulator's solution elements and allows its operator to create different variants of the driving simulator by selecting a combination of the solution elements, which are like building blocks. The design framework is validated by developing three variants of a reconfigurable driving simulator. This paper includes a modified procedure model, more detailed analysis of the state of the art and new results comparing with the previous published paper "Concept for a Task–Specific Reconfigurable Driving Simulator".**

Keywords - Advanced Driver Assistance Aystems (ADAS); reconfigurable driving simulator; confiuration mechanis; solution elements; bulding blocks; variants

I. INTRODUCTION

The development and testing of the in-vehicle systems, such as Advanced Driver Assistance Systems (ADAS), is a challenge due to their complexity and dependency on the other vehicle systems, initial conditions, and the surrounding environment [1] [2]. The testing of ADAS in reality leads to significant efforts and cost. Therefore, virtual prototyping and simulation are widely used instruments in the development of such complex systems [3].

Virtual prototyping is well-established in facilitating the development of new vehicle systems and components [4]. It is the process of building, simulating, and analyzing virtual prototypes. Virtual prototypes are the digital representations (models) of the real prototypes. It allows the verification of the properties and the functions of the product in the early development phases without having to build a real prototype. This saves time and costs [5]. One of the most useful virtual prototyping tools in the automotive field are driving simulators.

Driving simulators allow the ADAS developer to investigate the interaction between the human driver, the Electronic Control Unit "ECU" virtual prototype and the vehicle, while the human driver steers a virtual vehicle in a virtual environment. Driving Simulators rank among the most complex testing facilities used by automotive manufacturers during the development process. They are based on close collaboration of different simulation models at runtime [6]. These partial models represent dedicated aspects of the different vehicle components, as well as the vehicle environment [7].

Driving simulators vary in their structural complexity, fidelity and their cost. They range from simple low-fidelity, low-cost driving simulators such as computer-based driving simulators to complex high-fidelity, high-cost driving simulators such as high-end driving simulators with complex motion platforms [8].

Nowadays, existing driving simulators are usually task-specific devices, which are individually custom-developed by suppliers for a specific usage during the ADAS development. For example, a task-specific driving simulator is typically used for testing the ADAS main functionality without considering the human-machine-interfaces and another task-specific driving simulator is used for investigating different variant of human-machine-interfaces. These driving simulators can only be configured by a driving simulator expert. This is done by exchanging one or more of their entire components. Existing driving simulators do not allow their operator to change the system architecture or to exchange simulation models without in-depth knowledge of the driving simulator's components and structure.

The development of a driving simulator is a costly and complex task; the testing and training of ADAS often requires more than one configuration of a driving simulator. That is why there is a need for developing a reconfigurable driving simulator that allows the system operator to reconfigure it in a simple way without in-depth expertise in the system.

This work is based on a previous paper of the authors "Concept for a task–specific reconfigurable driving Simulator" [1]. However, this paper describes a modified procedure model, more detailed analysis of the state of the art, and presents the new reached results in more details.

II. Reconfigurable Driving Simulators Definition

In most of existing driving simulators' descriptions or brochures, they are defined as a "reconfigurable driving simulator". Therefore, the term "reconfigurable driving simulator" has to be clearly-defined with the help of three questions: "Which driving simulator components could be reconfigured?", "Who can reconfigure the driving simulator?" and "What is the difference between a configurable and a reconfigurable driving simulator?" Based on the answers of the questions, the term "Reconfigurable Driving Simulator" will then be defined.

Which driving simulator components could be reconfigured? The term "reconfigurable driving simulator" is sometimes misused instead of using the term "driving simulator with exchangeable components" or the term "driving simulator with parameterized models". Driving simulators consist of various components. These components are classified into three categories: hardware, software, and resources. There are many driving simulators which have exchangeable hardware components, e.g., vehicle mock-up, motion platform, and visualization system. Other driving simulators have exchangeable software components, e.g., vehicle model, traffic model, etc. Most driving simulators have parameterized simulation models, e.g., a parameterized vehicle model to simulate different vehicle types, parameterized traffic models to simulate different traffic scenarios, etc.

Who can reconfigure the driving simulator? The term "reconfigurable driving simulator" is sometimes misused instead of using the term "modular driving simulator" or "configurable driving simulator". Many driving simulators could be customized individually by their manufacturer according to the customer requirements. These are "modular driving simulators". Some driving simulator components could be exchangeable or some components could be added or removed. These are configurable driving simulators, which can be reconfigured or upgraded only by their manufacturer or developer.

What is the difference between a configurable and a reconfigurable driving simulator? A configurable driving simulator means that a variant of a driving simulator could be created by selecting its entire components during the development, but its structure and/or its entire components cannot be changed after the development. However, a reconfigurable driving simulator structure and entire components can be changed after the development. In this paper, we describe a reconfigurable driving simulator development approach in means of, adding, removing, modifying, and resampling the components of the driving simulator is granted after the development.

Reconfigurable driving simulator definition: A driving simulator is reconfigurable when different configurations can be used optimally in different tasks at different times. The reconfiguration should be feasible by the operator without in-depth expertise in the system structure. The operator can create different configurations by changing the system structure (adding or removing some of its entire components) and by exchanging the entire system components with other suitable components.

III. Related Work

There are thousands of driving simulators spread all around the globe. They are complex mechatronic systems and include different technologies, which widely range from computer graphics to controlling a complex motion platform. The publications about driving simulators usually take one technology into consideration or just a partial aspect of developing a specific driving simulator. The state of the art in this section will only consider the publications that are related to the development methods of driving simulators and the previous approaches towards developing a reconfigurable driving simulator.

This section surveys an existing driving simulator selection method and previous approaches towards developing a reconfigurable driving simulator.

A. The Driving Simulators Selection Method according to Negele[6]

Negele developed a method called the "Application Oriented Conception of Driving Simulators for the Automotive Development". He considered driving simulators as one of the most complex test rigs used in the automotive development. The development of a driving simulator requires a wide expertise in different technologies and disciplines, which widely range from the visualization techniques to platform motion control. This essential know-how is not in the core competence of the automotive manufacturer. Therefore, driving simulators, which are used as automotive test rigs, are usually developed by driving simulator suppliers. Nevertheless, it is tough for automotive engineers, who do not have a basic knowledge of driving simulator technologies to select and specify a driving simulator that fits with a specific-task [6].

Therefore, Negele developed a method, which allows automotive engineers to formulate the requirements and specifications of a driving simulator for a specific application. The main objective of the method is to define the relationships between the automotive applications and driving simulators' specification [6].

Automotive engineers could select a driving simulator type based on two main criteria: a driving task category and a driver stimulus-response mechanism, according to the application of the required driving simulator.

The driving tasks are categorized into primary tasks, secondary tasks, and tertiary tasks. The primary tasks consist of vehicle navigation, vehicle guidance and vehicle stabilization. The driver stimulus-response mechanisms are categorized into the following: skills-based responses, which are senso-motoric responses (e.g., acceleration or steering), rule-based responses (e.g., driving slower in a curve) and knowledge-based responses (e.g., route planning with the help of paper maps) [6].

The driving simulator application should be defined by means of the following: a driving task category (Which driving tasks should be investigated?) and a driver stimulus-response mechanism (Which driver stimulus-response

mechanism is relevant?). For example, if the driving simulator application is the testing of vehicle dynamics, then the application is focusing on a primary driving task (vehicle stabilization) and investigating a skills-based response of the vehicle driver [6].

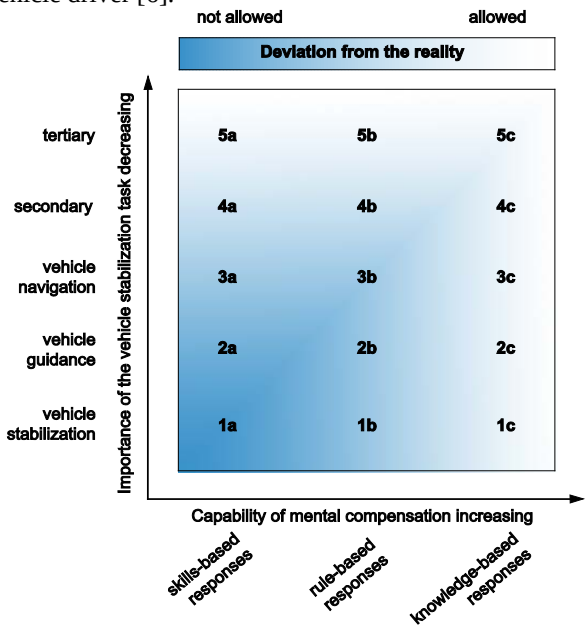


Figure 1. Scheme for classifying driving simulator applications [6].

Fig. 1 shows the intersections matrix between the five driving tasks categories: (vehicle stabilization, vehicle guidance, vehicle navigation, secondary tasks, and tertiary tasks) and the three driver stimulus-response mechanisms: (skills-based responses, rule-based responses, and knowledge-based responses). These result in 15 types of driving simulators, which are marked from 1a to 5c [6].

Each driving simulator type is described by a profile table. The profile table specifies the entire components of the driving simulator variant. Negele divided the simulator into 26 components grouped into 6 groups.

The method of Negele allows automotive engineers to formulate the requirements and the specifications of a task-specific driving simulator. The focus was on how to specify the requirements of a driving simulator to fit with a specific task. He did not consider the reconfigurability of driving simulators and he did not mention a driving simulator's development method.

Nevertheless, the method is useful as a preliminary work for driving simulator operators. They can use Negele's method to specify the preferred driving simulator's requirements and its entire components, then they can use the design framework described in this work in order to create a specific driving simulator variant.

B. *Existing Low-Level Driving Simulators*

Low-level driving simulators have restricted fidelity, high usability and they are usually low-cost driving simulators. Typically, they have a single display that provides a narrow horizontal field of view and a gaming steering wheel as a Human-Machine-Interface (HMI) [9].

The following sections describe one previous approach towards developing low-level reconfigurable driving simulator.

A Modular Architecture based on the FDMU Approach: Filippo et al. had developed "a modular architecture for a driving simulator based on the FDMU approach". This approach describes a modular and easily configurable simulation platform for ground vehicles based on the Functional Digital Mock-Up approach (FDMU). FDMU is a framework developed by the Fraunhofer Institute. The framework consists of a central component called "Master Simulator", which connects different components through an application called "Wrapper". Each module communicates with the master simulator through its own wrapper application and a standardized Functional Building Block (FBB) interface. Fig. 2 shows the basic scheme of the FDMU architecture [10].

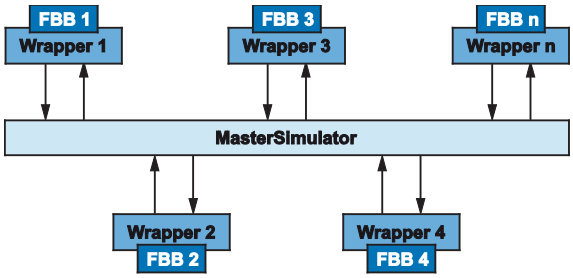


Figure 2. Basic scheme of FDMU architecture [10].

Filippo et al. [10] had developed a driving simulator based on the FMDU architecture. This driving simulator consists of two hardware components and two software components. The hardware components are a motion platform, which is an off-the-shelf Steward platform, and an input device, which is an off-the-shelf Universal Serial Bus "USB" steering wheel and pedals. The software components are the master simulator simulation core and a simple vehicle model implemented with the help of OpenModelica, which is an open-source modeling and simulation environment [10].

The developed approach: "A Modular Architecture for a driving simulator based on the FDMU Approach" focusses on the interfacing of the different components of the driving simulator with the help of an FMDU modular structure. The problem with this approach is that in order to add or exchange any component, a wrapper application has to be reprogrammed or adjusted for the new component. The approach does not describe how to add, remove or exchange any of the four pre-programmed components. Indeed, the approach is promising for simulation core components, which interface the driving simulator components with each other. But it could not be used in a reconfigurable driving simulator without some enhancements, e.g., the master simulation has to be dynamically adjustable depending on the connected modules without being pre-programmed by the user.

C. *Existing Mid-Level Driving Simulators*

Mid-level driving simulators have a greater fidelity than the low-level driving simulators, as well as high usability.

Typically, they have multi-displays, which provide a wide horizontal field of view, a real vehicle dashboard as an HMI, and they are sometimes equipped with a simple motion platform [9].

The following section describes one previous approach towards developing reconfigurable mid-level driving simulator.

The University of Central Florida Driving Simulator: The University of Central Florida (UCF) driving simulator is operated in the Centre of Advanced Transportation Systems Simulations (CATSS). It has evolved since the late 1990's into a mid-level driving simulator with the aim of conducting research in transportation, human factors and real-time simulation. The UCF driving simulator is equipped with a hexapod motion platform with 6 DoF. It has a passenger vehicle cabin as an input device. The vehicle cabin is mounted over the motion platform. The UCF has a visualization system that consists of 5 displays: one for the front view, two for side views and two for the left and middle rear mirrors. The simulator is also equipped with an audio system, force feedback steering wheel and the main operator console [11]. The simulator was designed with an exchangeable vehicle cabin. The user can choose from a commercial truck cabin and a passenger vehicle cabin according to the test requirements. The vehicle model could also be changed according to the used vehicle cabin [11].

The UCF driving simulator has exchangeable driving cabins and exchangeable vehicle models. It could be configured according to the customer requirements by choosing from the passenger car cabin with its respective vehicle model or the commercial truck cabin with its respective vehicle model. The UCF driving simulator is not a reconfigurable driving simulator because only the driving cabin and vehicle model are exchangeable. Moreover, the driving simulator user cannot exchange the entire components or add a new component to the system without the help of the manufacturer.

D. Existing High-Level Driving Simulators

High-Level driving simulators have great fidelity, high usability and they are high-cost driving simulators. Typically, they almost have a 360 degrees horizontal field of view and a complete real vehicle as an HMI, which is mounted on a high-end motion platform with at least 6 degrees of freedom [9].

The following section describes one previous approach towards developing reconfigurable high-level driving simulator.

Daimler Full-Scale Driving Simulator: Daimler AG inaugurated the Daimler full-scale driving simulator in October 2010 in Sindelfingen, Germany. The Daimler full-scale driving simulator is used mainly in developing new ADAS and the evaluation of different vehicle dynamics concepts. It is equipped with a 7 DoF motion platform that consists of the following two parts: the lateral 12 m long rail system, which provides linear motion in Y-direction and a hexapod which provides 6 DoF. The dome of Daimler full-scale driving simulator has a diameter of 7.5 m, which can be moved by a rail system for 12 m (in X or Y directions) and by the hexapod as follows: +1.4 to -1.3 m in X-direction, ±1.3 m in Y-direction, and ±1 m in Z-direction, ±20 degrees roll-rotation, -19 degrees to +24 degrees pitch-rotation and ±38 degrees yaw-rotation.

The Daimler full-scale driving simulator has a cylindrical visualization system powered by 8 projectors and gives 360 degrees horizontal field of view and three rear mirrors displays. It has several exchangeable driving cabins, e.g., S-Class, A-Class, Actros-Truck, etc. It is operated by a Daimler in-house developed software. The used software can also operate Daimler internal fixed-base driving simulator variants [12].

The Daimler full-scale driving simulator has exchangeable driving cabins and a parameterized vehicle model. It could be configured according to the test experiment requirements by choosing from different driving cabins and their respective vehicle model parameter set. The Daimler full-scale driving simulator is not a reconfigurable driving simulator because the driving simulator components are only compatible with Daimler internal components. The driving simulator user cannot exchange the entire components or add a new component to the system without the help of the manufacturer.

E. The National Advanced Multi-Level Driving Simulators

The multi-level driving simulators are different variants of a driving simulator as they have different levels of fidelity, usability and cost. But they are developed based on the same structure using the same software, hardware, and resources components. An example of the multi-level driving simulator is the NADS driving simulator, which is described in this section.

The National Advanced Driving Simulator (NADS) is a driving simulator centre located at the University of Iowa. The NADS centre has three driving simulators: the high-level driving simulator “NADS-1”, the mid-level driving simulator “NADS-2”, and the low-level driving simulator “NADS miniSim”. The NADS driving simulators are based on the same system architecture, software, and resources [13].

The NADS-1 and NADS miniSim driving simulators are modular driving simulators, which have been developed based on the same software components. They could be configured for different applications according to the customer specifications. The NADS minSim is a low-level configurable driving simulator. It is a promising approach towards developing a reconfigurable driving simulator. However, it is not a reconfigurable driving simulator, because as well-developed as it is, the user cannot exchange the entire components or add a new component to the system without the help of the manufacturer.

The analysis of the existing methods and approaches towards a reconfigurable driving simulator has shown that there is no method, approach or developed driving simulator to date which describes any systematics or approaches for the development of a reconfigurable driving simulator and none of them allows the operator of the driving simulator to reconfigure the system without in-depth expertise in the system structure.

IV. The Solution Approach

The main aim of this work is to simplify a driving simulator structure during the development. This simple structure allows the operator to create different task-specific variants by selecting the desired solution elements of the driving simulator.

The development of reconfigurable mechatronic systems, which consist almost of standardized modular components, can follow the "Building Blocks Concept". The benefits of using the building blocks concept are speeding up the learning curve of the system structure based on the many years of experiences in the development of their entire components [14].

The typical virtual prototyping cycle consists of three phases: modelling, simulation and analysis. The modelling process is the developing of simplified formal models of the system under development. The system models represent the system properties. The simulation process represents the calculations of the system models with the help of numerical algorithms in order to simulate the system behaviour. The analysis process represents the interpretation of the simulation results that are usually done by extracting, preparing and visualizing the relevant information [5] [15]. The usage of driving simulators allows ADAS developers to analyse the system under test functionality, the system behaviour in different simulation scenarios as well as the investigation of the interaction between the system, driver, and environment.

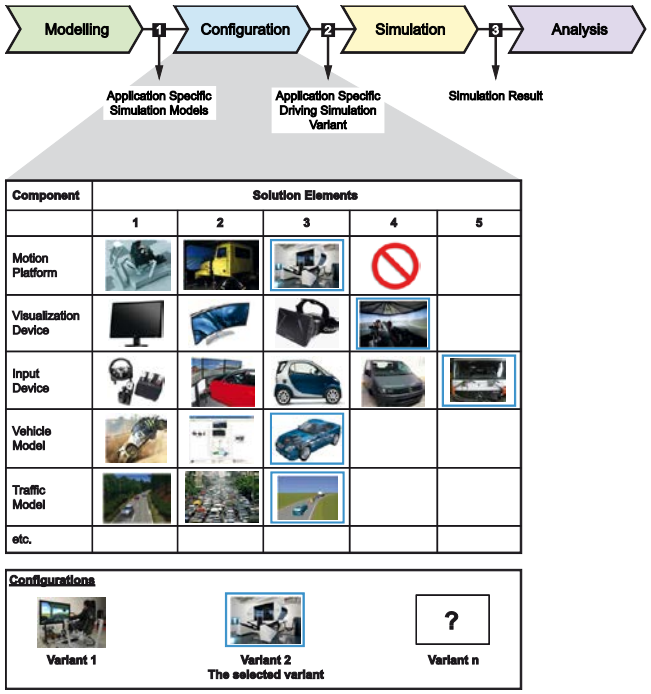


Figure 3. The solution approach of the reconfigurable driving simulator, according to the building blocks concept.

In order to reconfigure a driving simulator, there is a need to add a phase between the modelling and simulation phases. The new phase is the configuration phase shown in Fig. 3. In the configuration phase, the driving simulator operator can select the desired solution elements to create a task-specific variant of the driving simulator.

The models that have been developed during the modelling phase will be available for the selection in addition to other existing components. The operator selects a solution element for each component. These selected solution elements, acting as building blocks, build together a driving simulator variant. Fig. 3 shows a simplified example of the configuration process; the selected solution elements and the created variant are marked with a blue frame. As soon as a variant has been created, the driving simulator will be ready for the simulation and the analysis phases.

V. The Design Framework

This section is the core of the present work. It describes a design framework for developing a reconfigurable driving simulator. This design framework supports driving simulator developers and operators to develop and operate a reconfigurable driving simulator. The design framework consists mainly of the procedure model and the configuration tool. They are specifically described as follows:

- **The procedure model**, which defines the required phases in a hierarchy, in order to develop a reconfigurable driving simulator. Each phase contains entire tasks; these tasks have to be carried out in order to achieve the phase objectives. The procedure model organizes the required tasks in each phase and describes which method or algorithm should be used to fulfill each task. The used methods and algorithms contain existing approaches, as well as new approaches, which were developed during this work. Moreover, the procedure model defines the result of each phase. This is needed as an input for the following phases.
- **The configuration tool**, which supports the driving simulator operators in creating a driving simulator variant or in reconfiguring an existing variant. The configuration tool organizes the existing driving simulator software and hardware components and their corresponding solution elements in a solution elements database. As soon as the solution elements database is filled, the software guides the driving simulator operator in order to create the desired driving simulator variant. The variant creation will be done by selecting a combination of solution elements, which are available in the database. Moreover, the configuration tool can deal with guidelines for testing and/or for training approaches. They can be added to the tool, and the configuration tool can check whether the created variant guideline conforms or not.

Fig. 4 describes a design framework for developing a Rreconfigurable driving simulator. This design framework supports driving simulator developers and operators to develop and operate a reconfigurable driving simulator.

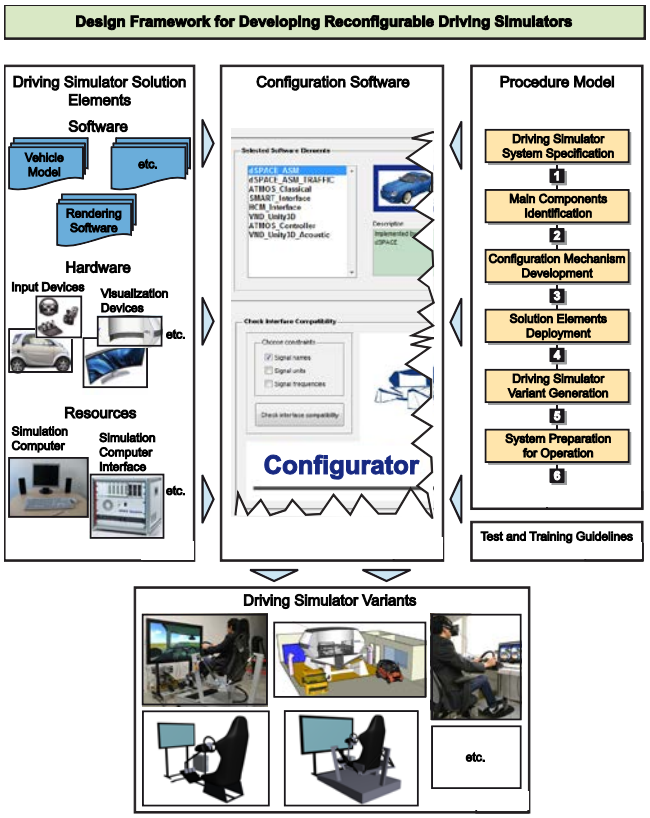


Figure 4. A design framework for developing a reconfigurable driving simulator structure and components.

Procedure Model Overview: the procedure model is the most essential part of the design framework; it describes the theoretical fundamentals of the design framework. The procedure model supports driving simulator developers in the development of a reconfigurable driving simulator. The procedure model is kept general and could be used for different driving simulator areas of use, as well as other mechatronic systems. It consists of six consequent phases divided into two stages. Fig. 5 shows the procedure model in the form of a phases/milestones diagram that shows each phase. It also shows the tasks that have to be carried out, as well as the results from each phase.

The six phases of the procedure model are generally divided into two stages: The system development stage and the variants creation stage. Each stage consists of three phases. The first three development phases have to be performed once by the driving simulator developer. As soon as the developer finishes the development phases, the driving simulator operator should carry out the variant creation phases each time he/she creates a driving simulator variant.

In the following sections, a detailed description of all needed tasks and operations during each phase, as well as the results of each phase, will be presented.

A. *Phase 1 – Driving Simulator System Specification*

The objective of the first phase is to specify a reconfigurable driving simulator, which is a complex multidisciplinary mechatronics system. Therefore, there is a need to specify the system under a multidisciplinary development with the help of a specification technique.

The CONSENS – "*Conceptual Design Specification Technique for the Engineering of Complex Systems*" will be used during this work. CONSENS is developed in order to specify complex mechatronic systems. The specifications are multidisciplinary and they simplify the complexity of the developed mechatronic system by describing it using a coherent system of partial models [16].

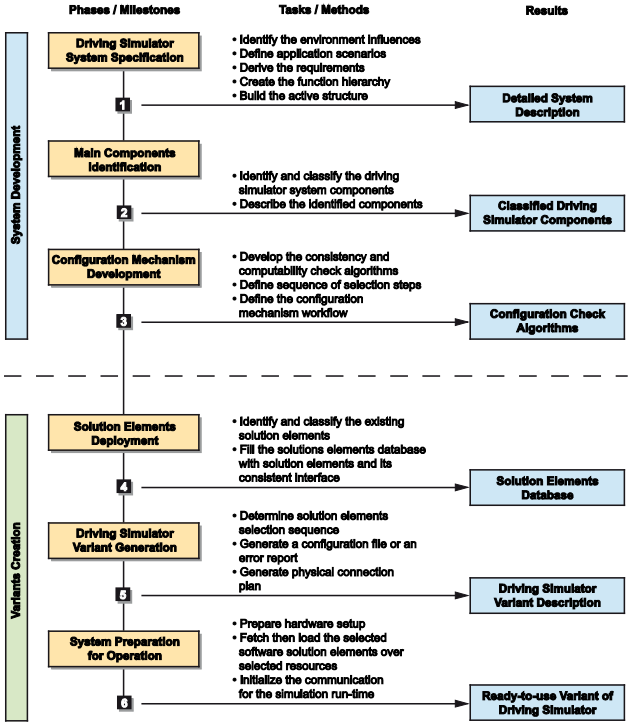


Figure 5. Procedure model for developing a reconfigurable driving simulator.

CONSENS Work Flow for a Reconfigurable Driving Simulator: the specification technique "CONSENS" divides the principle solution specification into coherent partial models. The CONSENS partial models are: requirements, environment, application scenarios, functions, active structure, shape, and behaviour. Each partial model specifies a precise aspect of the system under development [16].

The partial models' weights of importance are not equal within the development of reconfigurable driving simulators. During this work, the focus will be on five of seven CONSENS partial models. The relevant partial models are environment, application scenarios, requirements, functions, and active structure. The shape and behaviour partial models will be neglected within the scope of this work because they are not relevant to design a driving simulator. The both neglected partial models are important to design a new product.

The CONSENS work flow is divided into three steps: firstly, the environment, the application scenarios and the requirements have to be specified simultaneously. Secondly, based on the result of the first step, the function hierarchy

has to be derived. The third step is to build up the active structure based on the result of the previous steps. Fig. 6 shows the CONSENS work flow towards specifying a reconfigurable driving simulator.

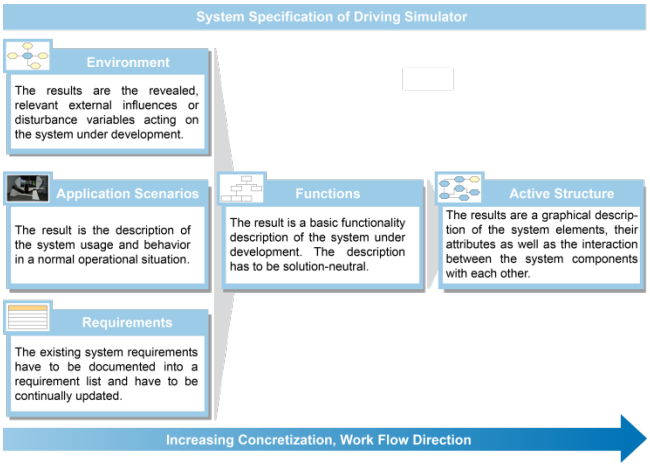


Figure 6. CONSENS work flow for reconfigurable driving simulator according to Gausemeier [17].

The specification of the system is typically carried out in the context of expert workshops with the help of a workshop cards set. The workshops' participants are usually experts in several disciplines such as mechanical engineering, software engineering, control engineering, and electrical engineering. The definition of each partial model is presented in the next sections.

1) Environment:

The environment partial model defines the external influences, which affect the system under development. The driving simulator has to be considered as a black box which means that the investigation is not of the system itself, but of the relevant external influences. These external influences are environment elements or disturbance variables [16].

Fig. 7 shows an environment model example of a driving simulator variant.

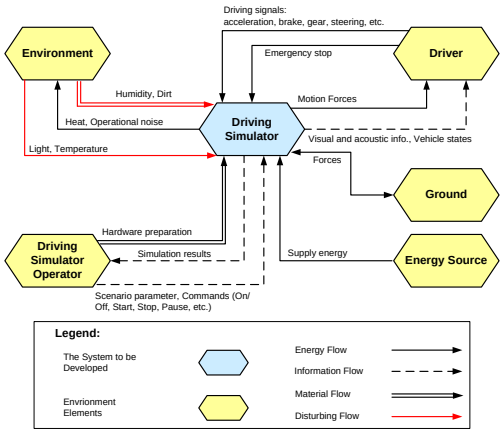


Figure 7. Environment model of a driving simulator variant.

2) Application Scenarios:

The application scenarios partial model is an essential partial model of the system specification. In this specification step, some operational application scenarios are defined. Each application scenario describes the system under development in terms of way of use, operation modes, system manner and main components. By using CONSENS, each application scenario will be described in a profile page, which contains the scenario title, scenario numbering, the scenario description and a simple sketch of the needed hardware components [16].

3) Requirements:

This partial model collects and organizes the system requirements of the system under development which need to be covered and implemented during the development process. The requirement list contains functional and non-functional requirements [16]. Additionally, the organized requirements distinguish between demands and wishes (D/W) [18].

4) Functions:

The functions partial model is built based on the previous partial models: environment, application scenarios and requirements. It describes the system and its entire components' functionality in a top-down hierarchy [16]. Each block describes a sub-function of the system. Function catalogues, according to Birkhoffer [19] or Langlotz [20], support the creation of the functional hierarchy.

Due to the variation of the main function, structure, and required components of the stated application scenarios, the functions specification also varies in its complexity and number of its entire sub functions. Therefore, there is a need to merge the identified functions of the stated application scenarios. Fig. 8 shows a function model example of a driving simulator variant.

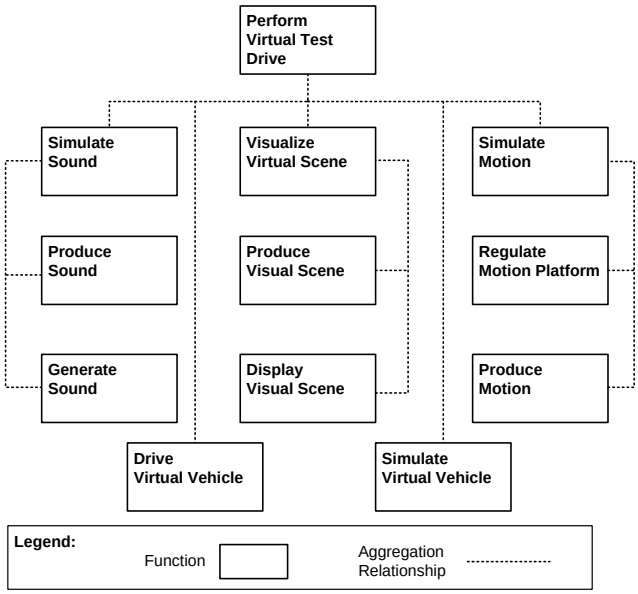


Figure 8. Function model of a driving simulator variant.

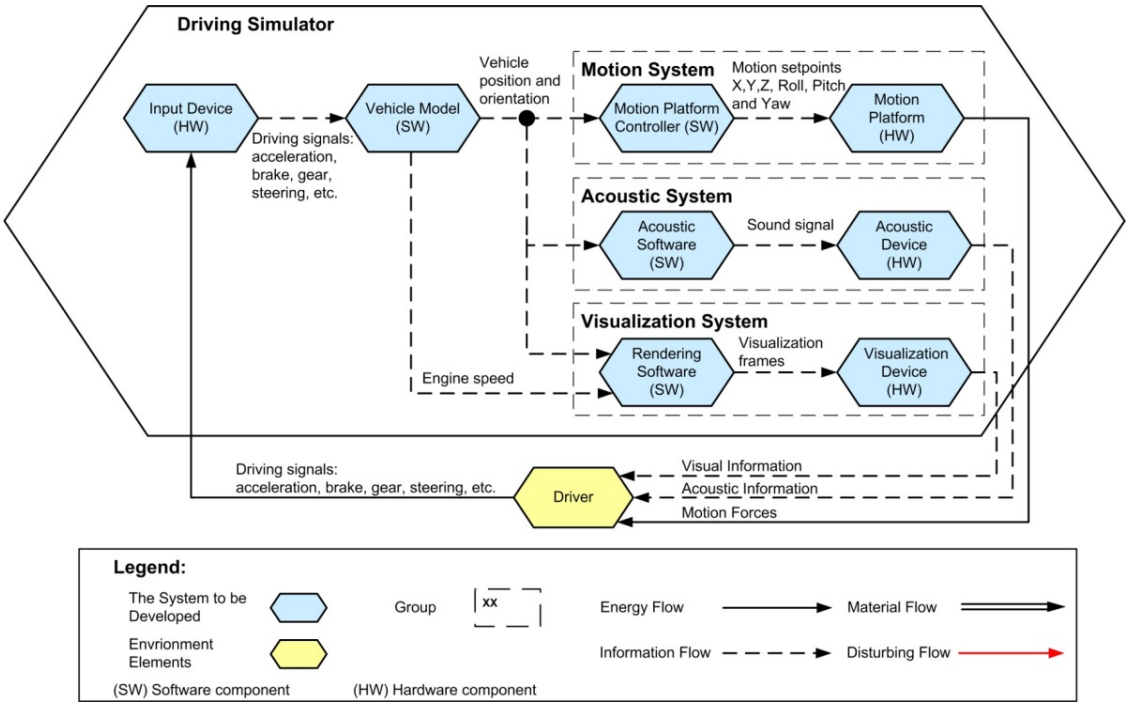


Figure 9. Active structure model of a driving simulator variant.

5) Active Structure

The active structure partial model is built based on the previous partial models results, specifically the functions partial model. The active structure describes the entire system in more details in the form of system component active principles. It describes the system components, their attributes, the entire interfaces and how the components interact with each other. Depending on the modeling level of details, each system element could be described abstractly as an active principle or a software pattern. Additionally, material, energy, and information flows, as well as logical relationships, describe the interactions between the system elements [16]. Fig. 9 shows an active structure model example of a driving simulator variant.

The first phase results, which are the driving simulator system specification describes in the form of five partial models, are: environment, application scenarios, requirements, functions, and active structure. This result is the input for the second phase.

B. *Phase 2 – System Components Identification*

The second phase objectives are the identification, classification and definition of the driving simulator components based on the results of the first phase. Towards the identification of the driving simulator system components, a distinction between optional components, key components and solution elements must be defined.

As the driving simulator structure could also be changed during the reconfiguration process, the key components have to be identified. The key components are the obligatory system components that always have to exist in the simulator structure. For example, each driving simulator has to have a visualization rendering software but a motion platform is an optional component and not a key component, because a driving simulator does not need to have a motion platform.

1) Identification of Driving Simulator Components

Based on the active structure partial model, the system components, as well as the system key components can be identified with the help of the following three operations:

1. Identify all components:

The reconfigurable driving simulator components are the union of the different variants components as follows:

$$Sim\_Cp = Var\_1\_cp \cup Var\_2\_cp \cup \dots Var\_n\_cp \quad (1)$$

Where: Sim_Cp is the reconfigurable driving simulator components, Var_1_cp is variant 1 components, Var_2_cp is variant 2 components, and n is the number of modelled variants.

For example, if variant 1 components are {A,B,C} and variant 2 components are {A,B,D,E}, and the common system components will be {A,B}.

2. Identify common components:

The common components of the reconfigurable driving simulator are defined based on the intersection between the different variants components as follows:

$$Sim\_Cp = Var\_1\_cp \cap Var\_2\_cp \cap \dots Var\_n\_cp \quad (2)$$

3. Identify key components:

In order to identify the system's key components, the selection will be done based on the common components set. Each component has to be investigated individually in a

logical way by eliminating the component from the set. If the driving simulator can be operated without this component, this means that it is an optional component. But if the driving simulator cannot be operated, then this means that it is a key component.

2) Classification of the Identified Components

In addition to the modelled software and hardware components, the reconfigurable driving simulator resources have to be taken into consideration. Each software or model needs a computing unit (e.g., a computer) to be executed on. Moreover, each hardware component needs a physical interface to communicate with its corresponding software interface.

In order to organize the identified components easily, these have to be classified under the following three categories: hardware, software, and resources. The software category contains two subcategories: the applications/models and the hardware interfaces. The resources category contains two subcategories: the computing units and the signal processing interfaces. Fig. 10 shows an example of the classification of the identified components.

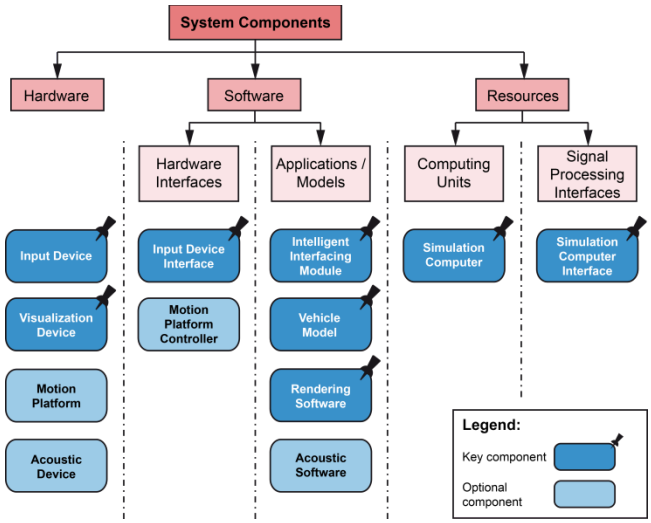


Figure 10. Classification of the identified components example.

3) Description of the Identified Components

In order to understand the function of each component, each component has to be defined from a solution-neutral point of view. The following are the description of two identified components as an example:

Input Device: This is a hardware MMI (Man-Machine Interface) between the driver and the driving simulator. It provides driving signals, e.g., acceleration pedal position, brake pedal position, etc. The input device provides the driving simulator with these signals in energy flow, which represents a physical signal.

Input Device Interface: This is a software component, which converts the energy flows of the input device to its computer representative information flows (digital signals).

C. *Phase 3 – Configuration Mechanism Development*

This is the third and last phase of the development stage. The objective of the third phase is to develop a configuration mechanism, which ensures that the selected solution elements could operate together. This check is done after selecting the preferred structure and the desired solution elements. The configuration mechanism has to ensure the consistency and the compatibility of the selected structure and its entire solution elements. After the configuration mechanism ensures the selected solution element consistency and compatibility of the solution elements, it generates a configuration file. The configuration file contains a list of the selected solution elements, the interfaces' topology and the selected resources.

The configuration mechanism checks the selected solution elements. However, the solution elements will be deployed in the next phase, but it is the preferred order of the procedure. Developing the configuration mechanism before deploying the solution elements allows the mechanism to also deal with unknown solution elements, which can be added in the future.

There are two types of relationships between the selected solution elements and each other. These relationships have to be checked and confirmed by the configuration mechanism. The first relationship is the logic consistency between the selected solution elements with each other. The second relationship is the compatibility between the interfaces of the selected solution elements.

1) Consistency Check Algorithm

The consistency relationship can be determined by two levels. The first level is the **logic dependency** between components, which determines if there is a logic correlation between two components or not. The second level is the **logic consistency** between two solution elements.

a) Logic dependency between two components:

It is a logic relationship between two components, which describes if they depend on each other logically or not. For example, the motion platform and the input device are a dependent pair of components. They depend on each other, i.e., an input device has to be mounted on a motion platform. Therefore, the motion platform dimensions and payload have to match with the selected input device.

Dependency matrix: the dependency matrix is a two-dimensional matrix that describes the logic dependency between the identified components. The components are stated in both the first row and the first column; the matrix is mirrored along its diagonal. Therefore, only the lower half of the matrix has to be filled with 0 or 1 by the **driving simulator developer**.

0: means the components pair is logically independent of each other, thus the inherited solution elements belonging to these components will also be logically independent of each other.

1: means the components pair is logically dependent on each other, thus the inherited solution elements belonging to these components will also be logically dependent on each other. Fig. 11 shows the dependency matrix based on the identified components.

Dependency Matrix 0 = Independent pair 1 = Dependent pair		Hardware Components				Software Components					Resources	
		Input Device	Visualization Device	Motion Platform	Acoustic Device	Vehicle Model	Rendering Software	Acoustic Software	Input Device Interface	Motion Platform Controller	Simulation Computer	Simulation Computer Interface
		A	B	C	D	E	F	G	H	I	J	K
Hardware	A. Input Device											
	B. Visualization Device	0										
	C. Motion Platform	1	1									
	D. Acoustic Device	0	0	0								
Software	E. Vehicle Model	0	0	0	0							
	F. Rendering Software	0	1	0	0	0						
	G. Acoustic Software	0	0	0	1	0	0					
	H. Input Device Interface	1	0	0	0	0	0	0				
	I. Motion Platform Controller	0	0	1	0	0	0	0	0			
Resources	J. Simulation Computer	0	0	0	0	1	1	1	1	1		
	K. Simulation Computer Interface	0	0	0	0	0	1	1	1	1	1	

Figure 11. Dependency matrix of the identified components.

b) Logic consistency between two solution elements

It is a logic relationship between two solution elements, which describes if they are logically consistent with each other or not. The first relationship depends on whether the solution elements' parent components are independent. This means that the two solution elements inherited the independence and there is no need to check their consistency. Otherwise, if the solution elements' parent components are dependent, this means that the two solution elements inherited the dependency and have to be checked if they are consistent or not.

Consistency matrix: the Consistency matrix is a two-dimensional matrix that describes the logic consistency between the available solution elements. The solution elements are stated in both the first row and the first column. The matrix is mirrored along its diagonal. Therefore, only the lower half of the matrix has to be filled with 0, 1 or 2 by the reconfigurable driving simulator operator.

0: means the solution elements pair is logically inconsistent with each other. This means that they could not be selected together in a driving simulator variant.

1: means the parent components pair was originally logically independent of each other, thus the inherited solution elements under those components will also be logically independent of each other. This means that the solution elements do not have to be checked for consistency.

2: means the solution elements pair is logically consistent with each other. This means that they could be selected together in a driving simulator variant.

Fig. 12 shows a part of a consistency matrix based on the result with the assumption that each component has two solution elements. Dealing with the solution elements in this section will be illustrated in an abstract form, e.g., the solution elements will be called (A1, A2, B1, etc.); where A and B are components and A1 is the first solution element for the component A, etc.

The consistency matrix is filled out based on the dependency matrix. If a pair of components is independent (0 value in the dependency matrix), e.g., A and B, their solution elements will inherit this relation (1 value in the consistency matrix). Otherwise, if a pair of components is dependent (1 value in the dependency matrix), e.g., A and C, their solution elements will inherit the dependency relationship and they are either consistent or not (respectively 2 or 0 value in the consistency matrix).

Consistency matrix 0 = Logically Inconsistent 1 = Logically Neutral 2 = Logically Consistent			Hardware Components									
			A. Input Device		B. Visualization Device		C. Motion Platform		D. Acoustic Device		E. Vehicle Model	
			A1	A2	B1	B2	C1	C2	D1	D2	E1	E2
Hardware	A. Input Device	A1										
		A2										
	B. Visualization Device	B1	1	1								
		B2	1	1								
	C. Motion Platform	C1	2	0	2	0						
		C2	0	2	0	2						
	D. Acoustic Device	D1	1	1	1	1	1	1				
		D2	1	1	1	1	1	1				

Figure 12. The consistency matrix – example of some solution elements.

Consistency check sequence: considering the consistency relationship, which is determined by two-level matrices, the consistency check will also be performed by two level checks.

Fig. 13 shows a flowchart of the consistency check. For example, the consistency between solution elements A1 and B2 has to be checked. The first check will be based on the dependency matrix between the two parent components A and B. The second level will be based on the consistency matrix between the solution elements A1 and B2.

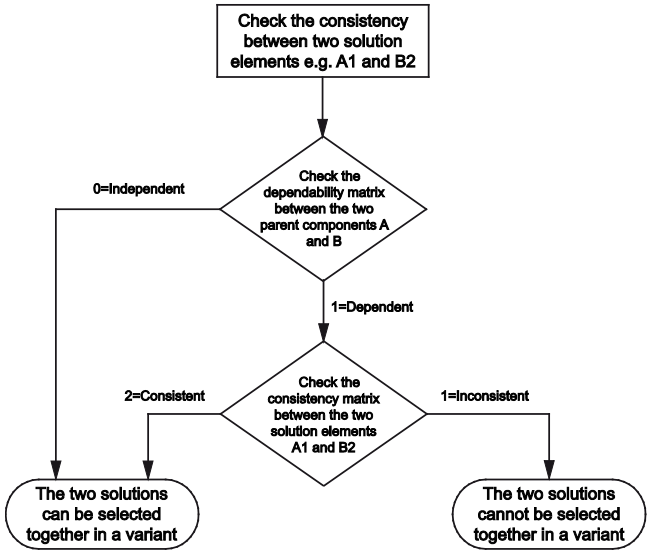


Figure 13. Consistency check flowchart.

2) Compatibility Check Algorithm

One of the main approaches to building a reconfigurable driving simulator is the ability of adding, removing or exchanging one or more solution elements. In order to build such a reconfigurable system, the applications/models interfaces have to be carried out automatically. Therefore, there is a need for an algorithm to check if all selected solution elements are compatible with each other or not. The compatibility here means whether the interfaces of the selected solution elements match together or not. Hence, each software component has its programming language and naming system of the input and output signals. Additionally, there is a need to extend the reconfigurable system

continuously by adding new unknown solution elements. Therefore, a generic solution elements' interface concept has been developed to manage and check different existing solution elements, as well as unknown solution elements that could be added in the future.

Generic solution elements' interface concept: in order to interface the entire solution elements, each solution element has to be considered as a black box. Mainly, only the input and output interfaces have to be considered. To keep the configuration process flexible and extendable, any solution element can be added as soon as its input and output interfaces are defined. The only required task for integrating any solution element is to map its inputs and outputs to the reconfigurable driving simulator's unique signal names there, this task is called signal multiplexing.

Fig. 14 shows an example of the signal multiplexing. A vehicle model has to be integrated as a solution element. The model will be considered as a black box, but all its input and output signals have to be mapped to the reconfigurable driving simulator's unique signal names. The output signal called "Otutput_ID563[m/s]" is the vehicle under test velocity in m/s, but this signal's unique name and unit predefined in the reconfigurable driving simulator has the name "Chassis_Velocity" and its unit is km/h. Also in this case, a simple unit conversion will be used.

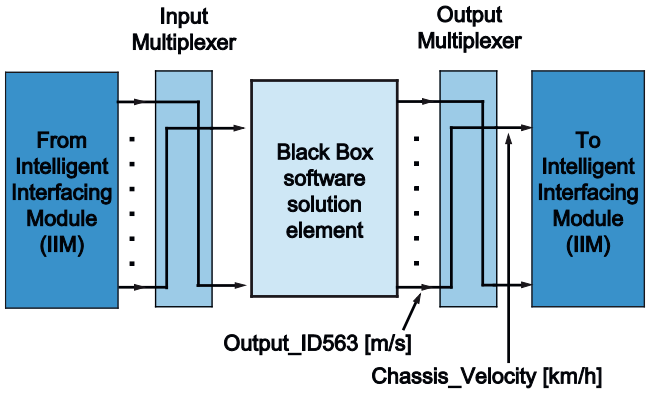


Figure 14. Generic solution elements interface concept.

In order to integrate this vehicle model, the user has to connect all the input and output signals with different names and units to the unique names and the units of the parent reconfigurable system. The input and output signals multiplexers should be programmed before registering the solution elements in the solution element database.

Compatibility check steps: after selecting the preferred solution elements, the compatibility check algorithm proofs the solution elements one by one to ensure that the input signals could be satisfied from the outputs from other solution elements. The compatibility check algorithm does not only check the signals' name but also other signal attributes such as frequency and unit to ensure the compatibility.

Fig. 15 shows a flowchart of the compatibility check. The compatibility check algorithm checks the compatibility of each signal through the following steps:

a) The algorithm checks each input signal of each selected solution element.

b) Each input signal has a unique name and must be delivered as an output from another selected solution element output. Therefore, the algorithm searches by the signal unique name in all output signals of the other selected solution element.

c) If the search engine finds the input signal as an output signal of the other selected solution elements that means this input signal could be satisfied.

d) Additionally, the search algorithm can check the compatibility of the signal unit and frequency. The output signal must have a greater frequency than the input signal or a sample rate converter will be required.

e) Then, the algorithm confirms the compatibility of this signal or stores an error in the error log.

These five steps have to be repeated for each input signal of each selected solution element.

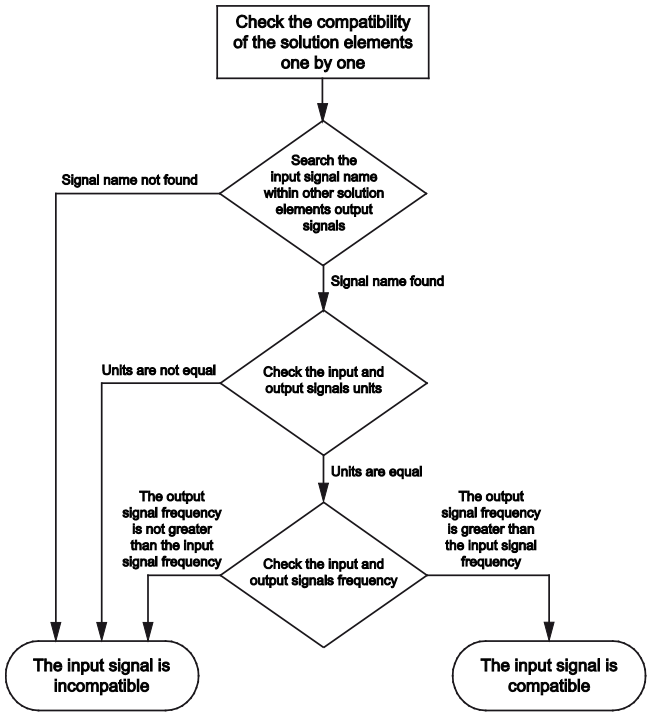


Figure 15. Compatibility check flowchart.

D. *Phase 4 – Solution Elements Deployment*

The first stage of the development procedure "System Development" was described, as well as its entire three phases. The first stage has to be carried out only once by the driving simulator developer. The result of the first stage is a reconfigurable driving simulator outline, which should be extended in the **variants creation stage** by the driving simulator operator. The first stage describes the system's entire components from a solution-neutral point of view. The second stage is the concretisation stage, which deals with solution elements instead of the solution-neutral components.

The second stage "variants creation" consists of three phases, starting with phase 4 "solution elements deployment". The main objective of this phase is to build a solution elements database, which contains the existing solution elements, their interfaces and attributes. This phase is an iterative process that has to be carried out each time to add or modify a solution element to the solution elements database.

The solution elements deployment is carried out in two steps. The first step is the identification and classification of the solution elements and the second step is the filling out of the solution elements database with the required attributes of each solution element.

1) Identify and Classify Solution Elements

The solution elements' identification and classification will be carried out based on the results of the first and second phases. The preferred solution elements will be carried out based on the morphological box concept according to Zwicky [21].

2) Filling the Solution Elements Database

In order to make the configuration tool deal with the component and solution elements, there is a need to register the identified components and solution elements in a database. This database stores and organizes the components and solution elements. It also has to be readable by the driving simulator operator and accessible by the configuration tool.

The main database operations are based on CRDU classes [22]: create, read, update, and delete. These operations must be covered by the database.

Create: This operation could be performed for both components and solution elements. The database is always extendable by adding a new component or by adding a new solution element for an existing component. This operation will be described in detail in this section.

Read: This operation can be executed for both components and solution elements. The database internal entries are accessible for the driving simulator operator, as well as for any software that would be used during the configuration process. All stored component and solution elements as well as their attributes can be accessed.

Update: This operation can be executed for both components and solution elements. Each stored component or solution element can be changed and restored.

Delete: This operation can be executed for both components and solution elements. Each stored component or solution element can be deleted from the database.

In this section, the create operation is described in detail in order to fill the solution elements database. The filling process is done in two steps: create component then create solution element.

Create a component entry: In order to create a component, the following attributes must be registered and stored in the database: **Component name** "which is the unique name of each component", **Component type "**a key component or an optional component", **Component classification** "hardware, software or resources", **Component description**, C**omponent symbol**, **Component logic dependency row** "which is a row contains the logic dependency between the components and the previously added components", and **Component guideline entry** " that is an optional attribute, which defines a preferred parameter value and condition regarding the component". For example, a guideline defines that the visualization device must have a minimum horizontal viewing angle of 100 degrees. This attribute can be added to the component in the form of the condition greater than (>) and parameter value (100 degrees).

Create a solution element entry: In order to create a solution element, the following attributes must be registered and stored in the database:

Solution Element Name: This attribute is the unique name for each solution element.

Solution Element Path: This attribute is the storage path on the file storage system. This is applicable only for an application/model.

Solution Element – Parent Component: This attribute is the name of the corresponding parent components. Therefore, it represents the relationship between this solution element and a component.

Solution Element Description: This attribute is a brief description of the solution element.

Solution Element Symbol: This attribute contains a symbol (logo) associated with the solution element.

Solution Element Author: This attribute is the solution element developer name, if known.

Solution Element Company: This attribute is the solution element producer company name if known.

Solution Element Release Date: This attribute is the date of when the solution element was released.

Solution Element Interface: This attribute is a table containing all the input and output signals of the solution element. Each signal has the following attributes:

Signal Name: It contains the names of the input and output signals of the corresponding solution element.

Input/Output: It indicates the direction of the signal, i.e., whether it is an input or an output signal.

From: It contains the component name from which this signal is to be fulfilled. This is applicable only for input signals.

Unit: It contains the measuring unit of the corresponding signal.

Frequency: It contains the sampling frequency of the corresponding signal.

Resolution: It contains the resolution of the corresponding signal.

Protocol: It contains the transmission protocol of the corresponding signal, e.g., Controller Area Network "CAN" or Transmission Control Protocol / Internet Protocol (TCP/IP) TCP/IP.

Physical Port: It contains the physical port used to transmit the corresponding signal.

Mandatory/Optional: It indicates whether the signal is mandatory or optional.

Description: It contains a brief description of the corresponding signal.

Solution Element Consistency Row: This attribute is a row, which contains the logic consistency between the

solution element and the previous added solution elements. This row is part of the solution elements consistency matrix.

Solution Element Guideline Entry: If the parent component has a guideline entry, the solution element inherits this entry and should define a parameter value for the entry to check the solution element confirmation with the guideline.

After registering all identified components and all preferred solution elements, which result from the metrological box in the database, the solution elements database is filled and ready to be used in the variant generation phase.

E. Phase 5 – Driving Simulator Variant Generation

The main objective of this phase is to define the configuration selection sequence, as well as define the configuration file structure, error reports structure and the physical connection plan.

1) Configuration Selection Sequence

In order to make a reasonable selection sequence for the solution elements, the identified components and their relationships have to be investigated. The selection sequence can be changed based on the area of use. During this phase, an example of the use case study shows how it can be determined.

The driving simulator components have been previously classified as three main classes: Hardware, software, and resources. A driving simulator structure is respectively based on hardware components, software, and finally, the used resources.

In order to make the selection sequence reasonable, it is not sufficient to make the selection sequence based on the classification, because of the tight correlation between some hardware and software components. Therefore, the identified components will be divided into groups of software and/or hardware based on the groups identified during the active structure specification step.

2) Configuration Files and Error Reports Structure

After the compilation of the solution elements' selection process, the configuration mechanism checks the selected components in terms of consistency and compatibility.

Based on the configuration mechanism check results, if the selected solution elements are consistent and compatible with each other, the configuration tool confirms that the selected solution elements can build a driving simulator variant and generates a configuration file. However, if the configuration tool finds any inconsistency or incompatibility between the selected solution elements, the configuration tool generates an error report. In the next section, the structures of the configuration file as well as the error report will be described.

Configuration File Structure: the configuration file is considered to be the result of the configuration process. It is a readable text file containing all the relative data about the selected variant. It consists of four parts: configuration data, hardware, software, and resources. The configuration data is the part that describes general information about the configuration itself, e.g., configuration name, author, etc. The hardware part contains all selected hardware solution elements attributes, parent component name and detailed input/output signal descriptions. The software part contains all selected software solution elements attributes, parent component name and detailed input/output signal descriptions. The resources part contains the selected resources.

Error Report Structure: the error report is a readable text file containing warnings and errors, which are detected by the configuration mechanism. It contains five parts: configuration data, hardware, software, resources and, errors/warning. The first four parts are the same as in the configuration file. The error and warning part lists all detected inconsistent solution elements, as well as all incompatible signals.

3) Physical Connections Plan

The configuration tool generates configuration files that contain the interfaces between the selected solution elements and the software side, but the configuration file does not contain the physical connections between the selected hardware solution elements and the selected resources. A physical connection plan is very useful for the driving simulator operator in order to prepare the driving simulator for operation. It shows in a simple way how the diverse hardware solution elements should be connected with the resource interfaces. It could be considered as a simple wiring plan.

Fig. 16 shows an example of the physical connection plan regarding. This variant consists of four hardware solution elements, which have to be connected to the simulation computer interfaces. With the help of the information stored in the solution elements database, the physical plan for the components can be generated. In this case, there were 4 connections, each hardware solution element is connected through one connection.

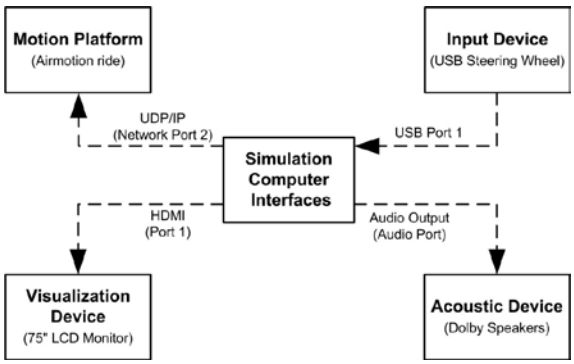


Figure 16. Example of a physical connection plan.

F. Phase 6 – System Preparation for Operation

The result of the fifth phase is the configuration file and a physical connection plan. The configuration file contains the selected solution elements, interface topology, and selected resources. Additionally, the physical connection plan contains the physical interfaces between the selected hardware solution elements.

There are two preparation steps required in order to build up the selected driving simulator variant and to prepare it for

the simulation. The first step is the preparation of the hardware connections and the second step is the software preparation.

1) Hardware Setup Preparation

Assuming that the selection process finished successfully and the configuration tool generated the physical connection plan, and then the driving simulator operator has to plug the different hardware solution elements together. The physical connection plan makes this step easy and understandable.

For the example, in Fig. 16, the driving simulator operator has to plug in 4 cables: a USB cable between the steering wheel and the simulation computer, an High-Definition Multimedia Interface "HDMI" cable between the 75" Liquid Crystal Display "LCD" monitor and the simulation computer, a network cable between the motion platform and the simulation computer, and an audio cable between the dolby speakers and the simulation computer. The example shows that the hardware preparation step can be easily done manually.

2) Simulation Software Preparation

To prepare the selected software solution elements for the operation, which is a complicated process (unlike the hardware preparation step) there is a need to develop software to assist this step. The software is called "Assistant". The assistant software is responsible for preparing the software solution elements for the simulation by the following three steps:

Read the configuration file: The assistant software can load and phrase the configuration file. It identifies the selected applications/models and their different attributes.

Fetch the applications/models: The assistant software retrieves the storage path for each application/model. It accesses the storage file system where the applications/models are stored.

Distribute the applications/models over resources: The assistant software loads each application/model on its corresponding source selected during the selection process.

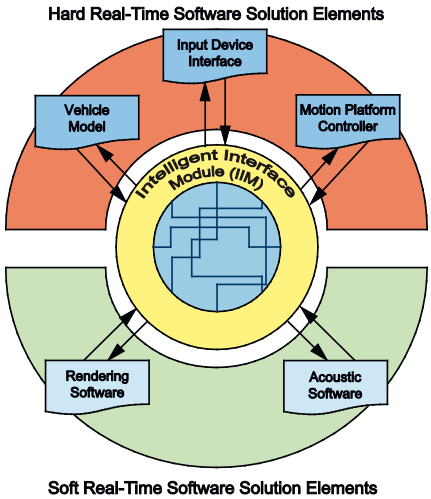


Figure 17. IIM function during simulation run-time.

The Intelligent Interfacing Module (IIM) initializes the communication between the selected software solution elements based on the interface topology, which is described in the configuration file. As soon as the user starts the simulation, the IIM ensures the communication between the simulation-related software solution elements during simulation run-time.

Fig. 17 shows the IIM function. The IIM exchanges the required input and output from and to the simulation related software solution elements during run-time. Moreover, IIM can connect the software solution elements together although a part of them runs under hard real-time conditions and the other part runs under soft real-time conditions.

The result of this phase is a ready-to-use driving simulator that consists of the selected software and hardware solution elements, as well as the selected resources.

VI. Implementation Prototype of the Configuration Tool

A prototype of the described concept has to be implemented as a part of this work. The implemented configuration tool consists of more than 150 embedded functions. This section describes the essential components of the configuration tool, the graphical user interface and the important tasks/functions covered by the tool.

The software was implemented using two software tools: Microsoft Office Excel and Matlab. The reconfigurable driving simulator database is implemented simply in MySQL. Further, the functions and algorithms are implemented with the help of Matlab M-Functions and the graphical user interface is implemented with the help of Matlab-GUI utility.

The development of the reconfigurable driving simulator database was done based on the relational database model approach. This approach is efficient and overcomes the complexity of the relationships between the entire different database tables. The implemented database mainly contains three types of tables: the components' table, the solution elements' table and the interfaces' table. These three types of tables are connected together based on a relational model of the database.

The dealing with the developed configuration tool is carried out mainly via a graphical user interface. Fig. 18 shows the start screen, which contains the main operations of the configuration tool and their correlation to the various phases of the development procedure model.

The start screen operations of the configuration tool are described as follows:

Configure New System: this operation is the essential task of the configuration tool. It is responsible for creating a new driving simulator variant by selecting solution elements for hardware, software, and resources in a predefined sequence; so that the user is prevented from dealing with complex algorithms such as consistency and compatibility check algorithms. Firstly, the consistency check algorithm runs in the background parallel to the selection steps. The configuration tool shows only the consistent solution elements that match with the previously selected solution element. Secondly, after the selection steps end, the

configuration tool executes the compatibility check algorithm to check the compatibility of the selected solution elements. After the compatibility check has finished, the configuration tool generates a configuration file if the selected solution elements are compatible with each other or it generates an error file if the selected solution elements are not compatible with each other.

Load Configuration File: this function allows the user to view and modify a previously generated configuration file. Moreover, it allows the operator to modify the previously generated configuration file by exchanging one or more of the previously selected solution elements.

View Components and Solution Elements: this function allows the user to deal with the stored components and the solution elements in the database. The user can view, modify or delete one or more component or solution element.

Add New Component: this function allows the user to add one new driving simulator component per execution. This function will guide the user through predefined schemes in order to register the different attributes of the new component.

Add New Solution Element: this function allows the user to add one new driving simulator solution element under a selected component per execution. This function will guide the user through predefined schemas in order to register the different attributes of the new solution elements.

Behind each operation in the main screen, a set of panels/schemas exists to accompany the user until he accomplishes the selected function.

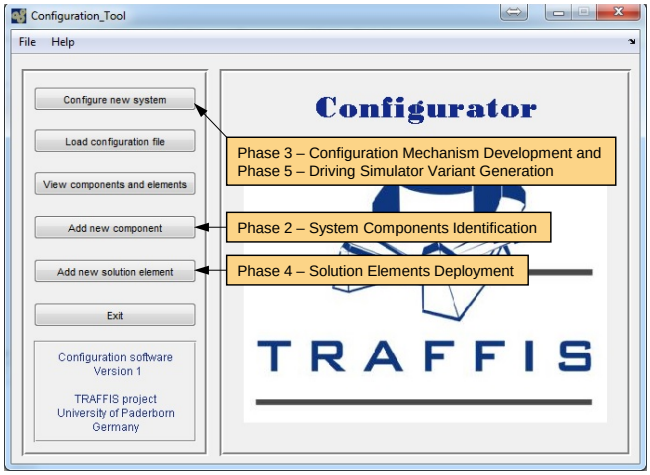


Figure 18. The graphical user interface of the configuration tool's implementation prototype – start screen.

VII. The Design Framework Validation

In order to validate the design framework, three ADAS driving simulator variants have been generated with the help of the described procedure model and the implementation prototype of the configuration tool. The three generated ADAS driving simulator variants were generated simply by selecting their desired components and their preferred solution elements.

A. *Configuration 1 – TRAFFIS-Full*

The name of the first generated variant is "TRAFFIS-Full". This variant has the most complex structure and it contains most of the ADAS reconfigurable driving simulator components. This variant is based on an application scenario. The main objective of the TRAFFIS-Full variant is testing the real Head-Lamp Control Module "HCM" control unit in HiL environment [23]. Additionally, the driving simulator motion platform and the real vehicle cabin allow the investigating of the inter-action between the driver and the HCM control unit in a Human-in-the-Loop environment. Fig. 19 shows the TRAFFIS-Full variant.

Figure 19. The TRAFFIS-Full variant.

The motion platform, which is used in this variant is the ATMOS motion platform. It consists of two dynamical parts with 5 DoF. The first dynamical part is the moving platform. It has 2 DOF and is used to simulate the lateral and longitudinal accelerations of the vehicle. It can move in the lateral plane and at the same time, it has the ability to tilt around its lateral axis with a maximum angle of 13.5 degrees and around the longitudinal axis with a maximum angle of 10 degrees. Four linear actuators are used to control the movements in both directions. The second dynamical part is the shaker system, which has 3 DOF to simulate the roll and pitch angular velocities and the vertical acceleration of the vehicle. It is driven by a three drive crank mechanism (three actuators).

B. *Configuration 2 – TRAFFIS-Portable*

The name of the second generated variant is "TRAFFIS-Portable". This driving simulator variant is a stripped-down version of the TRAFFIS-Full variant, which is based on an application scenario. The main objectives of the TRAFFIS-Portable variant are traffic safety training, as well as illustrating the bene-fits of ADAS functions. The traffic safety trainings typically take place on site at logistic agencies. Therefore, a portable driving simulator variant with a simple motion platform was needed. Fig. 20 shows the TRAFFIS- Portable variant.

Figure 20. The TRAFFIS-Portable variant.

C. Configuration 3 – TRAFFIS-Light

The name of the third generated variant is "TRAFFIS-Light". This variant has the simplest structure and contains the smallest number of ADAS reconfigurable driving simulator components. This variant is based on an application scenario. The main objective of the TRAFFIS-Light variant is testing the main HCM algorithms in the laboratory in a SiL simulation environment. The generated setup is a PC-based simulator with a simple vehicle model and a visualization system. Fig. 21 shows the TRAFFIS-Light variant.

Figure 21. The TRAFFIS-Light variant.

VIII. Conclusion and Outlook

Driving simulators have been used successfully for decades in different application fields. They vary in their structure, fidelity, complexity and cost from low-level driving simulators to high-level driving simulators. Nowadays, driving simulators are usually developed individually by suppliers and they are developed with a fixed structure to fulfil a specific task. Nevertheless, using a driving simulator in an application field, such as ADAS development, requires several variants of a driving simulator. These variants differ in their structure, in the used solution elements and in the level of detail of the entire models. Therefore, there is a need to develop a reconfigurable driving simulator, which allows its operator to easily create different variants without in-depth expertise in the system structure and without the help of the driving simulator's manufacturer.

Driving simulators are complex, interdisciplinary mechatronic systems. Therefore, the development of a reconfigurable driving simulator is a challenge. During the problem analysis, this challenge was analysed, the reconfigurable driving simulator term was de-fined and the essential requirements of the design framework were identified.

The extensive analysis of the state of the art has shown an existing method for the selection of the driving simulator and previous approaches towards developing reconfigurable driving simulators. The method named "Application Oriented Conception of Driving Simulators for the Automotive Development", developed by Negele, allows automotive engineers to formulate the requirements and specifications of a driving simulator for a specific application. Further to this, many driving simulators were investigated, but only seven of them could be identified as possible previous approaches towards developing a reconfigurable driving simulator. The seven identified driving simulators were classified into four categories: low-level, mid-level driving simulators, high-level, and multi-level driving simulators. The investigation of the existing methods and driving simulators has shown that there is no existing method or a developed driving simulator to date which covers all the design framework requirements. Therefore, a need for action was identified.

In order to solve the challenge of developing a reconfigurable driving simulator, a design framework for developing a reconfigurable driving simulator was developed to meet the defined requirements and to fulfil the need for action. The design framework consists mainly of the procedure model and the configuration tool.

The design framework has been validated with the help of a validation example. The validation example was the development of ADAS reconfigurable driving simulators. They are task-specific driving simulators, which are used for the testing and training of ADAS. During the validation, three variants of the reconfigurable driving simulator were successfully developed.

This paper described a modified procedure model comparing with [1]. Moreover, it showed a more detailed analysis of the state of the art, and it presented three validation examples of different driving simulators variants.

In summary, the developed design framework for developing a task-specific reconfigurable driving simulator is a comprehensive framework, which supports the driving simulator developers in their development of reconfigurable driving simulators. Moreover, it allows the driving simulator operators to easily create task-specific driving simulator variants.

Added value: In order to show the added value of using the design framework, two driving simulators variants: TRAFFIS-Portable and TRAFFIS-Light were developed individually. Each one of them has its fixed structure, certain software and hardware components. Furthermore, the interfaces between the different components were done manually. The development duration of the TRAFFIS-Portable variant was about four work months and of the TRAFFIS-Light was about three work months. By using the design framework the development duration of each was only two work weeks. That shows the benefits of using the design framework from the effort and cost points of view.

Outlook: The developed design framework for developing a reconfigurable driving simulator has considered the driving simulator as a mechatronic system. The procedure model and the configuration tool have been kept general, in order to be applicable for other mechatronic systems. The usage of the developed design framework for other mechatronic systems still has to be investigated. For example, in the plant engineering and construction field, most of the components are standard, e.g., conveyers, actuators, sensors, etc., as well as a customised components, e.g., controllers, robots, etc. This design framework can be easily adapted in order to configure customer-oriented plant solutions. These plant solutions are variants consisting of standard and customised components in a desired engineering design.

ACKNOWLEDGMENT

This work, as part of the project TRAFFIS (German acronym for "Test and Training Environment for Advanced Driver Assistance Systems"), which is funded by European Union "ERDF: European Regional Development Fund" and the Ministry of Economy, Energy, Industry, Trade and Craft of North Rhine Westphalia – Germany, within the "Ziel2" program.

We thank our project partner dSPACE for providing detailed vehicle and traffic models, as well as specific HiL-simulation hardware. We thank our project partner Varroc Lighting Systems GmbH for providing a head light control module for adaptive bending lights.

REFERENCES

[1] B. Hassan and J. Gausemeier, "Concept for a task–specific reconfigurable driving simulator," in Proc. International Conference on Advances in System Simulation (SIMUL 2013), IARIA, pp. 40-46, 2013.

[2] T. Hummel, M. Kühn, J. Bende, and A. Lang "Advanced Driver Assistance Systems – An investigation of their potential safety benefits based on an analysis of insurance claims in Germany," German Insurance Association – Insurers Accident Research, Research Report FS 03, Berlin, 2011.

[3] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of Advanced Driver Assistance Systems with Vehicle Hardware-in-the-Loop Simulations," The vehicle system dynamics, July 2006, volume 44, issue 7, pp. 569–590, 2006.

[4] M. Meywerk, "CAE-Methoden in der Fahrzeugtechnik," Springer-Verlag, Berlin, 2007.

[5] J. Gausemeier, P. Ebbesmeyer, and F. Kallmeyer, "Produktinnovation – Strategische Planung und Entwicklung der Produkte von Morgen," Carl Hanser Verlag München, 2011.

[6] J. Negele, "Anwendungsgerechte Konzipierung von Fahrsimulatoren für die Fahrzeugentwicklung," Ph.D. thesis, Faculty of Mechanical Engineering, 2007, Technische Universität München, Germany.

[7] S. Espié, E. Follin, G. Gallée, and D. Ganieux, "Automatic Road Networks Generation Dedicated to Night-Time Driving Simulation," in Proc. Driving Simulation Conference North America, 8.-10. October 2003, Dearborn, Michigan – ISSN 1546-5071.

[8] G. Weinberg and B. Harsham, "Developing a Low-Cost Driving Simulator for the Evaluation of In-Vehicle Technologies," in Proc. the First International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2009), September 21-22 2009, Essen, Germany.

[9] H. Jamson, "Cross-Platform Validation Issues," In: D. Fisher, J. Caird, M. Rizzo, J. Lee (Eds.): Handbook of Driving Simulation for Engineering, Medicine, and Psychology. CRC Press Taylor & Francis Group, USA, 2011, pp. 12.1-12.13 – ISBN 978-1-4200-6100-0.

[10] F. Filippo, A. Stork, H. Schmedt, and F. Bruno,"A modular architecture for a driving simulator based on the FDMU approach," International Journal on Interactive Design and Manufacturing (IJIDeM), Springer-Verlag, March 09 2013, Paris, France, 2013, ISSN 1955-2513.

[11] D. Gue, H. Klee, and E. Radwan, "Comparison of Lateral Control in a Reconfigurable Driving Simulator," in Proc. Driving Simulation Conference North America, 2003, Dearborn, Michigan, USA.

[12] E. Zeeb, "Daimler's new full-scale, high-dynamic driving simulator – A technical overview," in Proc. Driving Simulation Conference Europe 2010, September 9-10 2010, Paris, France, pp. 157-165 – ISBN 978-2-85782-685-9.

[13] National Advanced Driving Simulator, "Overview 2010," The University of Iowa – National Advanced Driving Simulator, Iowa City, USA, 2010.

[14] I. Gräßler, "Kundenindividuelle Massenproduktion". Springer-Verlag Berlin, 2004 – ISBN 978-3-642-18681-3

[15] S. Kreft, J. Gausemeier, M. Grafe, and B. Hassan "Automated generation of roadways based on geographic information systems," ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Washington DC, USA, 28-31 Aug. 2011.

[16] J. Gausemeier, U. Frank, J. Donoth, and S. Kahl,"Specification technique for the description of self-optimizing mechatronic systems," Research In: Research in Engineering Design, November 2009, Volume 20, Issue 4, Springer, London, 2009, pp. 201-223 – ISSN 0934-9839.

[17] M. Vaßholz, and J. Gausemeier, "Cost-Benefit Analysis – Requirements for the Evaluation of Self-Optimizing Systems," in Proc. 1st Joint International Symposium on System Integrated Intelligence 2012 – New Challenges for Product and Production Engineering, June 27-29 2012, Hannover, Germany, 2012, pp. 14-16.

[18] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, "Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung – Methoden und Anwendung," Springer-Verlag, Berlin, 7. Auflage, 2007.

[19] H. Birkhofer, "Analyse und Synthese der Funktionen technischer Produkte," VDI-Verlag Fortschritts-Bericht VDI-Z, Reihe 1, Nr. 70, Düsseldorf, Germany, 1980.

[20] G. Langlotz, "Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte. Forschungsberichte aus dem Institut für Rechneranwendung in Planung und Konstruktion," RPK der Universität Karlsruhe, Shaker Verlag, 2000.

[21] F. Zwicky, "Morphologische Forschung – Wesen und Wandel materieller und geistiger struktureller Zusammenhänge," Schriftenreihe der Fritz-Zwicky-Stiftung, Band 4, Verlag Baeschlin, Glarus, 1989 – ISBN 978-3-8135-0314-2.

[22] M. Brown, "Developing with Couchbase Server," O'Reilly Media, Sebastopol, California, USA, February 2013 – ISBN 978-1-4493-3116-0.

[23] C. Schmidt, "How to Make an AFS System Predictive: ADASIS Interface Implementation," in Proc. 7th International Symposium on Automotive Lighting, September 25-26, 2007, Darmstadt, Germany.

Ranked Particle Swarm Optimization with Lévy's Flight

Optimization of appliance scheduling for smart residential energy grids

Ennio Grasso, Giuseppe Di Bella, and Claudio Borean
Swarm Joint Open Lab
TELECOM ITALIA
Turin, Italy
e-mail: ennio.grasso@telecomitalia.it, giuseppe.dibella@telecomitalia.it, claudio.borean@telecomitalia.it

***Abstract*— This paper analyzes the problem of scheduling home appliances in the context of smart home applications. The optimization problem is modeled and different approaches to tackle it are presented and discussed. A new metaheuristic algorithm named Ranked Particle Swarm with Lévy flights (RaPSOL) is then proposed and described. The algorithm runs on the limited computational power provided by the home gateway device and in almost real-time as of user perception. Simulation results of RaPSOL algorithm applied in different use case scenarios are presented and compared with other approaches. The simulations include validation of the method in variable conditions considering both consumption, micro-generation and imposed user constraints.**

***Keywords*— scheduling; swarm intelligence; metaheuristic smart grids; smart homes.**

I. INTRODUCTION

This paper considers the minimum electricity cost scheduling problem of smart home appliances. Functional characteristics, such as expected duration and power consumption of the smart appliances can be modeled through a power profile signal. The optimal scheduling of power profile signals minimizes cost, while satisfying technical operation constraints and consumer preferences. Time and power constraints, and optimization cost are modeled in this framework using a metaheuristic algorithm based on a variant of Particle Swarm Optimization (PSO), presented in [1]. The algorithm runs on the limited computational power provided by the home gateway device and in almost real-time as of user perception. The context refers to the smart home environment, described in the INTrEPID European project [2], where a home environment equipped with plug-sensors and smart appliances can be used for enhanced smart energy management services.

The proposed framework can optimize appliance scheduling to minimize energy cost while avoiding the overload threshold. Very good quality solutions can be obtained in short computation time, in the order of a few seconds, which enables the deployment of this algorithm in low-cost embedded platforms.

Owing to the pliable characteristics of metaheuristic algorithms, the proposed algorithm is easily extended to incorporate solar power production forecasting in the presence of residential photovoltaic (PV) systems by simply adapting the objective function and using the solar energy forecaster as further input to the scheduler ([3][4]).

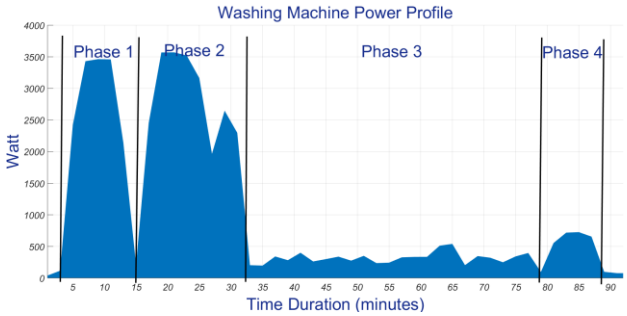


Figure 1. Example Power Profile with its phases generated by a washing machine

The paper is structured as it follows. Section II describes a model of the problem for the scheduling of smart appliance. Section III highlights how this problem can be classified as a NP-Hard Combinatorial Optimization Problem. In Section IV, a give a broad review of metaheuristic, while in Section V the new algorithm proposed in the paper is described. Section VI reports the results of the simulations of the proposed algorithm applied to the problem of scheduling smart appliances. Finally, Section VII contains concluding remarks and future analysis.

II. SCHEDULING PROBLEM OF SMART HOME APPLIANCES

Smart home applications are becoming one of the driving force of the Internet-of-Things (IoT), since connecting smart devices such as smart appliance to the internet envisions new scenarios that provide added value to both the final users and the other stakeholders. Possible applications are for instance the remote monitoring of smart appliances, remote activation/deactivation, automatic failure detection and alarm notification. Likewise, applications for appliance makers range from the remote diagnosis and assistance of appliances, thus reducing the assistance costs, to the collection of appliance statistics information useful to improve strategies for marketing new products, i.e., the appliance vendor could offer discounts in exchange for being allowed to get access to usage patterns and uncover the

features more appealing to their customers. To foster the pervasive adoption of these new IoT services, a common set of features need to be shared among the connected devices, so that "silo services" provided by each vendor are replaced by a smart home ecosystems where the connected appliance share value in participating.

One of the most successful application of smart home systems is energy management, since smart energy applications are enabled by IoT technologies and are shared by many home devices, which are mains powered. With the increased needs for energy sustainability, both regulatory and nationwide organizations are urging to the adoption of Renewable Energy Sources (RES) to reach the compelling target of the Horizon-2020 strategy. Since RES are by their nature variable and oftentimes difficult to predict exactly subject to variable weather conditions (PV performance mainly depends on cloud-cover condition, while wind turbines depends on the wind strength and direction), the final tariffs of electricity should match the fickle dynamics of the effective production cost instead of current two, or at most three tier model in most countries.

To enable a scenario with highly dynamic energy tariffs, it is essential the introduction if intelligent systems that can autonomously and conveniently schedule appliances to optimize energy use in presence of RES and variable tariffs.

On top of the above considerations, new actors such as Energy Aggregators are entering the market to collect and manage demands in so called "energy-districts". From the Aggregator standpoint, the proper management of energy demands of a set of users allows purchasing energy in the gross market and sharing the savings with the end users. These scenarios are explored in the INTrEPID project. Another important requirement is also the "shaving" of peak energy demands that cause inefficiencies in the electricity network (e.g., over-sizing electricity network to avoid blackouts) with additional costs and increased hazards (e.g., blackouts in case of power peaks not properly managed by the electricity network).

The management of users energy demand can be leveraged by the introduction of IoT systems, such as connected appliances, smart-plugs, smart-meters, apps for smartphones and tablet in order to visualize proposals to the users. These systems can take part in an energy management application with the aim to optimize the scheduling of appliances in the homes of district.

Taking into account the above considerations, not only is an automatic decision system highly desirable but even necessary in most cases, which either directly takes control of the appliances' operations (depending on the availability of smart appliances in the market), or at the very least is capable of providing advice to the home consumers (in case where using IoT system the appliance consumptions patterns can be learned and used for the scheduling).

This paper considers the minimum electricity cost scheduling problem of smart home appliances in the context of the INTrEPID Project. Functional characteristics, such as expected duration, mean and peak power consumption of smart appliances can be modeled through a power profile signal in time. Such power profiles could also be inferred by proper disaggregation of the cumulated power of a single smart meter with Non-Intrusive Load Monitoring (NILM) techniques. In other more advanced scenarios, the power profiles are notified by the smart appliances themselves. Protocols that enable that scenario have already been specified in several standard bodies and associations such as Energy@home [5].

In view of the above considerations, not only is an automatic decision system highly desirable but even necessary in most cases, which either directly takes control of the appliances' operations, or at the very least is capable of providing advice to the home consumers.

A. Smart Appliances in smart home

The smart home applications are enabled by communication between devices (e.g., smart appliances) in a home network typically enabled by wireless technologies.

The core element of a home network is the Home Gateway (HG) that coordinates and manages the smart appliances as end-devices. Among its functionalities, the HG provides the intelligence for real-time scheduling of residential appliances, typically in the time interval 24 hours, based on the tariff of the day, the forecasted energy power consumption, and possibly the forecasted wind/PV power generation.

The proposed scheduling framework borrows from the Power Profile Cluster defined in the E@H specifications [5], which specifies that each appliance operation cycle is modeled as a power profile composed by a set of sequential energy phases, as depicted in Figure 1. In some situation, and without loss of generality, a power profile has just a single phase, and in that simple case the power profile and its phase simply coincide.

In the more general case in which a power profile is composed of several energy phases, each phase represents an atomic subtask of the appliance's operation cycle. All phases are ordered sequentially since a phase cannot start until the previous phase is completed[1], however, there may be some degree of freedom in the time slack between one phase and the next.

Therefore, in general, each energy phase is characterized by a time duration and a power signal in time domain with the chosen sampling frequency[2], and a maximum activation delay after the end of the previous phase. Some phases have a maximum delay of zero, meaning that they cannot be delayed and must start soon after the previous phase completes. Other phases may be delayed adding extra flexibility in the scheduling of the power profile, e.g., the

[1] e.g., a washing machine agitator cannot start until the basin is filled with water

[2] Typical sampling frequency are 1 Hz or 1/60 Hz

washing machine agitator must start within ten minutes of the basin being filled.

Another input to the scheduler is the user's time constraints, demanding that certain appliances be scheduled within some particular time intervals, e.g., the dishwasher must run between 13:00 and 18:00.

The objective of the HG scheduler is to find the least expensive scheduling for a set of smart appliances, each characterized by a power profile with its energy phases, while satisfying the necessary operational constraints.

B. Modeling the Scheduling Problem

A first step in the scheduling problem modeling is to determine its dimension. Being *N* the number of appliances considered, and denoting by np_i the number of energy phases associated with each appliance i, the problem dimension, corresponding to the overall number of phases, is trivially given by

$$|P| \stackrel{\text{def}}{=} \sum_{i=1}^{N} np_i \quad (1)$$

The objective of the scheduler is to minimize the total electricity cost for operating the appliances based on the 24-hour electricity tariff while respecting time and energy constraints.

Denoting with $\boldsymbol{x} \in T^{|P|}$ the vector of start times of the $|P|$ phases, where T is the scheduling time interval, the problem can be stated as:

$$\boldsymbol{x} = \arg\min_{\boldsymbol{x}}(C(\boldsymbol{x})) \quad (2)$$

being $C(\boldsymbol{x})$, the total cost, expressed as

$$C(\boldsymbol{x}) = \sum_{i=1}^{N} \sum_{j=1}^{np_i} C(x_{ij}) \quad (3)$$

and $C(x_{ij})$ the cost of starting phase j of appliance i at time x_{ij}. The cost of a single phase at a given time is simply the product of the power phase signal and the tariff in the subinterval, L_{ij}, from the start time to the end of the energy phase.

$$C(x_{ij}) = \int_{x_{ij}}^{x_{ij}+L_{ij}} tariff(t)\, power_{ij}(t - x_{ij})dt \quad (4)$$

The integral notation assumes that the mean power is a Lebesgue integrable function. The above formulation is the most general possible, which assumes the power signal is a continuous function. An approximate formulation is to discretize the problem by choosing a reasonable sampling frequency, i.e., a trade-off with regard to the power profile signal variability and the desired system accuracy.

Following this idea, a reasonable approximation is to discretize the day time interval into 1440 time slots of 1 minute each. In such formulation, the above integral reduces to its summation approximate

$$C(x_{ij}) = \sum_{t=x_{ij}}^{x_{ij}+L_{ij}} tariff(t) \cdot power_{ij}(t - x_{ij}) \quad (5)$$

The max power constraint imposes that at any given time the amount of power required by all appliances' active phases be less than the peak power threshold specified by the grid operator. Let us define the auxiliary allocation function on the whole support of the scheduling interval *T*,

$$allocPower_{ij}(t) = \begin{cases} power_{ij}(t - x_{ij}) & if\ t \in [x_{ij}, x_{ij} + L_{ij}] \\ 0 & otherwise \end{cases}. \quad (6)$$

Now we can define the max power constraint as

$$\sum_{i=1}^{N} \sum_{j=1}^{np_i} allocPower_{ij}(t) \quad < maxPower\ ,\ \forall t \in T \quad (7)$$

While the max power constraints apply to the optimization problem, time constraints simply restrict the scheduling interval. Time constraints are twofold. On the one hand, the end user can impose a scheduling interval for any appliance, in terms of an earliest start time (EST), e.g., after 13:20, and a latest end time (LET), e.g., before 18:00.

$$EST_i \leq x_{i1}\ ;\ x_{iP} + L_{iP} \leq LET_i \quad (8)$$

The above time constraint means that start time of the 1st phase of appliance i, x_{i1}, must occur after the imposed EST_i. Likewise, the completion time of the last phase, denoted by $x_{iP} + L_{iP}$, must occur before the imposed LET_i.

The second time constraint is the maximum activation delay of each of the sequential phases that make up each power profile. While the scheduling interval specified in the first constraint is absolute, the maximum activation delays are relative and, therefore, the lower and upper bound time limits of each phase need to be adjusted based on the scheduling decisions for the previous phase.

$$(x_{ij} + L_{ij}) \leq x_{i(j+1)} \leq (x_{ij} + L_{ij}) + maxDelay_{i(j+1)} \quad (9)$$

III. NP-Hard Combinatorial Optimization Problems

Given the problem formulation, the scheduling of power profiles, each composed by a set of sequential and possibly delayable phases, under energy constraints is classified in the more general family of Resource Constrained Scheduling Problem (RCSP), which is known as being an NP-Hard combinatorial optimization problem [6][7] .

Moreover, the presence of time constraints introduces even another dimension to the complexity of problem, known as RCSP/max, i.e., RCSP with time windows. Combining the inherent complexity of the problem with the fact that the limited computing power of the HG which runs the logic of algorithm, and the almost real-time requirement for finding a solution (typically the end user wants a

perceived immediate answer), make the formulation a challenging problem.

From a theoretical perspective, combinatorial optimization problems have a well-structured definition consisting of an objective function that needs to be minimized (e.g., the energy cost) and a series of constraints. These problems are important for many real-life applications.

For some problems, exact methods can be exploited, such as branch-and-cut and Mixed Integer Linear Programming (MILP), with back-tracking and constraints propagation to prune the search space. However, in most circumstances, the solution space is highly irregular and finding the optimum is in general impossible. An exhaustive method that checks every single point in the solution space would be infeasible in these difficult cases, since it takes exponential time.

As a point of fact, [8] also addresses a similar scheduling problem of smart appliances, and relies on traditional MILP as a problem solver. They provide computation time statistics for their experiments, running on an Intel Core i5 2.53GHz equipped with 4GB of memory and using the commercial application CPLEX and MATLAB. According to their figures, discretizing the time interval in 10-minute discrete slots (for a total of 144 daily slots), takes their algorithm about 15.4 seconds to find a solution. With 5-minute slots the time rises to 83.6 seconds and with 3-minute slots to 860 seconds. From these figures, it is clear that a traditional approach like MILP is hardly acceptable for scheduling home appliances, and other more efficient methods need to be investigated.

A. Convex and Smooth Objective Functions

Generally speaking, optimization problems can be categorized, from a high-level perspective, as having either a convex or non-convex formulation.

A convex formulation enables to represent the objective function as a series of convex regions where traditional deterministic methods work best and fast, such as conjugate gradient descent and quasi-Newton variants, like L-BFGS (Limited memory Broyden–Fletcher–Goldfarb–Shanno). The main idea, in convex optimization problems, is that every constraint restricts the space of solutions to a certain convex region. By taking the intersection of all these regions we obtain the set of feasible solutions, which is also a convex region. Due to the nice structure of the solution space, every single local optimum is a global one. Most conventional or classic algorithms are deterministic. For example, the simplex method in linear programming is deterministic, and use gradient information in the search space, namely the function values and their derivatives.

Non-convex constraints create a many disjoint regions, and multiple locally optimal points within each of them. As a result, if a traditional search method is applied, there is a high risk of ending in a local optimum that may still be far away from the global optimum. But the main drawback is that it can take exponential time in the size of problem dimension to determine if a feasible solution even exists.

Another definition is that of smooth function, i.e., a function that is differentiable and its derivative is continuous. If the objective function is non-smooth, the solution space typically contains multiple disjoint regions and many locally optimal points within each of them. The lack of a nice structure makes the application of traditional mathematical tools, such as gradient information, very complicated or even impossible in these cases.

However, many real problems are neither convex nor smooth, and so deterministic optimization methods can hardly be applied.

B. An Overview of General Metaheuristic Algorithms

A problem is NP-Hard if there is not an exact algorithm that can solve the problem in polynomial time with respect to the problem's dimension. In other words, aside from some "toy-problems", an NP-Hard problem would require exponential time to find a solution by systematically "exploring" the solution space.

A common method to turn an NP-Hard problem into a manageable, feasible approach is to apply heuristics to "guide" the exploration of the search space. These heuristics are based on "common-sense" specific for each problem and are the basis for developing Greedy Algorithms that can build the solution by selecting at each step the most promising path in the solution space based on the suggested heuristics. Obviously, this approach is short-sighted since it proceeds with incomplete information at each step. Very rarely do greedy algorithms find the best solution or worse yet they might fail to find a feasible solution even if one does exist.

A better approach for solving complex NP-Hard problems that has shown great success is based on metaheuristic algorithms. The word *meta* means that their heuristics are not problem specific to a particular problem, but general enough to be applied to a broad range of problems. Examples of metaheuristic algorithms are *Genetic and Evolutionary Algorithms*, *Tabu search*, *Simulated Annealing*, *Greedy Randomized Adaptive Search Procedure*, *Particle-Swarm-Optimization*, and many others.

The idea of metaheuristics is to have efficient and practical algorithms that work most the time and are able to produce good quality solutions, some of them will be nearly optimal. Figuratively speaking, searching for the optimal solution is like *treasure-hunting*. Imagine we are trying to find a hidden treasure in a hilly landscape within a time limit. It would be a silly idea to search every single square meter of an extremely large region with limited resources and limited time. A more sensible approach is to go to some place almost randomly and then move to another plausible place using some hints we gather throughout.

Two are the main elements of all metaheuristic algorithms: intensification and diversification. *Diversification* via randomization means to generate diverse solutions so as to explore the search space on the global scale and to avoid being trapped at local optima.

Intensification means to focus the search in a local region by exploiting the information that a current good solution is found in this region as a basis to guide the next step in the search space. The fine balance between these two elements is very important to the overall efficiency and performance of an algorithm.

IV. CLASSIFICATION OF METAHEURISTIC ALGORITHMS

Metaheuristic algorithms are broadly classified in two large families: *population-based* and *trajectory-based*. Going back to the treasure-hunting metaphor, in a trajectory-based approach we are essentially performing the search alone, moving from one place to the next based on the hints we have gathered so far. On the other hand, in a population-based approach we are asking a group of people to participate in the hunting sharing all information gathered by all members to select the most promising paths for the next moves.

A. Genetic Algorithms

Genetic Algorithms (GA) were introduced by John Holland and his collaborators at the University of Michigan in 1975 [9]. A GA is a search method based on the abstraction of Darwinian evolution and natural selection of biological systems, and representing them in the mathematical operators: *crossover* (or recombination), *mutation*, *fitness evaluation* and *selection* of the best. The algorithm starts with a set of candidate solutions, the initial population, and generate new offspring through random mutation and crossover, and then applies a selection step in which the worst solutions are deleted while the best are passed on to the next generation. The entire process is repeated multiple times and gradually better and better solutions are obtained. GA algorithms represent the inseminating idea of all more recent population-based metaheuristics.

One major drawback of GA algorithms is the "conceptual impedance" that arises when trying to formulate the problem at hand with the genetic concepts of the algorithm. The formulation of the fitness function, population size, the mutation and crossover operators, and the selection criteria of the offspring population are crucially important for the algorithm to converge and find the best, or quasi-best, solution.

B. Simulated Annealing

Simulated Annealing (SA) was introduced by Kirkpatrick et al. in 1983 [10] and is a trajectory-based approach that simulates the evolution of a solid in a heat bath to thermal equilibrium. It was observed that heat causes the atoms to deviate from their original configuration and transition to states of higher energy. Then, if a slow cooling process is applied, there is a relatively high chance for the atoms to form a structure with lower internal energy than the original one. Metaphorically speaking, SA is like dropping a bouncing ball over a hilly landscape, and as the ball bounces and loses its energy it eventually settles down to some local minima. But if the ball loses energy slowly enough keeping its momentum, it might have a chance to overcome some local peaks and fall through a better minimum.

C. Particle Swarm Optimization

Particle Swarm Optimization (PSO), introduced in 1995 by American social psychologist James Kennedy, and engineer Russell C. Eberhart [11], represents a major milestone in the development of population-based metaheuristic algorithms. PSO is an optimization algorithm inspired by swarm intelligence of fish and birds or even human behavior. The multiple particles swarm around the search space starting from some initial random guess and communicate their current best solutions and also share their best. The greatest advantage of PSO over GA is that it is much simpler to apply in the formulation of the problem. Instead of using crossover and mutation operations it exploits global communication among the swarm particles. Each particle in the swarm modifies its position with a velocity that includes a first component that attracts the particle towards the best position so far achieved by the particle itself. This component represents the personal experience of the particle. The second component attracts the particle towards the best solution so far achieved by the swarm as a whole. This component represents the social communication skill of the particles.

Denoting with N the dimensionality of the search space, i.e., the number of independent variables that make up the exploring search space, each individual particle is characterized by its position and velocity N-vectors. Denoting with x_i^k and v_i^k respectively the position and velocity of particle i at iteration k, the following equations are used to iteratively modify the particles' velocities and positions:

$$v_i^{k+1} = w v_i^k + c_1 r_1 \left(p_i - x_i^k\right) + c_2 r_2 \left(g^* - x_i^k\right) \quad (10)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (11)$$

where w is the *inertia* parameter that weights the previous particle's momentum; c_1 and c_2 are the *cognitive* and *social* parameter of the particles multiplied by two random numbers r_1 and r_2 uniformly distributed in $[0-1]$, and are used to weight the velocity respectively towards the particle's personal best, $\left(p_i - x_i^k\right)$, and towards the global best solution, $\left(g^* - x_i^k\right)$, found so far by the whole swarm. Then the new particle position is determined simply by adding to the particle's current position the new computed velocity, as shown in Figure 2.

The PSO coefficients that need to be determined are the inertia weight w, the cognitive and social parameters c_1 and c_2, and the number of particles in the swarm.

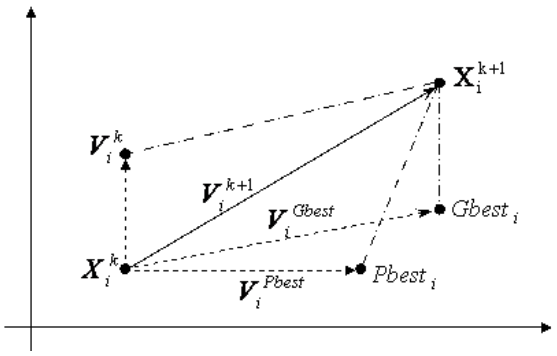


Figure 2. New particle position in PSO

We can interpret the motion of a particle as the integration of Newton's second law, where the components $c_1 r_1(p_i - x_i^k) + c_2 r_2(g^* - x_i^k)$ are the attractive forces produced by springs of random stiffness, while w introduces a virtual mass to stabilize the motion of the particles, avoiding the algorithm to diverge, and is typically a number such that $w \approx [0.5 - 0.9]$. It has been shown, without loss of generality, that for most general problems the number of parameters can even be reduced by taking $c_1 = c_2 \approx 2$.

D. Quantum Particle Swarm Optimization

Although much simpler to formulate than GA, classical PSO has still many control parameters and the convergence of the algorithm and its ability to find a near-best global solution is greatly affected by the value of these control parameters. To avoid this problem a variant of PSO, called Quantum PSO (QPSO) was formulated in 2004 by Sun et al. [12], in which the movement of particles is inspired by quantum mechanics.

The rationale behind QPSO stems from the observation that statistical analyses have demonstrated that in classical PSO each particle i converges to its local attractor a_i defined as

$$a_i = (c_1 p_i + c_2 g^*)/(c_1 + c_2) \quad (12)$$

where p_i and g^* are the personal best and global best of the particle. The local attractor of particle i is a stochastic attractor that lies in a hyper-rectangle with p_i and g^* being two ends of its diagonal, and the above formulation can also be rewritten as

$$a_i = r p_i + (1 - r) g^* \quad (13)$$

where r is a uniformly random number in the range $[0 - 1]$.

In classical PSO, particles have a mass and move in the search space by following Newtonian dynamics and updating their velocity and position at each step. In quantum mechanics, the position and velocity of a particle cannot be determined simultaneously according to uncertainty principle. In QPSO, the positions of the particles are determined by the Schrödinger equation where an attractive potential field will eventually pull all particles to the location defined by their local attractors. The probability of particle i appearing at a certain position at step $k + 1$ is given by:

$$x_l^{k+1} = a_i + \beta \left| x_{mbest}^k - x_l^k \right| \ln(1/u), if\ v \geq 0.5 \quad (14)$$

$$x_l^{k+1} = a_i - \beta \left| x_{mbest}^k - x_l^k \right| \ln(1/u), if\ v < 0.5 \quad (15)$$

where u and v are uniformly random numbers in the range $[0 - 1]$, x_{mbest}^k is the mean best of the population at step k defined as the mean of the best positions of all particles

$$x_{mbest}^k = \sum_{i=1}^{N} p_i \quad (16)$$

β is called *contraction-expansion* coefficient and controls the convergence speed of the algorithm.

The QPSO algorithm has been shown to perform better than classical PSO on several problems due to its ability to better explore the search space and also has the nice feature of requiring one single parameter to be tuned, namely the β coefficient. The exponential distribution of positions in the update formula makes QPSO search in a wide space.

Moreover, the use of the mean best position x_{mbest}, each particle cannot converge to the global best position without considering all other particles, making them explore more thoroughly around the global best until all particles are closer. However, this may be both a blessing and a curse; it may be more appropriate in some problems but it may slow the convergence of the algorithm in other problems. Again, there is a very fine balance between exploration and exploitation. How large is the search space, and how much time is given to explore before returning a solution.

E. Dealing with Constraints

Many real world optimization problems have constraints, for example, the available amount of certain resources, the boundary domain of certain variables, etc. So an important question is how to incorporate constraints in the problem formulation.

In some cases, it may be simple to incorporate the feasibility of solutions directly in the formulation of a problem. If we know the boundary domain of a certain dependent variable and the proposed solution violates such domain we can either reject the solution or modify it by constraining the variable within the boundaries. For example, suppose a time variable must satisfy the time interval between 9:00 and 13:00, while the proposed solution would place it at 14:34. One way to deal with the above violation is to constrain the variable to its upper bound (UB) 13:00 and reevaluate the objective function. This will be probably worse than before, but at least it will be feasible and need not be rejected altogether.

A common practice is to incorporate constrains directly in the formulation of the objective function through the

addition of a *penalty* element so that a constrained problem becomes unconstrained. If $f(x)$ is the objective function to be minimized, any equality / disequality constrains can be cast to penalty terms linearly added to the objective function, typically with a high weight w and a quadratic function in the measured violation $g(x) = \max(0, v(x)^2)$, where $v(\cdot)$ "measure" the amount of violation. Now the augmented optimization problem becomes

$$\arg\min_x (f(x) + w \cdot g(x)) \quad (17)$$

w is the penalty weight that needs to be large enough to skew the choice of the fittest solutions towards the smallest penalty component, typically in the range 10^9 - 10^{15}.

In our scheduling problem we have already defined the max power constraint as upper bound inequality.

$$g(t) \stackrel{\text{def}}{=} \max\left[0, \left(\sum_{i=1}^{N} \sum_{j=1}^{np_i} allocPower_{ij}(t) \right) - maxPower\right] \quad (18)$$

F. Nature Inspired Random Walks and Lévy Flights

A random walk is a series of consecutive random steps starting from an original point: $x_n = s_1 + \cdots + s_n = x_{n-1} + s_n$, which means that the next position x_n only depends on the current position x_{n-1} and the next step s_n. This is the typical main property of a Markov chain. Very generally, we can write the position in random walks at step $k+1$ as

$$x_{k+1} = x_k + s\sigma_k \quad (19)$$

where σ_k is a random number drawn from a certain probability distribution. In mathematical terms, each random variable follows a probability distribution. A typical example is the normal distribution and the random walk becomes a *Brownian* motion. Besides the normal distribution, the random walk may obey other non-Gaussian distributions.

For example, several studies have shown that the random walk behavior of many animals and insects have the typical characteristics of the *Lévy* probability distribution and the random walk is called a Lévy flight [13][14][15]. The Lévy distribution has the characteristic of being both stable and heavy-tailed. A stable distribution is such that any sum n of random number drawn from the distribution is finite and can be expressed as

$$\sum_{i=1}^{n} x_i = n^{1/\alpha} \cdot x \quad (20)$$

where α is called the index of stability and controls the shape of the Lévy distribution with $0 < \alpha \leq 2$. Notably, two value for α are special cases of two other distribution, the normal distribution for $\alpha = 2$, and the Cauchy distribution for $\alpha = 1$.

The heavy-tail characteristic implies that the Lévy distribution has an infinite variance, decaying for large x to $\lambda(x) \sim |x|^{-1-\alpha}$.

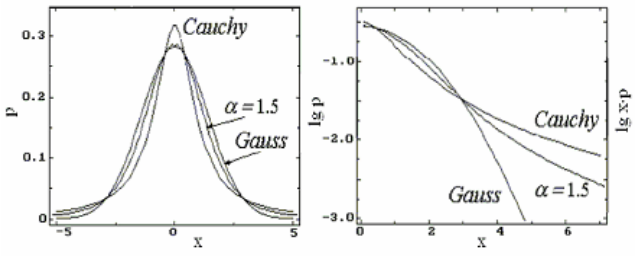


Figure 3. Cauchy

Figure 3 shows the shapes of the normal, Cauchy, and Lévy distribution with $\alpha = 1.5$. The difference becomes more pronounced in the logarithmic scale showing the asymptotic behavior of the Lévy and Cauchy distribution compared with the normal.

Due to the stable property, a random walker following the Lévy distribution will cover a finite distance from its original position after any number of steps. Also, due to the heavy-tail of the distribution, extremely long jumps may occur, and typical trajectories are self-similar, on all scales showing clusters of shorter steps interspersed by long excursions, as shown in Figure 4. In fact, the trajectory of a Lévy flight has fractal dimension $d_f = \alpha$.

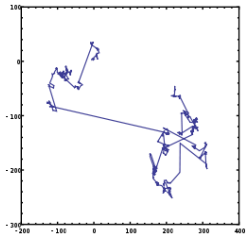


Figure 4. Levy's flight

In that sense, the normal distribution in Figure 5 represents the limiting case of the basin of attraction of the generalized central limit theorem for $\alpha = 2$ and the trajectory of the walker follows a Brownian motion.

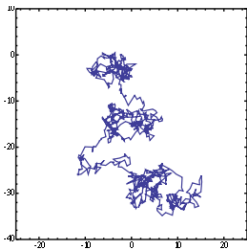


Figure 5. Brownian path

Due to the properties of being both stable and heavy-tailed, it is now believed that the Lévy distribution nicely describes many natural phenomena in physical, chemical, biological and economical systems. For instance, the foraging behaviors of bacteria and higher animals show typical Lévy flights, which optimize the search compared to Brownian motion giving a better chance to escape from local optima.

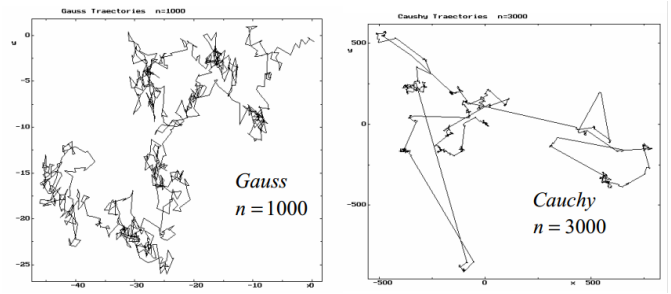


Figure 6. the trajectories of a Gaussian (left) and a Lévy (right) walker

Figure 6 shows the trajectories of a normal (left) and a Lévy (right) walker. Both trajectories are statistically self-similar, but the Lévy motion is characterized by island structure of clusters of small steps, connected by long steps.

G. Step Size in Random Walks.

In the general equation of a random walk $x_{k+1} = x_k + s\sigma_k$, a proper step size, which determines how far a random walker can travel after k number of iterations, is very important in the exploration of the search space. The two component that make up the step are the scaling factor *s* and the length of the random number in the distribution σ_k. A proper step size is very important to balance exploration and exploitation, too small a step and the walker will not have a chance to explore potential better places, on the other hand, too large steps will scatter the search from the focal best positions. From the theory of isotropic random walks, the distance traveled after k steps in N dimensional space is

$$D = s\sqrt{kN}\,. \tag{21}$$

In a length scale L of a dimension of interest, the local search is typically reasonably limited in the region $D = L/10$, which means that the scaling factor

$$s \approx \frac{L}{10\sqrt{kN}} \tag{22}$$

In typical metaheuristic optimization problems, we can expect the number of iterations k in the range 100 – 1000.

For example, with 100 iterations and $N = 1$ (a one dimensional problem) we have $s = 0.01L$, and to another extreme with 1000 iterations and $N = 10$ we have $s = 0.001L$. Therefore, a scaling factor between 0.01 – 0.001 is basically a reasonable choice in most optimization problems. L is still kept independent as each dimension of the problem may very well have a very different length scale.

V. RANKED PARTICLE SWARM WITH LÉVY FLIGHTS

In this section, we describe a variant of the QPSO, named Ranked PSO with Lévy flights (RaPSOL) that introduces some innovative strategies on the QPSO borrowed from other disciplines, like observations of natural phenomena, and the nice properties of ranking in descriptive statistics the nonparametric measures of dependence, namely Spearman's rho and Kendall's tau.

The result is an algorithm that provides a nice balance between exploration and exploitation and gives good-quality solutions in short time and with limited computing power.

In fact, the Home Gateway (HG) is a low power ARM embedded system running a Java Virtual Machine in the OSGi framework.

The first innovation is to replace the exponential probability density function with the Lévy distribution. A second innovation is to improve the global exploration search by shifting the attention from just the single best global leader, to all the ranked particles. In fact, one shortcoming of standard leader-oriented swarm algorithms is that they tend to converge very fast to the current best solution, sometimes missing other promising search area. With only one global leader, all particles quickly converge together, something missing better solutions.

To overcome that shortcoming, in RaPSOL, particles are ranked according to their fitness and instead of just considering the global best particle for determining the current attractor, any particle is entitled to choose any other better particle, not just the global best. This selection is uniform-random: the second best particle is only entitled to choose the best particle, and in general each particle may choose any other better particle as its current attractor.

The introduction of ranked selection is enough to guarantee a broader search in the problem domain avoiding premature convergence to local optima. The algorithm steps are thus:

1. Rank all particles according to their current fitness.
2. For each particle, randomly select any particle whose fitness is better than this particle's. Name such particle the relative leader.
3. Take a uniform random point in the linear hyperplane that intersects the particle's personal best position and the relative leader. Name this point the particle's attractor
4. Do a Lévy flight from the attractor with a step-size proportional to the swarm's current radius, by constraining on the current distance of the particle and the relative leader.

From our experiments and simulations, the effect of ranking, coupled with the Lévy distribution, has proven to

exhibit very good results compared to traditional PSO and QPSO.

For our purposes, the Lévy distribution coefficient α chosen in RaPSOL is actually the Cauchy coefficient $\alpha = 1$.

The Cauchy random generator is much simpler than the more general algorithm for Lévy generation and that is a determining factor in runtime execution. Since the random generation needs to be executed for an umpteen number of times (i.e., the dimension of the problem, by the number of particles in the swarm, by the number of iterations of the algorithm), the computing speed of the random generation is of paramount importance. From our experiments, within a given time limit allotted to the algorithm to find a solution, the Cauchy version of the algorithm is able to execute almost twice the number of iterations than the general Lévy version. Therefore, even if there was an optimal coefficient α that provides better results for the same number of iterations, it will be outperformed by the Cauchy variant that with more allowed iterations finds better solutions.

VI. SIMULATION AND RESULTS

We ran a number of simulations modeling the same scheduling problem both in the RaPSOL algorithm and a pure mathematical model with commercial linear programming (LP) solvers, namely XPress and CPLEX.

The scheduling problem was formalized with 4 instances of washing-machine power profiles, each profile being made of 4 phases, and 3 instances of dish-washing-machines each made of 5 phases, for a total of 31 independent variables to optimize in the scheduling problem instance.

Table I. Comparison of different tested algorithms over dataset described in the first two columns.

#Appl	Max Power	TiLim	XPRESS Cost	CPLEX Cost	SYMPHONY Cost	RaPSOL Cost
3	4000	10 s	2,5384	2,5381	3,2278	2,5381
		60 s	2,5381	2,5381	2,5381	2,5381
3	3000	10 s	2,5384	2,5381	3,2278	2,5381
		60 s	2,5381	2,5381	2,5381	2,5381
3	2000	10 s	2,5381	2,5381	2,5425	2,5381
		60 s	2,5381	2,5381	2,5381	2,5381
5	4000	10 s	4,8643	4,8643	4,8870	4,8643
		60 s	4,8643	4,8643	4,8685	4,8643
5	3000	10 s	4,8643	4,8671	4,8925	4,8643
		60 s	4,8643	4,8643	4,8925	4,8643
5	2000	10 s	4,9700	5,2445	–	4,9551
		60 s	4,9700	4,9545	–	4,9551
10	4000	10 s	12,8896	9,7318	–	9,1123
		60 s	9,1929	9,0141	–	9,0149
10	3000	10 s	12,3398	12,1620	–	9,6051
		60 s	10,0963	9,2056	–	9,3841
10	2000	10 s	11,4725	11,9863	–	11,022
		60 s	11,4561	11,6960	–	10,809
15	4000	10 s	–	19,2274	–	14,906
		60 s	15,0148	15,0497	–	14,681
15	3000	10 s	–	18,0702	–	16,347
		60 s	16,3381	16,8535	–	16,172

Due to the hard problem space for the brute-force exact algorithms, the scheduling horizon was limited to 12 hours and the time slots at multiples of 3 minutes, otherwise, with one-minute slot time, no feasible solutions were found even in 7 days of uninterrupted run.

Running 96 hours, XPress found a solution at a cost of € 2.57358. With the same problem and running 1 hour CPLEX found a solution at € 2.59123. Finally, the RaPSOL was given a bound time of 15 seconds, and run 10 times to have reliable statistics, finding a best solution at € 2.7877, with an average cost of € 2.9351 for the 10 times. We additionally report in Table I results of compared algorithms over an extended dataset (described in the first two columns), where missing entries mean that the target algorithm has not been able to achieve any result in the specified time limit.

The results obtained using linear programming and exact solvers are very important as they fix theoretical optima for benchmarking the convergence and performance of the metaheuristic approach of the RaPSOL. Results show that although RaPSOL finds a worse solution than the theoretical optimum by a 8 – 13 %, the very short allotted time to find a solution is anyway a very promising approach. In Figure 7 and Figure 8 are reported simulation results when considering appliance scheduling with constant overload threshold, variable tariff, and with the absence and presence of photovoltaic generation respectively. An interesting use case is the scheduling of an entire apartment building where tenants share a common contract with the utility provider in which the energy consumption of the apartment house as a whole must be below a given "virtual" threshold that changes in time. Figure 9 shows such scenario. The curved red line represents the virtual threshold that the apartment house should respect.

All energy above such threshold will not cause an overload but its cost grows exponentially with the net effect of encouraging a peak shaving of profile allocation. The case study of Figure 10 is a scheduling of 15 apartments, with 3 appliances each, for a total of 45 appliances. The apartment house is also provided with common PV-panels.

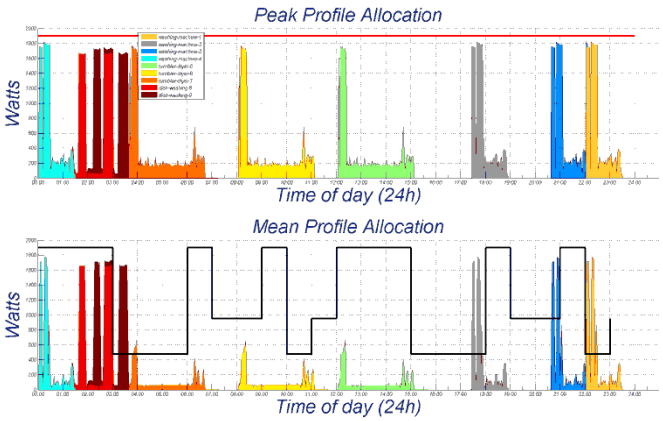


Figure 7. RAPSOL simulation results: appliance scheduling with constant overload threshold, variable tariff, no photovoltaic.

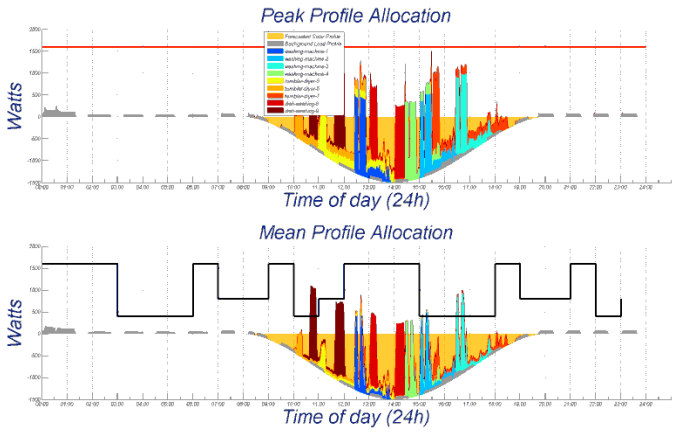


Figure 8. RAPSOL simulation results: appliance scheduling with constant overload threshold, variable tariff, photovolltaic.

The 3 case studies described here show the remarkable flexibility of the RaPSOL algorithm, and many other metaheuristic algorithms for that matter, i.e., the ability to adapt the algorithm to the unique attributes of a given problem and not based on predefined characteristics.

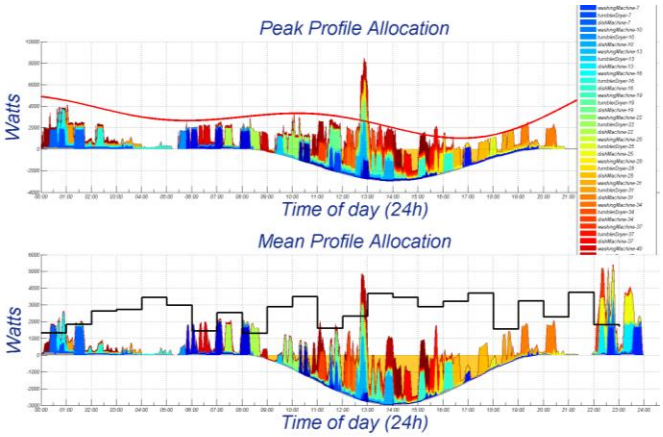


Figure 9. RaPSOL simulation results: appliance scheduling for different apartments with variable overload threshold, variable tariff, photovoltaic.

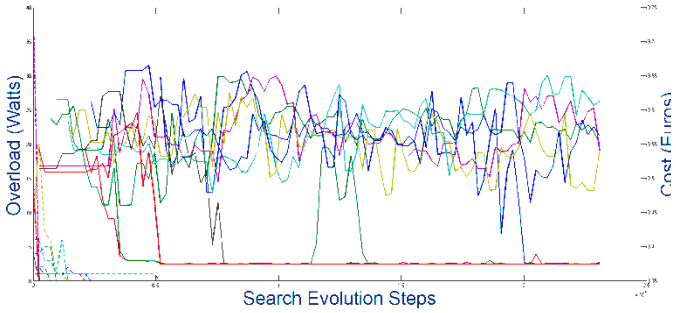


Figure 10. RaPSOL simulation results: overload avoidance and optimization of cost

A. *Extended simulations setups*

Extended simulations with a number of appliances equal to 50, overload threshold of 4kW and different tariffs schemes are shown in the following figures. The different conditions comprise:

- tariffs (in the lower part of each figure) highly dynamic or three tiers:
- solar generation: photovoltaic generation with clear sky conditions (present or not present).

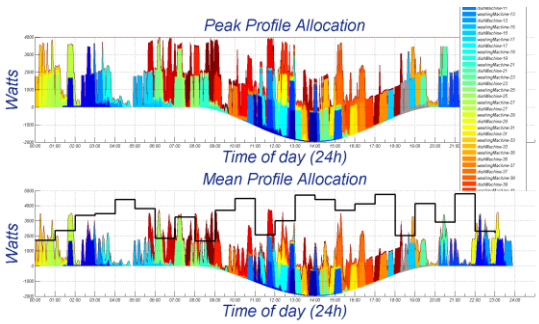


Figure 11. RaPSOL simulation results for the case: 50 appliances, overload threshold of 4kW, dynamic tariff, photovoltaic with clear sky condition

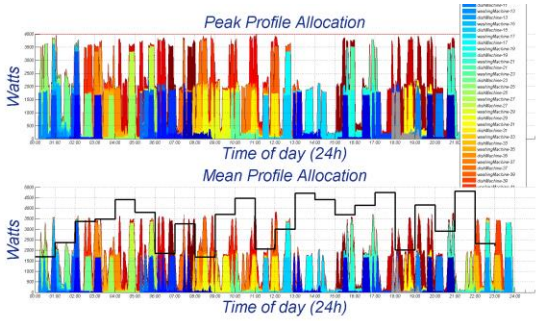


Figure 12. RaPSOL simulation results for the case: 50 appliances, dynamic tariff, overload threshold of 4kW, no photovoltaic

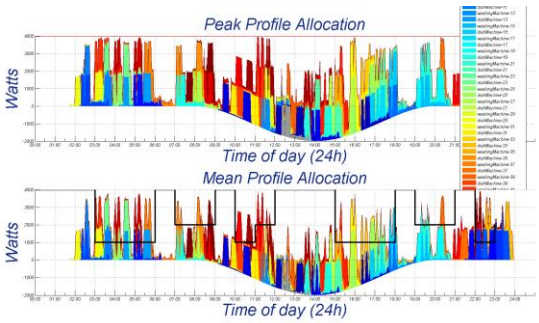


Figure 13. RaPSOL simulation results for the case: 50 appliances, overload threshold of 4kW, three-tiers tariff, photovoltaic with clear sky condition

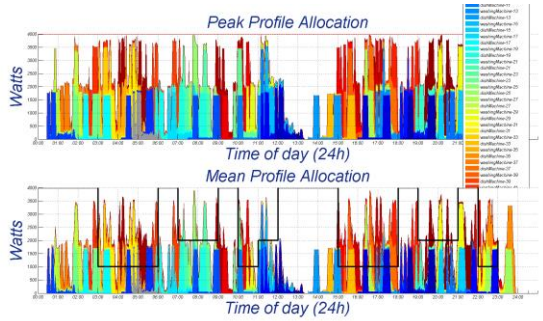


Figure 14. RaPSOL simulation results for the case: 50 appliances, three-tiers tariff, overload threshold of 4kW, no photovoltaic

B. Comparison with growing number of appliances

In order to verify the performances of RaPSOL considering a growing number of appliances, different simulations have been performed and compared in Figure 15. The cost normalized for a single appliance is shown. Clearly, the case with no solar generation has higher cost. As expected, increasing the number of appliances also downgrade the final solution because of the increased dimension and complexity of the optimization.

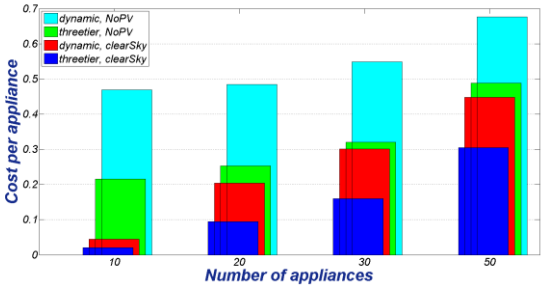


Figure 15. RaPSOL simulation results for a growing number of appliances for the different tariffs, overload threshold of 4kW, photovoltaic or no photovoltaic

C. Discussion and considerations

In a rapidly changing world, algorithmic paradigms that are flexible and easy to adjust offer a competitive advantage over rigid, tailor based methods. In such volatile domains, the usefulness of an algorithm framework will not be given by its ability to solve a static problem, rather its ability to adapt to changing conditions. Such requirement is likely to define the success or failure in optimization algorithms of tomorrow.

Exact and formal techniques decompose the optimization problems into mathematically tractable problems involving precise assumptions and well-defined problem classes. However, many practical optimization problems are not strictly members of these problem classes, and this becomes especially relevant for problems that are non-stationary during their lifecycle. Traditional deterministic techniques place constraints on the current problem definition and on how that problem definition may change over time. Under these circumstances, long-term algorithm survival / popularity is less likely to reflect the performance of the canonical algorithm and instead more likely reflects success in algorithm design modification across problem contexts [16].

VII. Conclusion

This work describes an innovative Ranked PSO with Lévy flights metaheuristic algorithm for scheduling home appliances, capturing all relevant appliance operations. With appropriately dynamic tariffs, the proposed framework can propose a schedule for achieving cost savings and overloads prevention. Good quality approximate solutions can be obtained in short computational time with almost optimal solutions.

The proposed framework can easily be extended to take into account solar power forecasting in the presence of a residential PV system by simply adapting the objective function and using the solar energy forecaster as further input to the scheduler.

Acknowledgment

This work has been partially supported by INTrEPID, INTelligent systems for Energy Prosumer buildIngs at District level, funded by the European Commission under FP7, Grant Agreement N. 317983.

The authors would like to thank Prof. Della Croce of Operational Research department of the Politecnico di Torino for the valuable insights and contribution on the linear programming solvers.

References

[1] E. Grasso, C. Borean, “QPSOL: Quantum Particle Swarm Optimization with Levy’s Flight,” ICCGI 2014, The Ninth International Multi-Conference on Computing in the Global Information Technology, pp. 14-23.

[2] INTrEPID FP7 project, “INTelligent systems for Energy Prosumer buildings at District level,” http://www.fp7-intrepid.eu .

[3] J. W. Taylor “Short-Term Load Forecasting with Exponentially Weighted Methods,” IEEE Transactions on Power Systems, vol. 27, pp. 458-464, February 2011.

[4] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy, “Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems,” SECON 2010, Boston, MA, June 2010.

[5] Energy@Home project, “Energy@Home Technical Specification version 0.95,” December 22, 2011

[6] R. Kolisch and S. Hartmann, “Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis,” . in J. Weglarz, editor, Project scheduling: Recent models, algorithms and applications, pp. 147–178, Kluwer Academic Publishers, 1999.

[7] R. Kolisch and S. Hartmann, “Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update,” European Journal of Operational Research 174, pp. 23-37, Elsevier, 2006.

[8] K. Cheong Sou, J. Weimer, H. Sandberg, and K. Henrik Johansson, "Scheduling Smart Home Appliances Using Mixed Integer Linear Programming," 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, December 12-15, 2011.

[9] J. Holland, "Adaptation in Natural and Artificial systems," University of Michigan Press, Ann Anbor, 1995.

[10] S. Kirkpatrick, C. D. Gellat, and M.P. Vecchi, "Optimization by Simulated Annealing," Science, 220, pp. 671-680, 1983.

[11] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in: Proc. of the IEEE Int. Conf. on Neural Networks, Perth, Australia, pp. 1942-1948, 1995.

[12] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in IEEE Congress on Evolutionary Computation, pp. 325-31, 2004.

[13] X. Yang, "Nature-Inspired Metaheuristic Algorithms," Luniver Press, 2008.

[14] X. Yang "Review of metaheuristics and generalized evolutionary walk algorithm," Int. J. Bio-Inspired Computation, vol. 3, No. 2, pp. 77-84, 2011.

[15] A. Chechkin, R. Metzler, J. Klafter, V. Gonchar, "Introduction to the theory of lévy flights." In: Klages R, Radons G, Sokolov IM (eds) Anomalous Transport: Foundations and Applications, Wiley-VCH, Berlin, 2008.

[16] J. M. Whitacre "Survival of the flexible: explaining the recent dominance of nature-inspired optimization within a rapidly evolving world," Journal Computing, Vol. 93, Issue 2-4 , pp 135-146 2009.

Contribution of Statistics and Value of Data for the Creation of Result Matrices from Objects of Knowledge Resources

Claus-Peter Rückemann
Westfälische Wilhelms-Universität Münster (WWU),
Leibniz Universität Hannover,
North-German Supercomputing Alliance (HLRN), Germany
Email: ruckema@uni-muenster.de

***Abstract*—This article presents and summarises the main research results on computing optimised result matrices from the practical creation of knowledge resources. With this paper we introduce the main implemented long-term multi-disciplinary and multi-lingual knowledge resources' means, fundamentals and application of documentation, structure, universal classification, and statistics and components for computational workflows and result matrix generation. The resources and workflows can benefit from High End Computing (HEC) resources. The paper presents a knowledge processing procedure using long-term knowledge resources and introduces the n-Probe Parallelised Workflow for an exemplary case study and discussion on a practical application. The goal of this research is to extend the applied features used with long-term knowledge resources' objects and context. The extensions are concentrating on structure and content as well as on processing. The focus is the contribution of statistics and the value of data for the creation of complex result matrices. The major outcome within the last years is the impact on long-term resources based on the scientific results regarding the systematics and methodologies for caring for knowledge.**

Keywords–Knowledge Resources; Processing and Discovery; n-Probe Parallelised Workflow; Universal Decimal Classification; High End Computing.

I. INTRODUCTION

Within the last decades the value of data has steadily increased and with this the demand for flexible and efficient discovery processes for creating results from requests on data sources. The fundamental research on optimising result matrices and statistics has been published and presented at the INFOCOMP conference in June 2014 in Paris [1]. This article presents the extended research, especially focussing on data aspects and practical workflows.

Comparable to statistical models used for on-line text classification [2] even more sophisticated models can be used with advanced, structured, and classified knowledge. These models can be assisted using statistical approaches for data analysis [3] in complex information systems as well as for measuring the reliability of classifications models [4] from the content side. The demand for long-term sustainability of the resources increases with the complexity of content and context. The organisation and structure of the resources are getting essentially important, the more important the more the data sizes and complexity as well as their intelligent use are required [5].

The article therefore introduces and discusses the background, including the systematics and methodologies required for an advanced long-term documentation, which can be deployed in most flexible ways – supported by a comprehensive knowledge definition. The general requirements have to consider the condition that it is not sufficient to support only an isolated or special methodology. The knowledge requires special qualities in order to be usable as well as the quantities of knowledge counts. A suitable general conceptual handling and a universal knowledge definition is required in this environment for supporting advanced workflows in benefit for higher qualities of resulting context and matrices. One the side of methodologies and statistics, some major instruments have been developed and successfully integrated. The combination of instruments and resources allows to flexibly compute optimised result matrices for discovery processes in information systems, expert and decision making system components, search engine algorithms, and last but not least supports the further development of the long-term knowledge resources. The presented results are the outcome of the developments and case studies conducted over the last years.

This paper is organised as follows. Section II discusses the motivation, Section III introduces the available knowledge resources regarding processing, workflows, value of data, and their needs for classification and computing. Section IV presents the details of methodologies and components used as it illustrates the details of the implemented resources' features and procedures, structure and classification, statistics. Section V illustrates the resulting, implemented workflow algorithms, and an example for a parallelised workflow, data-centric parallelisation results, and weighted results from statistics and value of data contributions. Section VI discusses the prominent statistics available and tested with the resources and Section VII shows the implementation results for the matrices on a sample case. Sections VIII and IX evaluate the main results and conclude the presented implementation, also discussing the future work.

II. MOTIVATION

Knowledge resources are the basic components in complex integrated systems. Their target is mostly to create a long-term multi-disciplinary knowledge base for various purposes. Request and selection processes result in requirements for computing result matrices from the available information and data. Optimisation in the context of result matrices means

"improved for a certain purpose". Here, the certain purpose is given by the target and intention of the application scenario, e.g., requests on search results or associations. Therefore, improving the result matrices is a very multi-fold process and "optimising result matrices" primarily refers to the content and context but in second order also to the workflows and algorithms. The major means presented here contributing to the optimisation are classification and statistics, based on the knowledge resources. The employed knowledge resources can provide any knowledge documentation and additional information on objects and knowledge references, e.g., from natural sciences and decision making. Any data used in case studies is embedded into millions of multi-disciplinary objects, including dynamical and spatial information and data files.

It is necessary to develop logical structures in order to govern the existing unstructured and structured big data today and in future, especially in volume, variability, and velocity and to keep the information addressable on long-term. Preparing and structuring big data is the essential process, which has to preceed creating and implementing algorithms. The systematic, methodological, and "clean" big data knowledge preparation and structuring must generally be named as largest achievement in this context and can be considered by far the most significant overall contribution [5]. The creation and optimisation of respective algorithms is of secondary importance, the more the data must be considered for long-term knowledge creation as, e.g., the benefits of most of those implementations depend on a certain generation of computing and storage architectures, which change all few 4–6 years.

III. Knowledge and resources

With the creation of result matrices we have to introduce a common understanding of knowledge and its processing and the value associated with its application.

A. Knowledge definition and understanding

The World Social Science Report 2013 [6] defines knowledge as "The way society and individuals apply meaning to experience ...". Accordingly, the report proposes that "New media and new forms of public participation and greater access to information, are crucial" for open knowledge systems.

In general, we can have an understanding, where knowledge is: Knowledge is created from a subjective combination of different attainments as there are intuition, experience, information, education, decision, power of persuasion and so on, which are selected, compared and balanced against each other, which are transformed and interpreted.

The consequences are: Authentic knowledge therefore does not exist, it always has to be enlived again. Knowledge must not be confused with information or data, which can be stored. Knowledge cannot be stored nor can it simply exist, neither in the Internet, nor in computers, databases, programs or books. Therefore, the demands for knowledge resources in support of the knowledge creation process are complex and multi-fold.

There is no universal "definition" of the term "knowledge", but UDC provides a good overview of the possible width, depth, and facets. For this research the classification references of UDC:0 (Science and knowledge) define the view on universal knowledge [7], which reflects the conceptual dimension and is intended to be used with the full bandwidth of knowledge and knowledge resources.

B. Processing and workflows

Workflows based on the knowledge resources' objects and facilities have been created for different applications. The knowledge resources can make sustainable and vital use of Object Carousels [8] in order to create knowledge object references and modularise the required algorithms [9]. This provides a universal means for improving coverage, e.g., dark data, and quality within the workflow. Secondary resources being available for data, information, and knowledge integration, besides Integrated Information and Computing System (IICS) applications, allow for workflows and intelligent components on High End Computing (HEC) and High Performance Computing (HPC) resources [10], [11]. This paper presents the up-to-date experiences with selected components for structures and workflows.

C. Value of data

The value of data is a central driving force for creating sustainable knowledge resources, the more as data is increasingly important for long period of times. Long-term in cases of sustainable high-value data means many decades of availability and usability. Therefore, usability, security, and archiving are most important aspects of the value of data sets. Value is not the price a data set can be sold as there are many individual factors.

The long-term studies, as the "Cost of Data Breach" study at the Ponemon Institute [12] summarise that the costs related to data loss are high and as predicted [13] do increase [14] every year [15], [16] (sponsored by Symantec), [17] (sponsored by IBM). Straight approaches for calculating individual risks and data loss, as with the Symantec Data Breach Calculator [18] illustrate the effects. Besides science and industry, assessing knowledge loss risks resulting from departing personnel and other factors of loss [19], [20] can be summarised by the risk of knowledge loss, the probability for loss of employees, the consequences of human knowledge loss, and the quality of knowledge resources.

The high quality and value of the knowledge resources used for supporting discovery processes are results of the multi- and trans-disciplinary long-term creation and documentation processes, the structuring of the data, the context of knowledge objects, and the availableness of an universal classification.

D. Knowledge resources

The knowledge resources implement structure and features and can be integrated most flexibly into information and computing system components. Main elements are so called knowledge objects. The objects can consist of any content and context documentation and can employ a multitude of means for description and referencing of objects, data sets, collections, used with computational workflows. Essential core

attributes are a facetted universal classification and various content views and attributes, created manually and automated in interactive and batch operation. Developing workflow implementations for various purposes requires to compute result matrices from the knowledge objects and referred knowledge. The purposes can require individual processing means, complex algorithms, and a base of big data collections. Advanced discovery workflows can easily demand large computational requirements for High End Computing (HEC) resources supporting an efficient implementation.

IV. Methodologies and components employed

The following passages refer to the main components and methodologies and introduce the main aspects for the creation of result matrices.

A. Content, context, and procedures

The data used here is based on the content and context from the knowledge resources, provided by the LX Foundation Scientific Resources [21], [22]. The LX structure and the classification references based on UDC [23], [24] are essential means for the processing workflows and evaluation of the knowledge objects and containers. Both provide strong multi-disciplinary and multi-lingual support. The analysis of different classifications and development of concepts for intermediate classifications from the Knowledge in Motion (KiM) long-term project [25] has contributed to the application of UDC in the context of knowledge resources.

An instructive example for an archaeological and geoscientific use case, deploying knowledge resources, classification, references, and Object Carousels has been recently published [8]. With this research the presentation complements the use case by an important methodology, statistics for intermediate result matrices, usable in any associated workflow. In order to get an overview, the following practical example for a specific workflow as part of an application component shows how result matrices for requests can be computed iteratively.

1) Application component request,
2) Object search (i.e., knowledge objects, classification, references, associations),
3) Creation of intermediate result matrices,
4) Iterative and alternating matrix element creation (i.e., based on intermediate result matrices, object search, referenced content, classification, and statistics),
5) Creation of result matrix,
6) Application component response.

The workflow will mostly be linear if the used algorithms are linear and the data involved is fixed in number and content.

The knowledge objects are under continuous development for more than twenty-five years. The classification information has been added in order to describe the objects with the ongoing research and in order to enable more detailed documentation in a multi-disciplinary and multi-lingual context.

Classification is state-of-the-art with the development of the knowledge resources, which implicitly means that the classification is not created statically or even fixed. It can be used and dynamically modified on the fly, e.g., when required by a discovery workflow description. Representations and references can be handled dynamically with the context of a discovery process. So, the classification can be dynamically modelled with the workflow context. The applied workflows and processing are based on the data and extended features developed for the Gottfried Wilhelm Leibniz resources [26].

Mathematical statistics is a central means for data analysis [27], [28]. It can be of huge benefits when analysing regularities and patterns when used for machine learning with information system components [29]. It is a valuable means deployed in natural sciences and has been integrated in multi-disciplinary humanities-based disciplines, e.g., in archaeology [30]. The span of fields for statistics is not only very broad but statistics itself goes far beyond a simple "tool" status [31].

Methodological means, which have been created in order to be deployed for regular use are workflows improving result quantity and result quality, various filters, universal classifications, statistics applications, manually documented resources' components, integration interfaces for knowledge resources, comparative methods, combination of several means.

The methodologies with the knowledge resources are based on computational methods, processing, classification and structuring of multi-disciplinary knowledge, systematic documentation, long-term knowledge creation, vitality of data concepts, sustainable resources architecture, and collaboration frameworks.

In the past, many algorithms have been developed and implemented [21], [22] for supporting different targets, e.g., silken criteria, statistics, classification, references and citation evaluation, translation, transliteration, and correction support, regular expression based applications, phonetic analysis support, acronym expansions, data and application assignments, request iteration, centralised and distributed discovery, and automated and manual contributions to the workflow.

B. Structure and classification

The key issues for computing result matrices from knowledge resources are that they require long-term tasks on efficiently structuring and classifying content and context. The classification, which has shown up being most important with complex multi-disciplinary long-term classification with practical simple and advanced applications of knowledge resources is the Universal Decimal Classification (UDC) [32].

According to Wikipedia currently about 150,000 institutions, mostly libraries and institutions handling large amounts of data and information, e.g., the ETH Library (Eidgenössische Technische Hochschule), are using basic UDC classification worldwide [33], e.g., with documentation of their resources, library content, bibliographic purposes on publications and references, for digital and realia objects. Just regarding the library applications UDC is present in more than 144,000 institutions and 130 countries [34]. Further operational areas are author-side content classifications and museum collections.

UDC allows an efficient and effective processing of knowledge data. UDC provides facilities to obtain a universal and systematical view on the classified objects. UDC in combination with statistical methods can be used for analysing knowledge data for many purposes and in a multitude of ways.

With the knowledge resources in this research handling 70,000 classes, for 100,000 objects and several millions of referenced data then simple workflows can be linear but the more complex the algorithms get the workflows will mostly become non-linear. They allow interactive use, dynamical communication, computing, decision support, and pre- and postprocessing, e.g., visualisation.

The classification deployed for documentation [35] is able to document any object with any relation, structure, and level of detail as well as intelligently selected nearby hits and references. Objects include any media, textual documents, illustrations, photos, maps, videos, sound recordings, as well as realia, physical objects, such as museum objects. UDC is a suitable background classification, for example:

The objects use preliminary classifications for multi-disciplinary content. Standardised operations used with UDC are coordination and addition ("+"), consecutive extension ("/"), relation (":"), order-fixing ("::"), subgrouping ("[]"), non-UDC notation ("*"), alphabetic extension ("A-Z"), besides place, time, nationality, language, form, and characteristics.

C. Statistics implementation for the knowledge resources

A vast range of statistics, e.g., mathematical statistics, can be deployed based on the knowledge resources. The application of mathematical statistics benefits from an increased number of probes or elements. Probes can result from measurements, e.g., from applied natural sciences and from available material. In many cases, without further analysis a distribution or result may seem random. If the accumulation of an occurrence may indicate a regularity or a rule then this may correlate with a statistical method. Many cases require that statistical results have to be verified for realness. This can be done checking against experience and understanding and using mathematical means, e.g., computing probabilities based on probes.

Statistics have been used for steering the development of the resources. Classification and keyword statistics support the optimisation of the quality of data within the knowledge resources. Counts of terms, references, homophones, synonyms and many more support the improvement of the discovery workflows. Comparisons of content with different language representations increase the intermediate associated result matrices for a discovery process.

The created knowledge resources' architecture is very flexible and efficient because the components allow a natural integration of multi-disciplinary knowledge. The processes of optimising a result matrix differ from a statistical optimisation by the fact that statistics is only one of the factors within the workflows.

V. Implemented knowledge resources' means

The goals for the combination of statistics and classification are, for example:

- Creating and improving result matrices.
- Decision making within workflows.
- Further development of knowledge resources.
- Extrapolation and prediction.

The implementation for the required flexible workflow creation and levels is shown in the following sketch (Figure 1).

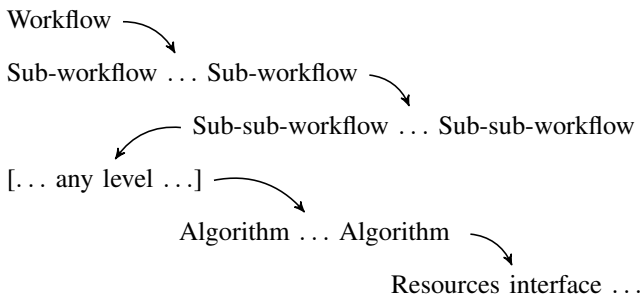


Figure 1. Workflow-algorithm sketch of the implementation (non-hierarchical): Workflow chains, algorithm calls, and resources' interfaces.

The architecture is non-hierarchical. Any workflows can be applied in chains. Each workflow can use sub-workflows, these can use sub-sub-workflows and so on. Each workflow can call or implement algorithms, e.g., for discovery processes, evaluation, and statistics. The workflows and algorithms can use or implement interfaces to the resources. The ellipses indicate that any step can be called or executed in parallel on HEC resources, e.g., in data-parallel or task-parallel processes, in any number of required instances.

An example for this is a "multi-probe parallelised optimisation" workflow, which generates an intermediate result matrix and uses the elements in order to create additional results, all of which are combined for an overall optimised result matrix. The intermediate result matrices are deploying statistical, numerical methods, and various algorithms on base of additional knowledge and information resources.

The knowledge resources allow to implement non-hierarchical and hierarchical architectures. Depending on the workflows these architectures may be created dynamically. Figure 2 shows a workflow-algorithm sketch of a hierarchical implementation based on the resources and emphasizing the methodological aspects.

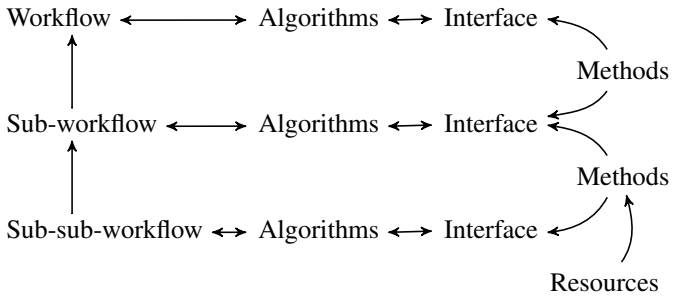


Figure 2. Workflow-algorithm sketch of a hierarchical implementation: Hierarchies of workflows, resources provided by methods.

In this scenario workflows are implemented in a hierarchy of sets workflows, sub-workflows, sub-sub-workflows and so on. Algorithms can be employed by each of these workflows on any level of this hierarchy. The algorithms in turn are connected to the resources through interfaces. The resources can be provided by creating different methods (e.g., static access, dynamic access, batch operation).

A. n-Probe Parallelised Workflow

Computing result matrices can be handled in a multitude of ways. An illustrating example is the n-Probe Parallelised Workflow (nPPW). The workflow is defined by the following steps:

1) A request is started searching for a term called start-element.
2) The search delivers a number of resulting elements, called primary result-elements, being in context with the start-element.
3) The primary result-elements are sorted by a defined attribute (e.g., number of appearance or quality marker).
4) The n most prominent primary result-elements as from the previous step are retained.
5) Secondary requests are started with each of the prominent primary result-elements from the last step.
6) The n most prominent secondary result-elements are gathered for each request, according to the procedure for the primary result-elements.

The workflow is not limited to a single type of elements. Elements can be terms, numbers or other items depending on the use case. For the same reason there is neither a limitation on how to select or weight the elements or which algorithms to use.

The following sketch (Figure 3) demonstrates this at the example of a 5-probe parallelised workflow used for optimisation. In principle, the probes can consist of any type of object, in this example, terms ("T") are used, which are represented by text strings for illustration. c indicates the count for an element in the respective instance while in this example, the absolute count is not in the focus. n indicates the position of the elements.

The "flat search" results in a primary result matrix (Figure 3a) containing terms corresponding with the request for Term 1. In this 5-probe case the resulting primary matrix M0 consists of five elements, Term 1 to Term 5 (dark blue colour).

The "iterative parallelised search instances" in turn get the elements of the result matrix from the flat search as starting seed. In this case, 5-probe means that besides the flat search another four secondary search instances have to be created.

The results of the four secondary requests are secondary result matrices, here, Result Matrix 1 (M1) to Result Matrix 4 (M4) (Figures 3b to 3e). The terms are indexed "Term (m, n)" in short "T (m, n)" with result matrix index m and matrix element index n, starting on the primary matrix at zero.

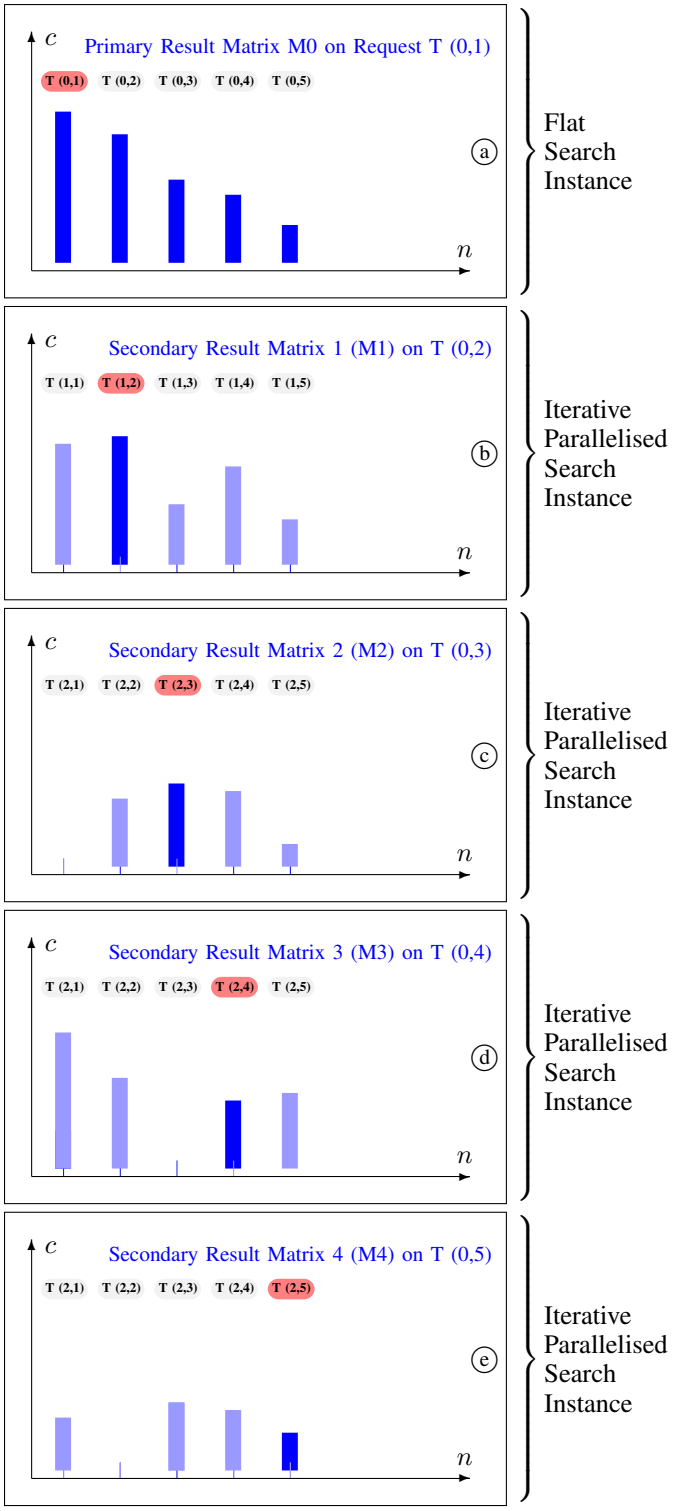


Figure 3. Result matrix creation from a single sub-sub-workflow via intermediate matrices (5-probe parallelised optimisation).

Only those secondary result elements fitting with the original primary elements are considered. The results of the secondary instances are shown in light blue colour. The counts on the

various terms differ significantly. Also some secondary search instance can deliver higher counts on a term than the primary search. The larger the primary result matrix is, the higher the number of required consecutive secondary iterative search instances is. In most cases it is a good approach to parallelise the secondary instances, e.g., depending on the available compute resources. The sum of the secondary instances contribute to the overall workflow with an approachingly linear parallelisation curve for an increasing number of instances. As shown, this approach allows statistical support for the iterations, sub-workflows, and discovery algorithms.

The knowledge resources can contribute to the processes and optimisation with increased numbers of objects and also more structured higher quality data included in the processes. In many cases, e.g., for factual knowledge, manually created components provide the highest values with the optimisation. Hybrid "semi-automatically" and automatically created components especially contribute due to their number, dynamical content, and properties.

B. Data-centric parallelisation results

Common workflows can contain an arbitrary number of result matrix operations. In this simple case the matrix contains 5×5 count elements, which may consist of 5 to 21 different terms. As we want to discuss an elementary set of matrix operations every other operations as considered to be pre- and postprocessing in this case:

- Preprocessing workflow,
- Set of result matrix operations,
- Postprocessing workflow.

The calculation depends on the assumption that the resources can provide a sufficient number elements on a specific request via the workflow algorithms.

The following summary (Table I) shows the consequences with n-probe result matrix operations for different numbers n of elements, with $n_{\max} = (n-1)^2 + n$:

TABLE I. n-PROBE: CONSEQUENCES WITH RESULT MATRIX OPERATIONS FOR DIFFERENT NUMBERS n OF ELEMENTS (5, 10, AND 100,000).

Matrix	*Different Elements*	*Parallelisation ⇒ opt. time fact.*
5×5	5–21	5e; 4c ⇒ n:2
10×10	10–91	10e; 9c ⇒ n:2
100,000×100,000	100,000–9,999,900,001	100,000e: 99,999c ⇒ n:2

That means, the algorithm provides a core set of elements and a larger outer race set of elements, which absolutely and relatively increases with increasing matrix sizes.

For a certain implementation allowing soft criteria for the result matrices the relative and absolute numbers and content of the core and outer race set of elements can be adapted in order to create an implementation scalable in terms of data, architecture, operation. In the example presented here (Figure 3) 5 and 16 are particular numbers.

The "core" cores are a reasonable set of cores, which will contribute to the efficiency of the respective result matrix operation. The outer race cores can be handled very flexible. While different distributions of core and outer race sets can still deliver the same results, e.g., for a given set of knowledge resources, they can especially contribute to the workflow scalability and optimisation process. The parallelisation of n elements ("e") with $n-1$ outer race cores ("c") can improve the speedup from an optimisation time factor n to 2, compared with the non-parallel implementation. For a per-instance-cycle of 1 minute a full multi-parallel cycle takes about 2 minutes. Any lower casts of multitude lead to the respective increase of wall times. Under the assumption that the algorithm is not modified for a set of different constellations of compute resources then the process scales about linear. In general, options for providing computing resources are a fixed number of many cores or a situative number of cores. The workflows in this case can be adapted and react to certain compute and storage architectures, considering the situative "Core number of Cores" and the "Cloud Cores" (2C:2C) for the core and outer race sets. This is even more significant as most workflows can integrate dynamical and intelligent components.

C. Weighted results: Statistics and value

Figure 4 illustrates the weighting of the result elements with the above 5-probe parallelised workflow (Figure 3).

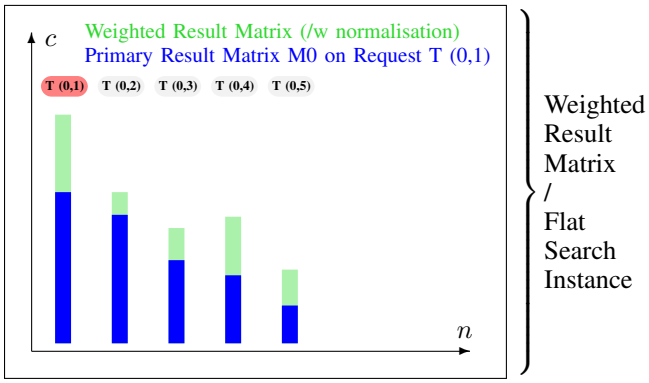


Figure 4. Weighted result matrix (green, without normalisation) creation via intermediate matrices compared to flat search instance (blue), via 5-probe parallelised optimisation.

The weighted result matrix without (/w) normalisation is resulting from the application of the 5-probe parallelised optimisation on the result matrices of the search instances. In this constellation the process is a single sub-sub-workflow, for which we consider the result matrices as intermediate matrices. In contrast to the flat search instance (dark blue colour) the weighted result matrix (green colour) shows different counts. These may consequently result in different priorities and sort orders, as in case of the weighted result for T (0,4) in relation to T (0,3). The weighted priorities and sort orders represent the content and context of the deployed resources, e.g., the asymmetries and references.

The attributes of the content and context can require appropriate algorithms depending of the purposes and workflows for the optimisation. Examples are mean values on counts and fitting to distribution curves with data sets.

D. Statistics and value based on resources

As implementations of statistics are based on counting and numbers the statistics sub-workflows can deploy everything, e.g., any feature or attributes, which can be counted. Sources and means of statistics and computation are:

- Dynamical statistics on the internal and external content and context (e.g., overall statistics, keyword-, categories-, classification-, and media-statistics).
- Mathematics and formula on statistics from the content.
- Elements' statistics (structuring, content, references).
- Statistics based on UDC classification.
- UDC-based statistics computed from comparisons and associations of UDC groups and descriptions.
- Statistics based on any combination of classification, keywords, content, references, context, and computation.

Workflows based on the statistics can be type "semi-manually" or "automated". Besides the major processing and optimisation goals descriptive statistics can be done with each workflow or sub-workflow. Any change of the means supported within a workflow can contribute to the optimisation of the result matrix. Suitable and appropriate means have to be determined for best supporting the goals of the respective step in the workflow. The implementation considers measuring the optimisation by quantity and quality of attributes and features, on intelligence-based and learning processes. With either use there is no general quality measure. Possible quality measures depend on purpose, view, and deployed means. In addition, the decision on these measures can be well supported by statistics, e.g., comparing result matrices from different workflows on the same request. Learning systems components can be used for capturing the success of different measures. The knowledge resources can contain equations and formulary of any grade of complexity. Due to the very high complexity level of the multi-disciplinary components it is necessary to use the basic instances for a comparison in this context of matrix statistics.

The following passages show basic excerpts of statistics objects (LaTeX representation) being part of the implemented knowledge resources. These statistics methods/equations are selected and shown mainly for two reasons: The selected methods are taken from the knowledge objects contained in the resources. These methods are used for result matrix calculations and compared with the evaluation in this research.

VI. Statistics: Fundamentals and application

Statistics on itself can rarely give an overall decisive answer on a question. Statistic means merely can be used as tools for supporting valuations and decisions. Statistics, probability, and distributions are valuable auxiliaries within workflows and integrated application components, e.g., on numbers of objects, spatial or georeferences, phonetic variations, and series of measurement values. Probability and statistics measures are used with integrated applications, e.g., with search requests, with seismic components (e.g., Median and Mean Stacks), which can also be implemented on base of the resources.

A. Basic algorithms applied with knowledge resources

The mean value, arithmetic mean or average M for n values is given by

$$M = \frac{1}{n}\sum_{\nu=1}^{n} x_\nu \tag{1}$$

Calculating the mean value is described by a linear operation. The median value or central value is the middle value in a size-depending sort order of a number of values. For making a statement on the extent of a group of values, the variance ("scattering") can be calculated, with the mean deviation m and the squared mean deviation m^2.

$$m^2 = \frac{1}{n}\sum_{\nu=1}^{n}(x_\nu - M)^2 = \overline{(x-M)^2} \tag{2}$$

For any value this holds $m^2(A) = m^2 + (M-A)^2$. When applying statistics, especially when calculating the propagated error, the following definition of the variance is used:

$$m^2 = \frac{1}{n-1}\sum_{\nu=1}^{n}(x_\nu - M)^2 \tag{3}$$

The mean deviation $\zeta(A)$ is defined as:

$$\zeta(A) = \overline{|x-A|} \text{ for which holds } \zeta(A) = \text{min. for } A = Z \tag{4}$$

The probable deviation or probable error ρ with the probable limits Q_1 and Q_3 is defined as:

$$\rho = \frac{Q_3 - Q_1}{2} \tag{5}$$

The relative frequency h_i is defined as:

$$h_i = \frac{n_i}{n}, \text{ then it holds } \sum_{i=1}^{k} h_i = 1 \tag{6}$$

where n_i is the class frequency, which means the number of elements in a class of which the middle element is x_i.

B. Distributions deployed with knowledge resources

A continuous summation results in the cumulative frequency distribution

$$H_i = \sum_{j=1}^{i} h_j \tag{7}$$

which gives the relative number, for which holds $x \leq x_i \cdot H_i$ is a function discretly increasing from 0 to 1. The presentation results in a summation line. With steady variables, for which at an interval width of Δx the quotient $h_i/\Delta x$ nears a limit, one can calculate a frequency density $h(x_i)$ and for the summation frequency $H(x)$:

$$h(x_i) = \lim_{\Delta x \to 0} \frac{h_i}{\Delta x} \quad \text{and} \quad \frac{dH(x)}{dx} = h(x) \tag{8}$$

With statistical distributions the Gaussian normal distribution is of basic importance.

$$h(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{9}$$

$H(x)$ can not be given "closed". It can be shown that

$$K = \int_{-\infty}^{+\infty} h(x)dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-\frac{1}{2}x^2} dx = 1 \tag{10}$$

The Binominal distribution $w_k(s)$ is defined by

$$w_k(s) = \binom{k}{s} p^s q^{k-s} \tag{11}$$

The sum of the two binominal coefficients is equal to $\binom{k+1}{s}$. This is described by Pascals' Triangle. It holds:

$$M = \sum_{s=0}^{k} w_k(s) \cdot s = kp \quad \text{and} \quad m = \sqrt{kqp} \tag{12}$$

Accordingly, the mean error of the mean value decreases proportional to $1/\sqrt{k}$. This describes the error propagation law.

$$h(X) = \frac{1}{\sqrt{2\pi}m} e^{-\frac{1}{2}\left(\frac{X-M}{m}\right)^2} \text{ for } -\infty < X < +\infty \tag{13}$$

From this Gaussian curves, binominal distributions, correlation coefficients and advanced measures can be developed.

C. Application of fundamental theorems of probability

The probability p is defined by:

$$p(x_i) = \lim_{n\to\infty} h_i = \lim_{n\to\infty} \frac{n_i}{n} \tag{14}$$

The classical definition of p_{classic} is:

$$p_{\text{classic}} = \frac{\text{number of favoured cases}}{\text{number of possible cases}} \tag{15}$$

The following can be said if independence is supposed. $\vee$ means the logical OR, $\wedge$ the logical AND.
Either-OR: If $E_1, E_2, \ldots, E_m$ are events excluding each other and the respective probabilities are $p_1, p_2, \ldots, p_m$, then the probability for *either* E_1 *OR* E_2 *OR* $\ldots$ *OR* E_m is:

$$p(E_1 \vee E_2 \vee \ldots \vee E_m) = p_1 + p_2 + \ldots + p_m \tag{16}$$

As-well-as: If $E_1, E_2, \ldots, E_m$ are event pairwise independent from each other then the probability of E_1 *as well as* E_2 *as well as* $\ldots$ *as well as* E_m is:

$$p(E_1 \wedge E_2 \wedge \ldots \wedge E_m) = p_1 \cdot p_2 \cdot \ldots \cdot p_m \tag{17}$$

VII. Implementation for the result matrix cases

A. Measures for optimisation and purposes

The measures for optimisation are on the one hand object of the services and workflows but on the other hand they can be of concern for the knowledge resources themselves.

Conforming with the goals, measures for optimisation mean fitness for a purpose, e.g., search for a regularity with statistics and result matrices. After a search for regularities any statistical procedure benefits from checking against experiences and associating the procedure and result with a meaning. In many cases, e.g., "relevance" means numbers, uniqueness, proximity for objects, content, and attributes, e.g., terms.

Optimisation can be achieved by various means, e.g., by intelligent selection, by self-learning based optimisation, and by comparisons and statistics. The first measures include manual procedures and essences of results being stored for learning processes. They can also deploy comparisons and statistics, which also mean probability and distributions. This case study is focussed on comparisons and statistics applied with the knowledge resources. The subject of the statistics deals with the collection, description, presentation, and interpretation of data. Especially, the methodology can be based on computing more than the minimal number of comparisons, computing more than the minimal number of distributions, computing result matrices considering the mean of several distributions or extreme distributions. In the case of "relevance", information on weighting may come from sources of different qualities.

The general steps with the knowledge resources, including external sources, can be summarised as: Knowledge resource requests, integrating search engine results (e.g., Google), integrating results more or less randomly, without explicit considerate classification and correlation between content and request, comparing the content of search result matrix elements with the knowledge resources result matrix containing classified elements, statistics on an accumulation of terms, selecting accumulated terms, elimination of less concentrated results, selecting the appropriate number of search results.

B. Sources and Structure: Knowledge resources

The full content, structure, and classification of the knowledge resources have been used. In the context of the case discussed here, the sources, which have been integrated and referenced with the knowledge resources consist of:

- Classical natural sciences data sources.
- Environmental and climatological information.
- Geological and volcanological information.
- Natural and man-made factor/event information.
- Data sets and compilations from natural sciences.
- Archaeological and historical information.
- Archive objects references to realia objects.
- Photo and video objects.
- Dynamical and non-dynamical computation of content.

The sources consist of primary and secondary data and are used for workflows, as far as content or references are accessible and policies, licenses, and data security do not restrict.

C. Classification and statistics in this sample case

Table II shows a small excerpt of resulting main UDC classification references practically used for the statistics with the knowledge resources in the example case presented here.

TABLE II. UNIVERSAL DECIMAL CLASSIFICATION OF STATISTICS FEATURES WITH THE KNOWLEDGE RESOURCES (EXCERPT).

UDC Code	*Description*
UDC:3	Social Sciences
UDC:310	Demography. Sociology. Statistics
UDC:311	Statistics as a science. Statistical theory
UDC:311.1	Fundamentals, bases of statistics
UDC:311.21	Statistical research
UDC:311.3	General organization of statistics. Official statistics
UDC:5	Mathematics. Natural sciences
UDC:519.2	Probability. Mathematical Statistics
UDC:531.19	Statistical mechanics
UDC:570.087.1	Biometry. Statistical study and treatment of biological data
UDC:615.036	Clinical results. Statistics etc.

The small unsorted excerpts of the knowledge resources objects only refer to main UDC-based classes, which for this part of the publication are taken from the Multilingual Universal Decimal Classification Summary (UDCC Publication No. 088) [23] released by the UDC Consortium under the Creative Commons Attribution Share Alike 3.0 license [36] (first release 2009, subsequent update 2012).

As with any object the statistics features can be combined for facets and views for any classification subject. On the other hand statistics objects from the resources can be selected and applied. The listing (Figure 5) shows an excerpt intermediate object result matrix on statistics content.

```
1  ANOVA          [Statistics, ...]:
2                 Analysis of Variance.
3  BIWS           [Whaling]:
4                 Bureau of International Whaling Statistic.
5  GSP            [Geophysics]:
6                 Geophysical Statistics Project.
7  Median         [Statistics]:
8                 In the middle line.
9                 s. also Median-Stack
10 Median-Stack   [Seismics]:
11                Stacking based on the median value of
                   adjacent traces.
12 MSWD           [Mathematics]:
13                Mean Square Weighted Deviation.
14 MSA            [Abbbreviation, GIS]:
15                Metropolitan Statistical Area.
16 MOS            [Abbreviation]:
17                Model Output Statistics.
18 MCDM           [GIS, GDI, Statistics, ...]:
19                Multi-Criteria Decision Making.
20 SHIPS          [Meteorology]:
21                Statistical Hurricane Intensity Prediction
                   Scheme.
22 SAND           [Abbreviation]:
23                Statistical Analysis of Natural resource
                   Data, Norway.
```

Figure 5. Intermediate object result matrix on "statistics" content.

Learning from this: The classifications used for this intermediate matrix are based on contributions from more than one discipline. The elements themselves do not necessarily have to contain a requested term because the classification contributes. Several steps may be necessary in order to improve the matrix, e.g., selecting disciplines, time intervals on the entries, references, and associations. Because different content carries different attributes and features the evaluation can be used in comparative as well as in complementary context.

The implemented knowledge resources means of statistics and computation described above are integrated in the workflows, including classification, dating, and localisation of objects. In addition, probability distributions, linear and non-linear modelling, and other supportive tools are used within the workflow components.

D. Resulting numbers on processing and computing

The processing and computational demands per workflow instance result from the implementation scenarios. The following comparison (Table III) results from a minimal workflow request for a result matrix compared to a workflow request for a result matrix supporting classification views referring to UDC, supporting references and statistics on intermediate results. Both scenarios are based on the same number of elements and entries and can be considered atomic instances in a larger workflow. Views and result matrices can be created manually and automated in interactive and batch operation.

TABLE III. PROCESSING AND COMPUTATIONAL DEMANDS: 2 SCENARIOS, BASED ON 50000 OBJECT ELEMENTS AND 10 RESULT MATRIX ENTRIES.

Scenario Workflow Request for Result Matrix	*Value*
"`geosciences archaeology`" (minimal)	
Number of elements	50,000
Number of result matrix entries (defined)	10
Number of workflow operations	15
Wall time on one core	14 s
"`geosciences archaeology`" (UDC, references, statistics)	
Number of elements	50,000
Number of result matrix entries (defined)	10
Number of workflow operations	6,500
Wall time on one core	6,700 s

As the discussed scenarios are instances this means workflows based on n of these instances will at least require n-times the time for an execution on the same system. It must be remembered that the parallelisation will have a significant effect when workflows are created based on many of these instances when required in parallel. Without modifying the algorithms of the instances, which mostly means simplifying, the positive parallelisation effect for the workflows can be nearly linear. Besides the large requirements per instance with most workflows there are significant beneficial effects from parallelising even within single instances as soon as the number of comparable tasks based on the instances increases. A typical case where parallelisation within a workflow is favourable is the implementation of an application creating result matrices and being used with many parallel instances, e.g., with providing services. The number of 70,000 elementary UDC classes currently results in 3 million basic elements when only considering multi-lingual entries – without any

combinations. With most isolated resources only several thousand combinations are used in practice each. The variety and statistics are mostly deployed for decision processing, increasing quantity, and increasing quality. Many of the above cases require to compute more than one data-workflow set to create a decision. A review and an auditing process are mandatory for mission critical applications. The computational requirements can increase drastically with the computation of multiple workflows. Each workflow will consist of one or more processes, which can contain different configurations and parameters. Therefore, creating a base for an improved result matrix starts with creating several intermediate result matrices. With a ten process workflow, e.g., the possible configurations and parameters can easily lead to computing a reasonable set of thousands to millions of intermediate result matrices.

The objects and methods used can be long-term documented as knowledge objects. Nevertheless, there is explicitely no demand for a certain programming language. Even multiple implementations can be done with any object. The workflows and algorithms with the cases discussed here have been implemented as objects in Fortran, Perl, and Shell. Anyhow, the implementation of algorithms is explicitely not part of any core resources. It is the task of anyone having an application to do this and to decide on the appropriate means and methods.

E. Complementary Components

As an example we choose to mention three state-of-the-art components for implementing the "data-base", operating system, and distributed platform. With this it should be possible to build and use containers. For implementation of very simple non-hierarchical but data-set centred scenarios the MongoDB [37] concept may be used. This database model greps the concept of a data-set centred approach and extends the traditional database models. CoreOS [38] can be used for data-warehouse style computing, providing and operating system for massive server deployment. In addition, Docker [39] can be used, an open platform for distributed applications, which shall enable to build, ship and run applications anywhere.

Anyhow, these components are not data-centric themselves. It is also more than questionable if data can be sustainably preserved in close integration with these components, even for mid-term purposes of a few decades only.

The knowledge resources, including their creation and further development, should be kept in a long-term and portable concept, as an implementation based on such above components has shown to be still much too application centric.

VIII. Case results and evaluation

Computing result matrices is an arbitrary complex task, which can depend on various factors. Applying statistics and classification to knowledge resources has successfully provided excellent solutions, which can be used for optimising result matrices in context of natural sciences, e.g., geosciences, archaeology, volcanology or with spatial disciplines, as well as for universal knowledge. The method and application types used for optimisation imply some general characteristics when putting discovery workflows into practice regarding components like terms, media, and other context (Table IV).

Table IV. Resulting per-instance-calls for method and application types on optimisation with knowledge discovery.

Type	*Terms*	*Media*	*Workflow*	*Algorithm*	*Combination*
Mean	500	20	20	50,000	3,000
Median	10	5	2	5,000	50
Deviation	30	5	5	200	20
Distribution	90	40	15	20	120
Correlation	15	10	5	20	90
Probability	140	15	20	50	150
Phonetics	50	5	10	20	50
Regular expr.	920	100	50	40	1,500
References	720	120	30	5	900
Association	610	60	10	5	420
UDC	530	120	20	5	660
Keywords	820	100	10	5	600
Translations	245	20	5	5	650
Corrections	60	10	5	5	150
External res.	40	30	5	5	40

Statistics methods have shown to be an important means for successfully optimising result matrices. The most widely implemented methods for the creation of result matrices are intermediate result matrices based on regular expressions and intermediate result matrices based on combined regular expressions, classification, and statistics, giving their numbers special weight. Based on these per-instance numbers this results in demanding requirements for complex applications – On numerical data: Millions of calls are done per algorithm and dataset, hundreds in parallel/compact numeric routines. On "terms": Hundred thousands of calls are done per sub-workflow, thousands in parallel/complex routines, are done.

Most resources are used for one application scenario only. Only 5–10 percent overlap between disciplines – due to mostly isolated use. Large benefits result from multi-disciplinary multi-lingual integration. The multi-lingual application adds an additional dimension to the knowledge matrix, which can be used by most discovery processes. As this implemented dimension is of very high quality the matrix space can benefit vastly from content and references.

IX. Conclusion and future work

This paper presented the extended research, focussing on data aspects and practical workflows, based on the fundamental research on optimising result matrices from knowledge discovery workflows. This research has extended the applied features used with long-term knowledge resources' objects and context. Starting with the multi-disciplinary and multi-lingual knowledge resources examples for non-hierarchical and hierarchical workflows have been presented.

First, knowledge resources' objects with their structured content, references, and conceptual knowledge are providing an excellent means for long-term multi-disciplinary and multi-lingual documentation and reuse. This especially includes the flexible universal classification of any objects. The quality of

data can be used to contribute to the discovery and optimisation processes, which increases the emphasis on the values of data the more the long-term significance gets into the focus.

Second, the use of statistics and algorithms based on statistics has shown to provide solid tools for creating and improving result matrices. Both, the documentation and resources and the statistics applicable in workflows result in benefits for complex result matrix generation.

The case study introduced the application of n-Probe Parallelised Workflows, which can be used for result matrix generation. The matrix generation and processes have been discussed in detail. Workflows like these have been successfully used for the optimisation of result matrices. They allow to use statistics methods and data value weighting and can contribute to the creation and development of resources. A number of structuring elements and workflow procedures have been successfully implemented for processing objects from knowledge resources, which allow optimising result matrices in very flexible ways. Long-term multi-disciplinary and multi-lingual knowledge resources can provide a solid source of structured content and references for a wealth of result matrices. The long-term results confirm that for the usability the organisation of the content and the data structures are most important and should have the overall focus compared to algorithm adaptation and optimisation. Nevertheless, the computational requirements may be very high but compared against the long-term data creation issues, they should be regarded secondary from the scientific point of view. Employing a classification like UDC has shown to be a universal and most flexible solution with statistics for supporting long-term multi-disciplinary knowledge resources. Computing optimised result matrices from objects of universally classified knowledge resources can be efficiently supported by various statistics and probability measures. With the quality and quantity of matrix elements this can also improve the decision making processes within the workflows.

The research conducted provided that advanced discovery will have to go into depth as well as into broad surface of the context of the multi-disciplinary and multi-lingual information in order to effectively improve the quality for most workflows. Many of these workflow processes can be very well parallelised on HEC resources. A typical case where parallelisation is required is the implementation of an application creating result matrices and used with many parallel instances. This introduces benefits for the applicability of the discovery facing big data resources to be included. The integration of the above strategies and means has proven an excellent method for computing optimised result matrices.

On the computational side, the workflows contribute to the parallelisation of processes and result in higher scalability regarding data resources, architectures, and operation. Therefore, the resources and processing workflows can benefit from a flexible deployment of High End Computing resources. The major outcome on the content side is the impact on long-term resources based on the scientific results regarding the systematics and methodologies for caring for knowledge.

Besides all future application scenarios, the further creation of development of content and context and its documentation is a main goal. Future work will be focussed on the workflow processes and standardisation and best practice for container and resources' objects but also concentrate on the development of flexible structures for objects and the automation of processes.

ACKNOWLEDGEMENTS

We are grateful to all national and international partners in the GEXI cooperations for their support and contributions. We thank the Science and High Performance Supercomputing Centre (SHPSC) for long-term support of collaborative research since 1997, including the GEXI developments and case studies on archaeological and geoscientific information systems. Special thanks go to the scientific colleagues at the Gottfried Wilhelm Leibniz Bibliothek (GWLB) Hannover, especially Dr. Friedrich Hülsmann, for collaboration and prolific discussion within the "Knowledge in Motion" project, for inspiration, and practical case studies. Many thanks go to the scientific colleagues at the Leibniz Universität Hannover, especially to Dipl.-Biol. Birgit Gersbeck-Schierholz and to Dr. Bernhard Bandow, Max Planck Institute for Solar System Research (MPS), Göttingen, for their collaboration and discussion on non-hierarchical and hierarchical workflows, as well as to the scientific colleagues at the Institute for Legal Informatics (IRI), Leibniz Universität Hannover, and to the Westfälische Wilhelms-Universität (WWU), for discussion, support, and sharing experiences on collaborative computing and knowledge resources and for participating in fruitful case studies as well as to my students and participants of the postgraduate European Legal Informatics Study Programme (EULISP) for prolific discussion of scientific, legal, and technical aspects over the last years.

REFERENCES

[1] C.-P. Rückemann, "Computing Optimised Result Matrices for the Processing of Objects from Knowledge Resources," in Proceedings of The Fourth International Conference on Advanced Communications and Computation (INFOCOMP 2014), July 20–24, 2014, Paris, France. XPS Press, 2014, pages 156–162, ISSN: 2308-3484, ISBN-13: 978-1-61208-365-0, URL: http://www.thinkmind.org/index.php?view=article&articleid=infocomp_2014_7_20_60039 [accessed: 2015-02-01].

[2] P. Cerchiello and P. Giudici, "Non parametric statistical models for online text classification," Advances in Data Analysis and Classification – Theory, Methods, and Applications in Data Science, vol. 6, no. 4, 2012, pp. 277–288, special issue on "Data analysis and classification in marketing" Baier, D. and Decker, R. (guest eds.) ISSN: 1862-5347 (print), ISSN: 1862-5355 (electronic).

[3] W. Gaul and M. Schader, Eds., Classification As a Tool of Research. North-Holland, Amsterdam, 1986, Proceedings, Annual Meeting of the Classification Society, (Proceedings der Fachtagung der Gesellschaft für Klassifikation), ISBN-13: 978-0444879806, ISBN-10: 0-444-87980-3, Hardcover, XIII, 502 p., May 1, 1986.

[4] J. Templin and L. Bradshaw, "Measuring the Reliability of Diagnostic Classification Model Examinee Estimates," Journal of Classification, vol. 30, no. 2, 2013, pp. 251–275, Heiser, W. J. (ed.), ISSN: 0176-4268 (print), ISSN: 1432-1343 (electronic), URL: http://dx.doi.org/10.1007/s00357-013-9129-4 [accessed: 2015-02-01].

[5] A. Woodie, "Forget the Algorithms and Start Cleaning Your Data," Datanami, 2014, March 26, 2014, URL: http://www.datanami.com/datanami/2014-03-26/forget_the_algorithms_and_start_cleaning_your_data.html [accessed: 2015-02-01].

[6] World Social Science Report 2013, Changing Global Environments, 1st ed. Published jointly by the United Nations Educational, Scientific and Cultural Organization (UNESCO), the International Social Science Council (ISSC), the Organisation for Economic Co-operation and Development (OECD), 2013, DOI: 10.1787/9789264203419-en, OECD ISBN 978-92-64-20340-2 (print), OECD ISBN 978-92-64-20341-9 (PDF), UNESCO ISBN 978-92-3-104254-6 (PDF and print).

[7] C.-P. Rückemann, "From Multi-disciplinary Knowledge Objects to Universal Knowledge Dimensions: Creating Computational Views," International Journal On Advances in Intelligent Systems, vol. 7, no. 3&4, 2014, pp. 385–401, ISSN: 1942-2679, LCCN: 2008212456 (Library of Congress).

[8] C.-P. Rückemann, "Sustainable Knowledge Resources Supporting Scientific Supercomputing for Archaeological and Geoscientific Information Systems," in Proceedings of The Third International Conference on Advanced Communications and Computation (INFOCOMP 2013), November 17–22, 2013, Lisbon, Portugal. XPS Press, 2013, pp. 55–60, ISSN: 2308-3484, ISBN: 978-1-61208-310-0, URL: http://www.thinkmind.org/download.php?articleid=infocomp_2012_3_10_10012 [accessed: 2015-02-01].

[9] C.-P. Rückemann, "High End Computing for Diffraction Amplitudes," in The Third Symposium on Advanced Computation and Information in Natural and Applied Sciences, Proceedings of The 11th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 21–27, 2013, Rhodes, Greece, Proceedings of the American Institute of Physics (AIP), AIP Conference Proceedings, vol. 1558. AIP Press, 2013, pp. 305–308, ISBN: 978-0-7354-1184-5, ISSN: 0094-243X, DOI: 10.1063/1.4825483.

[10] U. Inden, D. T. Meridou, M.-E. C. Papadopoulou, A.-C. G. Anadiotis, and C.-P. Rückemann, "Complex Landscapes of Risk in Operations Systems Aspects of Processing and Modelling," in Proceedings of The Third International Conference on Advanced Communications and Computation (INFOCOMP 2013), November 17–22, 2013, Lisbon, Portugal. XPS Press, 2013, pp. 99–104, ISSN: 2308-3484, ISBN: 978-1-61208-310-0, URL: http://www.thinkmind.org/download.php?articleid=infocomp_2013_5_10_10114 [accessed: 2015-02-01].

[11] P. Leitão, U. Inden, and C.-P. Rückemann, "Parallelising Multi-agent Systems for High Performance Computing," in Proceedings of The Third International Conference on Advanced Communications and Computation (INFOCOMP 2013), November 17–22, 2013, Lisbon, Portugal. XPS Press, 2013, pp. 1–6, ISSN: 2308-3484, ISBN: 978-1-61208-310-0, URL: http://www.thinkmind.org/download.php?articleid=infocomp_2013_1_10_10055 [accessed: 2015-02-01].

[12] Ponemon Institute, "Data-Security," 2014, Ponemon Institute, URL: http://www.ponemon.org/data-security [accessed: 2015-02-01].

[13] C.-P. Rückemann, "High End Computing Using Advanced Archaeology and Geoscience Objects," International Journal On Advances in Intelligent Systems, vol. 6, no. 3&4, 2013, pp. 235–255, iSSN: 1942-2679, URL: http://www.iariajournals.org/intelligent_systems/intsys_v6_n34_2013_paged.pdf [accessed: 2015-02-01].

[14] Ponemon Institute, "Ponemon Study Shows the Cost of a Data Breach Continues to Increase," 2014, Ponemon Institute, URL: http://www.ponemon.org/news-2/23 [accessed: 2015-02-01].

[15] Ponemon Institute, "Cost of Data Breach 2011," 2011, Ponemon Institute / Symantec, URL: http://www.ponemon.org/library/archives/2012/03 [accessed: 2015-02-01].

[16] Ponemon Institute, "2013 Cost of Data Breach: Global Analysis," 2013, Ponemon Institute / Symantec, URL: http://www.ponemon.org/local/upload/file/2013%20Report%20GLOBAL%20CODB%20FINAL%205-2.pdf [accessed: 2015-02-01].

[17] Ponemon Institute, "2014 Cost of Data Breach: Global Analysis," May 2014, Ponemon Institute / IBM, Ponemon Institute (c) Research Report, Benchmark research sponsored by IBM, Independently conducted by Ponemon Institute LLC, IBM Document Number: SEL03027USEN, URL: http://www.ibm.com/services/costofbreach [accessed: 2015-02-01], URL: http://public.dhe.ibm.com/common/ssi/ecm/en/sel03027usen/SEL03027USEN.PDF [accessed: 2015-02-01].

[18] Symantec, "Symantec Data Breach Calculator," 2014, Symantec, URL: https://databreachcalculator.com/ [accessed: 2015-02-01].

[19] "Wissensverlust vermeiden beim Abgang von Wissensarbeitern," library essentials, LE_Informationsdienst, Juni/Juli 2014, 2014, pp. 9–11, ISSN: 2194-0126, URL: http://www.libess.de [accessed: 2015-02-01].

[20] M. E. Jennex, "A Proposed Method for Assessing Knowledge Loss Risk with Departing Personnel," vol. 44, no. 2, 2014.

[21] "LX-Project," 2014, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/en/#LX (Information) [accessed: 2015-02-01].

[22] C.-P. Rückemann, "Enabling Dynamical Use of Integrated Systems and Scientific Supercomputing Resources for Archaeological Information Systems," in Proc. INFOCOMP 2012, Oct. 21–26, 2012, Venice, Italy, 2012, pp. 36–41, ISBN: 978-1-61208-226-4.

[23] "Multilingual Universal Decimal Classification Summary," 2012, UDC Consortium, 2012, Web resource, v. 1.1. The Hague: UDC Consortium (UDCC Publication No. 088), URL: http://www.udcc.org/udcsummary/php/index.php [accessed: 2015-02-01].

[24] "UDC Online," 2014, URL: http://www.udc-hub.com/ [accessed: 2015-02-01].

[25] F. Hülsmann and C.-P. Rückemann, "Value of Data and Long-term Knowledge," KiMrise, Knowledge in Motion, August 12, 2014, 10 Year Anniversary Workgroup Meeting, "Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF)", Hannover, Germany, 2014.

[26] C.-P. Rückemann, "Archaeological and Geoscientific Objects used with Integrated Systems and Scientific Supercomputing Resources," International Journal on Advances in Systems and Measurements, vol. 6, no. 1&2, 2013, pp. 200–213, ISSN: 1942-261x, LCCN: 2008212470 (Library of Congress), URL: http://www.thinkmind.org/download.php?articleid=sysmea_v6_n12_2013_15 [accessed: 2015-02-01], URL: http://lccn.loc.gov/2008212470 [accessed: 2015-02-01].

[27] Y. Dodge, The Oxford Dictionary of Statistical Terms. Oxford University Press, 2006, ISBN: 0-19-920613-9.

[28] B. S. Everitt, The Cambridge Dictionary of Statistics, 3rd ed. Cambridge University Press, Cambridge, 2006, ISBN: 0-521-69027-7.

[29] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006, ISBN: 0-387-31073-8.

[30] R. D. Drennan, Statistics in Archaeology. Elsevier Inc., 2008, in: Pearsall, Deborah M. (ed.), Encyclopedia of Archaeology, pp. 2093-2100, Elsevier Inc., ISBN: 978-0-12-373962-9.

[31] D. Lindley, "The Philosophy of Statistics," Journal of the Royal Statistical Society, 2000, JSTOR 2681060, Series D 49 (3), pp. 293–337, DOI: 10.1111/1467-9884.00238.

[32] "Universal Decimal Classification Consortium (UDCC)," 2014, URL: http://www.udcc.org [accessed: 2015-02-01].

[33] "Universal Decimal Classification (UDC)," 2015, Wikipedia, URL: http://en.wikipedia.org/wiki/Universal_Decimal_Classification [accessed: 2015-02-01].

[34] A. Slavic, "UDC libraries in the world - 2012 study," universaldecimalclassification.blogspot.de, 2012, Monday, 20 August 2012, URL: http://universaldecimalclassification.blogspot.de/2012/08/udc-libraries-in-world-2012-study.html [accessed: 2015-02-01].

[35] C.-P. Rückemann, "Integrating Information Systems and Scientific Computing," International Journal on Advances in Systems and Measurements, vol. 5, no. 3&4, 2012, pp. 113–127, ISSN: 1942-261x, LCCN: 2008212470 (Library of Congress), URL: http://www.thinkmind.org/index.php?view=article&articleid=sysmea_v5_n34_2012_3/ [accessed: 2015-02-01].

[36] "Creative Commons Attribution Share Alike 3.0 license," 2012, URL: http://creativecommons.org/licenses/by-sa/3.0/ [accessed: 2015-02-01].

[37] "MongoDB," 2015, URL: http://www.mongodb.org/ [accessed: 2015-02-01].

[38] "CoreOS," 2015, URL: https://coreos.com/ [accessed: 2015-02-01].

[39] "Docker," 2015, URL: https://www.docker.com/ [accessed: 2015-02-01].

Optimizing Early Detection of Production Faults by Applying Time Series Analysis on Integrated Information

Thomas Leitner* and Wolfram Wöß†
Institute for Application Oriented Knowledge Processing, Johannes Kepler University Linz
Altenberger Straße 69, Linz, Austria
Email: *thomas.leitner@jku.at, †wolfram.woess@jku.at

***Abstract*—According to the *Industry 4.0* initiative, industry aims for total automation and customizability using sensors for data retrieval, computer systems such as clusters and cloud services for large-scale processing, and actuators to react in the production environment. Additionally, the automotive industry is focusing increasingly on gathering information from the after-sales market using sensors and diagnostic mechanisms. All this information enables more accurate classification of faults when cars malfunction or exhibit undesired behaviour. Since finding systematic faults as quickly as possible is key to maintaining a good reputation and reducing warranty costs, techniques must be established that recognize increasing occurrences of fault types at the earliest possible point in time. Several sources of information exist that store heterogeneous datasets of varying quality and at various stages of approval. Using as much data as possible is fundamental for accurately detecting critical developing faults. In order to appropriately support the combination of these different datasets, information should be treated differently depending on its data quality. To this end, a concept to optimizing early fault detection consisting of four components is proposed, each of them with a particular goal; (i) determination of data quality metrics of different datasets storing warranty data, (ii) analysis of univariate time series to generate forecasts and the application of linear regression, (iii) weighted combination of course parameters that are calculated using different predictions, and (iv) improvement of the system accuracy by integrating prediction errors. This concept can be employed in various application areas where multiple datasets are to be analyzed using data quality metrics and forecasts in order to identify negative courses as early as possible.**

Keywords–data mining; time series analysis; data quality metrics; automotive industry.

I. Introduction

In recent years the capabilities of storing large data volumes that originate in various industries, ranging from the manufacturing industry to social web companies lie beyond the possibilities of analyzing them. From the management perspective, information hidden in raw data from various data sources provides decision support and guidance, and is therefore gaining importance. In order to draw reliable conclusions, new sophisticated ways of processing these large datasets are required.

In cooperation with the industrial partner *BMW Motoren GmbH* (engine manufacturing plant), located in Steyr, Austria, the *Quality - abnormality and cause analysis* (Q-AURA) application has been developed and is currently being improved and optimized. The core functionality of Q-AURA is to shorten the problem-solving time for finding causes of automobile engine faults in the after-sales market. The system consists of several components that support the quality management experts in their daily work. The first step in the Q-AURA analysis process is to find significant faults (i.e., those with negative consequences) to determine which fault types are occurring increasingly in the after-sales market. The result is a set of significant faults, which are analyzed further by calculating histograms and attribute distributions of engines that are affected by the same fault type in order to identify similarities between them. The last step is to analyze bills of materials (BOM) consisting of engine parts, components, and technical modifications from the development department to determine a set of modifications, that is the most likely cause of a particular fault. Q-AURA was evaluated positively and is already being used by quality management experts in their daily work. Although it delivers good results, improvements are being considered to further enhance Q-AURA's functionality. Currently, the application uses only one dataset from one warranty information system to determine critical developing (significant) faults. Since datasets residing in other information sources store warranty and after-sales information at various stages of approval, an extension is needed that integrates them into Q-AURA. This would provide an improved overall view of real-world situations and allow techniques to be used that help to find significant faults earlier, but must be thoroughly validated to achieve robust results [1].

This paper focuses on a concept that uses data quality metrics to determine dataset quality, time series analysis including forecasting methods to reveal trends and predict future values, and weighting mechanisms for optimized combination of multiple datasets. The structure of this paper is as follows: Section II describes the requirements for such a concept and the associated research issues and challenges. Section III gives an overview of related approaches and describes different methods and mechanisms that are addressed and used by the proposed concept. Subsequently, Section IV introduces Q-AURA, details the proposed improvement, and presents its integration into the Q-AURA analysis process. Finally, Section V concludes the paper, providing an outlook on future improvements.

II. Research Issues and Challenges

As mentioned in Section I, the overall goal of the proposed concept is to earlier identify significant faults, which required rethinking the Q-AURA concept. Currently, only historic customer claims (from the last six weeks) are used to determine whether faults are significant. However, improving the approach requires not only data from previous weeks, but also predicting future values. Calculating future values based on past observations is challenging, because it introduces some degree of uncertainty. Therefore, we propose using multiple datasets to improve the prediction process. In detail

the proposed approach consists of four tasks, each of which addresses a particular challenge.

The first task is to *validate each dataset, which stores partially contradictory, complementary, and/or redundant information*. The business process addressed by Q-AURA begins in the early development phase and comprises the development of new, and the improvement of existing, benzine and diesel engine generations. The process ends in the after-sales market, where information about warranty claims and data generated during a car's usage is stored. If a customer experiences a particular fault, the car must be checked at a dealer's workshop. There, information about the car and the fault are retrieved and sent to the car manufacturer. Since BMW sells cars in many countries partly different classifications of faults exist, which may lead to discrepancies. These must be addressed and a solution found to obtain a correct and consistent overall view of the fault data. Additionally, datasets exist that store information at different stages of approval. Data quality metrics must be defined to determine numeric criteria that can be interpreted and used in further processing steps.

The second task deals with the challenge of *detecting critically developing faults as early as possible using time series and regression analysis*. This is important because each week a critical developing fault is identified earlier reduces warranty costs and simultaneously enhances customer satisfaction. In the current Q-AURA implementation regression analysis is performed on data from the latest six weeks, and thresholds are then applied to regression parameters to determine whether the fault is significant. This time period (of six weeks) proved to provide the best trade-off between early detection (using as few recent weeks as possible) and robustness. The only way to improve the concept was therefore to incorporate predictions. Forecasting methods are used to compute the most likely future performance, which is then used as input to regression analysis. The most suitable time series and prediction methods were evaluated and selected.

The third task concerns the development of a *verification and weighting mechanism to determine how different datasets should be treated in the analysis process*. Since multiple datasets are used to obtain more robust results, the best way of combining them must be found. Two types of weighting factors are central to the proposed concept: those based on the overall data quality metric of each dataset and those based on the prediction accuracy of a particular fault's course. The prediction accuracy can be calculated using the prediction of the previous week and the observation of the current week. The proposed concept also defines how the weighting factors are used to combine the data from these different datasets to finally determine whether an analyzed fault is significant or not.

The last task is the *integration of the invented concept into the Q-AURA application and its verification*. Therefore, the new Q-AURA concept is described to demonstrate the benefit of the improved approach.

The resulting approach consists of a set of methods that enables earlier detection of badly developing fault courses. First, data quality metrics of different datasets are calculated, which are then used for weighting. Time series analysis using forecasting mechanisms is applied to predict future values based on historical data from these different information systems. The prediction accuracy is determined on the basis of the following week's observation, which is also used as a weighting factor for the datasets. Finally, the calculated weighting factors and the regression parameters are used to determine the significance of a particular fault.

III. Related Work

This section presents related approaches, information about the methods applied and an overview of the concepts on which the proposed approach is based under three different headings: (i) data quality metrics including their assessment, (ii) analysis of univariate time series, and (iii) determination of forecasting accuracy.

A. Related Approaches

The proposed concept is tailored to the particular needs related to identifying significant faults using time series analysis, forecasts, and regression analysis based on data from multiple information systems. Other approaches exist that focus on similar topics.

Chan et al. [2] presented a case study of predicting future demands in inventory management. They focused on combining different forecasts to improve prediction accuracy compared to only using one forecast. Their approach differs from the presented approach in several aspects: it seems to use only one information source as dataset, applies different time series forecasting methods and calculates weighting factors based on the results of those different methods. A major difference between the introduced concept and the approach used by Chan et al. is that the presented concept is based on different data sources. Thus, different time series are generated leading to different predictions. Further, the combination step of the proposed approach is not carried out at the level of the forecast result, but later after the data has been evaluated. The results are then weighted based not only on accuracy metrics of the forecasts, but also on the data quality of the particular information source.

Research by Widodo and Budi [3] focused on predicting the yearly passenger number for six consecutive years using 11 time series. Their approach uses the mean squared error (MSE) to compare prediction accuracy. In their research work the comparison of forecasts is done using the same dataset. The following points distinguish the proposed approach from theirs: More than one dataset is used in the presented concept. The forecasts are calculated separately for each data source with the same forecasting method and are combined after evaluation. In the proposed concept the different forecasts are combined using two types of weighting factors, (i) weights based on the prediction error, and (ii) weights based on data quality.

In [4], the authors described a method for analyzing the lifetime of products using Weibull distributions. Their application area is focused on electronic components in the automotive industry. The approach employs a day-in-service metric to identify the potential lifetime of the products analyzed. Day-in-service specifies how long a product has already been in use. In the automotive industry the day-in-service metric usually begins with the delivery of the car to the customer. The *bathtub* reliability curve is used, representing lower reliability at the beginning and the end of product life. Their approach has a different objective than the proposed one: They want to know

how long the majority of components will survive before they fail. Hence, they are not interested in what (fault type) occurred and how it developed in recent weeks, but how reliable the products are across all fault types.

Montgomery et al. [5] published a detailed paper about combining forecasts from different methods, and particularly how they can be weighted for the best results in social sciences. They proposed an enhancement to the *ensemble Bayesian model averaging (EBMA)* method that improves accuracy and performance for social science applications. They evaluated their approach in two use cases: prediction of (i) the 2012 US election and (ii) the development of the US unemployment rate. EBMA is mentioned in various research papers and has proved useful in combining different prediction methods. Since the proposed concept in this research work integrates different datasets, data quality metrics must be used, as the quality of those different sources may vary. Also, it has to be outlined that the combination task is performed on the regression analysis parameters of each dataset, which is necessary to identify whether a particular fault is significant or not.

Armstrong [6] published an overview of requirements and the possible ways of combining forecasting methods. Various approaches were analyzed, and it was emphasized that the combination can be achieved using different forecasting methods, different datasets or both. When multiple datasets are analyzed, their heterogeneity may require that more than one forecasting method is used. The approaches investigated address similar use cases, since they seek to improve prediction accuracy using multiple datasets or methods. However, unlike the proposed one, none of these approaches implements a two-step method that uses weighting factors based on data quality evaluation for each dataset and regression analysis (including computed predictions) to determine a significant course.

B. Data Quality

Previously, the data quality of information stored in databases or data warehouses had often been neglected. Redman [7] described the impact of poor data quality at different levels of decision-making and the ensuing problems. Considerable effort has since been put into enhancing data quality and quality assurance, but there remains room for improvement; information derived from data in information systems continues to be of lower quality than expected. Heinrich et al. [8] presented statistics that show various problems due to poor data quality, and mentioned that awareness must be raised.

In many cases decision-makers do not know that the data from a particular information source is of poor quality [7]. Thus, not only must the quality of the stored data be improved as much as possible, but users of this data must be made aware that it is not completely reliable. In modern businesses, many automated procedures and processes exist that transform and aggregate stored data, and compute new values which are then used by other processes to derive and generate new information, decision parameters, and other content. Clearly, if the original data is of poor quality, all the workflows and subsequent processes that use this information generate even poorer results, which may lead to problems, incorrect decisions, or other negative consequences. Hence, it would be advisable that these workflows should not rely solely on the data assuming it is completely correct, but to use quality metrics that determine the level of uncertainty. If multiple information systems exist that store partially redundant information originating from different processes, subsequent processes can use all data from all systems to achieve a better overall view. In order to know how to treat information from these information systems, methods are required that consider and measure the quality of stored data. In the scientific literature, a variety of data quality metrics and dimensions have been defined and specified, each of them tackling a particular aspect of data quality [9][10]. Wang and Strong [11] defined the term data quality dimension as a set of data quality attributes that define a single aspect or construct of data quality. They aimed to categorize data quality metrics in terms of *accuracy of data*, *relevancy of data*, *representation of data*, and *accessibility of data*, while in [12] and [13] the classes were labeled *intrinsic data quality*, *accessibility data quality*, *contextual data quality*, and *representation data quality*. Naumann and Rolker [14] based their distinction on the usage and retrieval process of information, dividing data quality metrics into *subject-criteria scores*, *process-criteria scores*, and *object-criteria scores*. Other publications, among them [15] and [16], investigated dependencies and tradeoffs between data quality metrics. Note that data quality metrics can be determined in a task-dependent and a task-independent manner, depending respectively on whether they are computed with or without the contextual knowledge of their usage [17]. Such context can be included, for instance, by applying business rules or government regulations. Bobrowski et al. [18] distinguished between direct and indirect metrics, where the former are determined directly from the data, and the latter are computed from the former, taking the dependencies between them into account.

In accordance with these classifications, those data quality metrics that are important in the context of the proposed approach are identified and described below. In the application area of the proposed approach data is processed automatically, using a reliable connection. Consequently, data quality metrics concerning the representation or the accessibility of data are not relevant, since they do not describe the data itself. The *intrinsic* and *contextual* categories, however, are important in the addressed context. The *subject criteria* and *process criteria* classes according to Naumann and Rolker [14] are not relevant to the proposed concept, because they seal with how the user perceives the information or how the query processing is treated. In [16], a distinction was made between quality metrics related to a particular user's view and data-related quality metrics. Since the user's view is not important in the presented concept, only metrics that have an impact on the data itself are applied. The remaining data quality metrics that are relevant in the particular application scenario are *Completeness*, *Consistency*, and *Correctness*.

Completeness has been addressed in various research papers, with - in some cases - different interpretations of the definition depending on application area and point of view. Table I lists various contributions with different definitions of completeness. While some concentrate on the presence or absence of entries, others - such as Kahn et al. [19] and Ballou et al. [15] - take a closer look by evaluating whether the amount of information represented by the content is sufficient. Generally, a system is complete if it includes the whole truth. The completeness quality metric is often related to NULL values in databases and information systems. The general understanding is that a NULL value must be treated like a

missing value, but it may also be that it is not known whether it exists or that it does not exist at all, which describes a considerably different perspective on completeness [9]. This means that the conceptual organization of an information system can be seen from two different points of view called *closed world assumption* (CWA) and *open world assumption* (OWA). Under the CWA, all information captured by the information system represents facts of the real world and anything that is not described is assumed to be false. Under the OWA, it cannot be stated whether a fact not stored in the information system is false or whether it does not exist at all. In an OWA-based system that does not store NULL values, identifying the completeness of an information system requires the introduction of a new concept called reference relation. This concept stores all real-world facts with respect to the structure of the particular relation. In comparison to a relation of an information system storing all facts of the real world except one object, the reference relation would contain all information of the relation plus the missing object not captured by the relation. The metric completeness can be defined formally as follows. For a database scheme D, we assume a hypothetical database instance d_0 that perfectly represents all information of the real world that is modeled by D. Furthermore, we assume that one or more instances d_i $(i \geq 1)$ exist, each of them is an approximation of d_0. Next, we consider some views, where v_0 is an ideal extension of d_0 and v_i $(i \geq 1)$ are extensions of the instances d_i. Equation (1) represents this concept, where the absolute values represent the number of tuples [20][21].

$$\frac{|v_i \cap v_0|}{|v_0|} \tag{1}$$

Under the CWA, completeness is defined differently, because NULL values indicate entries that do not exist in the real world. Completeness can therefore be seen from the granularity perspective [10]. The following four types of completeness can be distinguished according to their granularity:

- *Value completeness*: When this type is applied, completeness is determined at the finest-grained level, and the ratio between existing values of particular fields and the total number of fields (including NULL values) is calculated.
- *Tuple completeness*: On a more general level tuple completeness represents the completeness of a particular tuple represented by the tuple's ID. For example, if a relation has four attributes and a particular tuple contains one NULL value, the completeness for this tuple would be 75%.
- *Attribute completeness*: Similar to tuple completeness, this describes the completeness value of a particular attribute. It is calculated as the ratio of existing values and the total number of tuples (containing NULL values).
- *Relation completeness*: This type of completeness is based on the number of NULL values and the total number of values in a whole relation.

It is important to analyze a particular application in detail to determine how NULL values are treated correctly, because they can have different meanings. For example, when the relational model is used there is often a primary key defined for a relation. Since members of the primary key cannot be NULL, missing objects cannot be expressed using NULL entries for these attributes. If a particular attribute is not member of the primary key, it can be NULL (assuming there are no NOT NULL constraints), and therefore it is possible to represent missing objects as NULL values. In the application area of the proposed concept, the scenario is similar, as unknown or non-existent features are represented as NULL values if the particular attribute is not in the set of primary key attributes. If an object exists in the real world but is not represented in the dataset, then no tuple is stored in the database, since primary key attributes cannot be set to NULL.

TABLE I. Completeness definitions in scientific papers.

Reference	Definition
extent to which the value is present for that specific data element	[7]
breadth, depth, and scope of information contained in the data	[11]
presence of all defined content on both data element and dataset levels	[15]
schema completeness is the degree to which entities and attributes are not missing from the schema; column completeness is a function of missing values in a column of a table	[17]
every fact of the real world is represented; it is possible to consider two different aspects of completeness; (i) certain values may not be present at the time, and (ii) certain attributes cannot be stored	[18]
extent to which information is not missing and is of sufficient breadth and depth for the task at hand	[19]
related to the Closed World Assumption (CWA); the information stores the whole truth	[22]
ability of an information system to represent every meaningful state of a real-world system	[23]
degree to which data values are included in a data collection	[24] (via [9])
percentage of real-world information entered in data sources and/or data warehouses	[25] (via [9])
information having all required parts of an entity's description	[26]
ratio between the number of non-NULL values in a source and the size of the universal relation	[27] (via [9])
all values that are supposed to be collected as per a collection theory	[28]

Consistency is a data quality metric whose definition is very similar across different research papers: multiple entries with the same meaning should be represented identically or in a similar way. Interestingly, consistency is often closely related to integrity and integrity constraints. Batini et al. [9] defined consistency as the ratio of values that do not violate specific rules and the overall information set. They stated that these rules can be either integrity constraints (referring to relational theory) or consistency checks in the field of statistics. Integrity constraints can be further subdivided into inter-relational constraints and intra-relational constraints, depending on whether the constraint relates to one or more tables. Pipino et al. [17] also defined consistency as closely connected to integrity constraints (e.g., Codd's Referential Integrity constraint). They proposed that consistency can be calculated as a ratio using the number of violations of a specific consistency check and the total number of consistency checks. Bovee et al. [26] defined consistency as a sub-metric of integrity dealing with different representations of the same information in multiple entries. A summary of the different definitions is listed in Table II.

In the context of the presented approach, consistency is considered as the entries' violation of - or, more specifically, their compliance with - rules that represent consistency checks.

TABLE II. Consistency definitions in scientific papers.

Reference	Definition
refer to the violation of semantic rules over a set of items	[9]
format and definitional uniformity within and across all comparable datasets	[15]
consistency of the same (redundant) data values across tables (e.g., Codd's referential integrity constraint); ratio of violations of a specific consistency type to the total number of consistency checks subtracted from 1	[17]
there is no contradiction in the data stored	[18]
requires that multiple recordings of the value(s) for an entry's attribute(s) be the same or closely similar across time or space	[26]
different data in a database are logically compatible	[28]

TABLE III. Correctness definitions in scientific papers.

Reference	Definition
[accuracy] data are certified error-free, accurate, correct, flawless, reliable, errors can be easily identified, the integrity of the data precisely	[11]
[free-of-error] number of data units in error divided by the total number of data units subtracted from 1	[17]
every set of data stored represents a real-world situation	[18]
[free-of-error] extent to which information is correct and reliable	[19]
[validity] the data sources store nothing but the truth	[22]
[accuracy] refers to information being true or error free with respect to some known, designated, or measured values	[26]
[accuracy] extent to which collected data are free of measurement errors	[28]
[accuracy] data are accurate when the data values stored in the database correspond to real-world values	[29]

It is calculated as the ratio of entries satisfying all consistency checks and the total number of entries. An example of such a consistency check is the proof of duplicates in the dataset.

Correctness is a metric that indicates whether the stored information is valid. A summary of different definitions from scientific papers is listed in Table III. Since different terms are often used for the same concept, the original attributes are given in brackets. Pipino et al. [17] provided a very technical definition that explains how the metric is calculated. In [18], the definition was very general, defining correctness of a particular dataset as the presence of a corresponding real-world subject. In this contribution, correctness is also seen as a valid representation of real-world entities. Semantic rules are required to determine whether a particular entry is correct or in the correct range. Since functional requirements can change over time, it is important to modify these rules if necessary [23].

It is very difficult to verify the correctness of data, since tacit information from domain experts is required in most cases. Hence, expert knowledge must be represented as a set of semantic rules, which are applied to the data in information systems to determine whether the content satisfies these conditions. Note that correctness heavily depends on the application area, which means that, even if a particular entry in a dataset complies with all rules of one application area, it might still fail checks of another.

C. Analysis of Univariate Time Series

The proposed approach uses time series analysis to estimate a model that fits the observed data and computes forecasts to determine future values. For this purpose, models must be compared in order to find the most suitable one. Since the application area is based on a single observed variable, we focused on methods that address univariate time series.

Time series analysis is a very popular research field and dates back to 1906, where Schuster recorded sunspot numbers in a monthly schedule, which was one of the first recorded time series. Nowadays, a wide variety of applications exists, ranging from stock analysis and calculations concerning demography to sunspot observations. The basic purpose of a time series is to capture a set of sequential observations over a time period. Methods are needed to compute a model for generating a time series with minimal differences between the observations and the model-generated data points [30]. Time series analysis has two major goals: (i) to express the underlying process that leads to the observations as accurately as possible, and (ii) to obtain a model that predicts future values based on the course of the time series. The smaller the difference between the generated course and the data points the better the model supposedly describes the underlying process. However, this statement is not entirely correct, since a model can also be fitted too closely to the curve (called overfitting), which means that it expresses the observations in too much detail, and also models outliers that might not have a systematic impact. Overfitting results in poorer out-of-sample prediction performance (calculating forecasts) than a model that is fitted less exactly. Time series analysis is closely connected to forecasts, since it focuses on the prediction of future values for a known time series. Weather forecasts are a popular example, where former observations are known and future values are predicted on their basis (considering the laws of physics) [30]. A very basic classification of time series distinguishes between univariate and multivariate types, depending on whether they focus on one or multiple target variables, respectively. Thus, different courses (variables) are analyzed for the same time period, which means that different features are observed at a single point in time (represented as vectors) [31]. The proposed approach focuses on univariate time series and the following time series models were compared to find the most suitable one for the application area.

Box-Cox Transformation, ARMA errors, Trend, and Seasonality (BATS) is a method introduced by De Livera et al. [32]. Since it uses Box-Cox transformations, it does not focus exclusively on linear homoscedastic time series, but also supports nonlinear ones. Furthermore, the method also considers ARMA errors, where ARMA parameters are evaluated and determined in a two-step procedure, as this leads to the best results [33]. Additionally, the trend component is computed using an adaption to the damped trend. The method incorporates mechanisms to deal with seasonal influences, as these often occur in time series. In [32], *Trigonometric BATS* (TBATS) was proposed as an extension to the BATS model, which replaces the seasonal definition of BATS with a trigonometric formulation. A method that was used very often in the past is *Simple Exponential Smoothing* (SES), which applies weights to the individual observations of the time series [34]. As the name indicates, these weights are not equally distributed but decrease exponentially over time giving more

recent observations a higher impact than previous ones. An extension to SES was introduced by Hyndman et al. [35]. They proposed a framework called *Exponential Smoothing State Space Model* that makes it possible automatically determining the best exponential smoothing algorithm and its parameters using state space models. Since their approach delivered good results on the M3-data, it was also investigated and tested in the context of the proposed concept. The quality criteria that is used by this framework is *Akaike's Information Criterion (AIC)* [35]. Another method that was introduced in the application area of demand forecasting is *Croston's Method* (CROSTON) [36][37], which uses multiple single simple exponential smoothing forecasts and treats zero observations separately (in the application area of demand modeling, these are the observations where the demand is zero). *Auto-Regressive Integrated Moving Average* (ARIMA), which belongs to the family of *Auto-Regressive Moving Average* ARMA models, is a popular method for fitting time series and forecasting. ARMA models consist of two components: the auto-regressive component (AR) and the moving average component (MA). The AR-component computes the dependencies between previous values/observations and their impact on the current observation, while the MA-component estimates the smoothing function for the observations in a particular time period. Various modifications to the ARMA model have been proposed, among them ARIMA, which considers also non-stationary processes [38]. Neural networks are used more often for time series analysis. A popular representative is the *feed-forward network with a single hidden layer* (NNETAR). Artificial neural networks are based on inputs and dependent variables; the parameters are transformed, weighted, and combined using one or more hidden or intermediate layers in order to determine the output variable. In [39], the authors presented a comparison of neural networks in different usage scenarios, and - based on recent research - concluded that the risk of over-parameterization is a well-known problem. Hence, they recommended using feed-forward neural networks with a single hidden layer [39].

D. Determination of the Forecasting Accuracy

In the presented concept, assessment of the quality and thus the reliability of a prediction is a key task. In order to determine the reliability of a predicted value, it is important to know how good the particular prediction is. Hence, predictions should be evaluated using new observations as soon as they become available. As this topic is often tightly coupled with time series analysis, many research papers have addressed it. Below, we provide an overview of error terms including their benefits and drawbacks, since these are the terms in which accuracy measures are often considered.

Hyndman and Koehler [40] distinguished between four different types of error measures: (i) scale-dependent measures, (ii) measures based on percentage errors, (iii) measures based on relative errors, and (iv) relative measures (Table IV). In addition to these categories they proposed a scale-independent metric called *Mean Absolute Scaled Error (MASE)*.

The first category of scale-dependent measures includes *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, *Mean Absolute Error (MAE)*, and *Median Absolute Error (MdAE)*. The problem with these metrics is that they cannot be compared easily across various time series of different scale. A wide range of applications use these metrics to determine the forecast accuracy of univariate time series [41]. Armstrong and Collopy [42] also addressed the problem arising from scale dependency. The second category is about measures based on percentage errors. Commonly used metrics are *Mean Absolute Percentage Error (MAPE)*, *Median Absolute Percentage Error (MdAPE)*, *Root Mean Square Percentage Error (RMSPE)*, and *Root Median Square Percentage Error (RMdSPE)*. An advantage of these methods is that they are scale-independent and therefore suited to comparing the forecasts of different time series. However, there are also some disadvantages: Firstly, it is not always guaranteed that they are finite or defined. MAPE, for example, encounters problems when a time series is close or equal to zero [39]. Additionally, MAPE and MdAPE come with the drawback that they treat positive errors worse than negative ones, which results in asymmetry. Makridakis [43] described extensions to these metrics in order to find symmetric error metrics, which are called *Symmetric Mean Absolute Percentage Error (sMAPE)* and *Symmetric Median Absolute Percentage Error (sMdAPE)* as an attempt to overcome the asymmetry problem. However, sMAPE and sMdAPE are less symmetrical as their names might imply: It has been shown that the resulting error is greater for overpredictions than for underpredictions by the same amount [39][44]. The third category of forecast accuracy metrics covers measures based on relative errors. Popular metrics of this category are *Mean Relative Absolute Error (MRAE)*, *Median Relative Absolute Error (MdRAE)*, and *Geometric Mean Relative Absolute Error (GMRAE)* [39][40][42]. The advantage of these methods is that the metrics not only compare the times series with the corresponding forecasts, but also compare it with predictions from a different forecasting method that serves as a benchmark method. In many cases, *random walk* is used for this purpose. The fourth category also defines measures on the basis of a comparison between the method applied and a benchmark method. The *Relative Mean Absolute Error (RMAE)* is defined as the ratio between the MAE of the applied method and the MAE of the benchmark method. Similar metrics can be calculated comparing error metrics of the applied model with those of the benchmark method (e.g., *Relative Mean Squared Error (RMSE)*). The im-

TABLE IV. Overview of forecast accuracy metrics.

Category	Metric	Definition
scale-dependent measures	MSE	Mean Squared Error
scale-dependent measures	RMSE	Root Mean Squared Error
scale-dependent measures	MAE	Mean Absolute Error
scale-dependent measures	MdAE	Median Absolute Error
percentage errors	MAPE	Mean Absolute Percentage Error
percentage errors	MdAPE	Median Absolute Percentage Error
percentage errors	sMAPE	Symmetric Mean Percentage Error
percentage errors	sMdAPE	Symmetric Median Percentage Error
percentage errors	RMSPE	Root Mean Square Percentage Error
percentage errors	RMdSPE	Root Median Square Percentage Error
relative errors	MRAE	Mean Relative Absolute Error
relative errors	MdRAE	Median Relative Absolute Error
relative errors	GMRAE	Geometric Mean Relative Absolute Error
relative measures	RMAE	Relative Mean Absolute Error
scale-independent measures	MASE	Mean Absolute Scaled Error

provement provided by the applied method is always expressed in relation to a benchmark method. The drawback of these measures is that they do not indicate an *absolute goodness* of the forecast itself.

IV. Improving Early Detection of Significant Faults in Quality Management

This section covers the Q-AURA analysis process, the invented improvements of it, and their integration into Q-AURA. Q-AURA is a system that supports quality management experts in analyzing faults occurring in the after-sales market. Defect and warranty information is gathered from car dealers who inspect customers' cars and detect faults. The business process relevant for Q-AURA, which ranges from the development of an engine to the after-sales market, is illustrated in Figure 1.

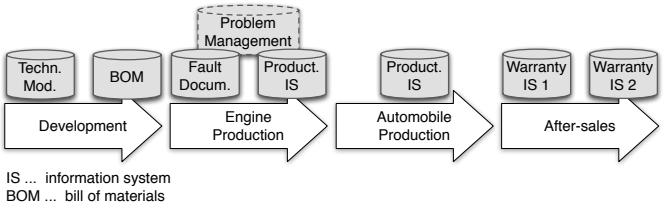


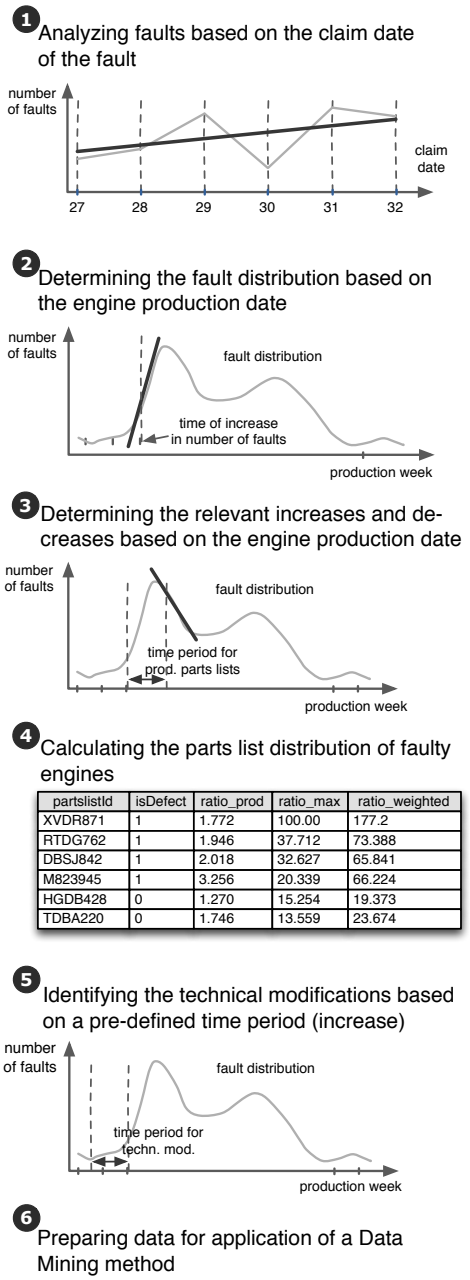
Figure 1. Flow chart of the business process relevant to Q-AURA.

Fault and warranty information is distributed across information systems, which contain partially redundant information. Since partially different data is also stored in the information systems, integration would result in a more complete, holistic view of real-world situations. In combination with Q-AURA's primary aim of identifying significant faults, this extension targets more accurate and robust results if the information is processed and interpreted correctly. Q-AURA's secondary aim of analyzing significant faults further in order to determine technical modifications that might underlie them requires additional information residing in information systems from other process steps. Therefore, these data sources must also be integrated to cover the whole engine lifecycle.

A. Q-AURA Approach

This section describes the Q-AURA approach and its analysis process, which forms the basis of the invented improvement [45]. The underlying analysis process is divided into different steps, which modify the information such that (i) data mining methods can be applied and (ii) the most appropriate representation of the data can be found. These six process steps are illustrated in Figure 2.

The first step is the identification of significant faults that occur in the after-sales market, which are then further analyzed (cf. Figure 2-1). The term *significant* is used for faults with negative consequences that have developed in recent weeks. The information base that is used for this step covers cars that were manufactured in the last three years. To detect faults that have occurred recently and indicate current problems, the last six weeks are considered. These boundaries were set carefully in order to take those cars into account that influence the ongoing development process. Since various engine types exist and since fault types have a different distribution depending on the car brand (e.g., BMW and MINI), the appropriate level of granularity for the analysis had to be found: finally, the



partslistId	isDefect	ratio_prod	ratio_max	ratio_weighted
XVDR871	1	1.772	100.00	177.2
RTDG762	1	1.946	37.712	73.388
DBSJ842	1	2.018	32.627	65.841
M823945	1	3.256	20.339	66.224
HGDB428	0	1.270	15.254	19.373
TDBA220	0	1.746	13.559	23.674

Figure 2. Q-AURA process in detail.

result was to classify faults according to fuel type, car brand, and engine type. Thus, faults that occur in BMW automobiles are not in the same analysis set as faults in engines that have the same engine type and fuel type, but are built into MINI automobiles. Regression analysis is used to determine significant faults [46]. Three different approaches to regression analysis based on convex functions, smoothing functions, and a straight line were tested to find the best method. The evaluation was done using fault courses from most of the analysis sets for diesel engines over several weeks. Experts from the diesel quality management department, who helped in finding the method that best identifies significant fault courses were contacted weekly. The evaluation revealed that the straight-line approach outperformed the others. Different

metrics of the regression line can be calculated to determine its characteristics. Q-AURA previously used *gradient*, *mean value*, and *coefficient of determination*. The *coefficient of determination* (indicating how well the regression line fits the actual course) and the mean value were replaced with a new metric called *gradient_ppm* (Equation (2)). This value is calculated as the ratio between the gradient and the number of faults (regardless of the fault type) of the engine type ($n_{enginetype}$).

$$k_{ppm} = \frac{k * 1.000.000}{n_{enginetype}} \quad (2)$$

Those faults that exceed specific thresholds are analyzed in more detail. These thresholds were investigated and evaluated carefully together with quality management experts. Faults that are not classified as significant are not analyzed further.

For each significant fault, the production week histogram is calculated in the second step (cf. Figure 2-2). The histogram is based on cars that were produced in the preceding three years with claims from the last two years. It shows the number of produced engines with the particular fault in relation to the total number of produced engines of the same class (according to fuel type, car brand, and engine type). This is done to take production fluctuations into account, because an increase in the number of engines produced will most likely affect the number of faults, but does not necessarily indicate a systematic failure during engine development. The course is then normalized by the highest value in order to identify more clearly the highest fault peaks in time. A 5-point smoothing function is applied to eliminate outliers. The resulting course forms the basis for identifying the critical time periods, which are bound by an initial significant increase and a decrease. An increase of the course, which is defined as the ratio between faulty engines and the total number of engines produced, indicates that one or more negative effects have occurred that influence product quality (e.g., a new technical modification that changed the engines). The identification of significant increases is illustrated in Figure 2-2.

Afterwards (cf. Figure 2-3), the decreases of the course are determined. Both steps (finding increases and decreases) are performed using sliding windows and calculating the slope. Subsequently, interesting time periods can be identified, each of which is bound by a significant increase and the subsequent decrease. Such a time period represents the time when most of the engines affected by the particular fault were produced.

In the next step (cf. Figure 2-4), the faulty engines identified for a time period are investigated in more detail. In order to determine more exactly which subset of them is affected most by a particular fault, the distribution of the engine material number is analyzed. The engine material number represents a particular bill of materials (BOM) and, therefore, defines an engine in much detail. The bill of materials specifies all components and parts that are necessary for assembling an engine. A BOM entry contains information such as part number, number required, and unit. More interesting for Q-AURA is that a BOM also stores the technical modification identifier. A technical modification describes the reason why a particular part is in the BOM and which former part it substitutes (if it is a substitution). Possible reasons could be a new supplier or that the former part lead to a quality issue. The BOM distribution is put in relation to the engines produced with the same engine material number to select those material numbers that have a bad ratio. The ratio is then normalized to identify the BOMs that must be analyzed further, since they are affected most by the analyzed fault.

Step 5 in Figure 2 illustrates how the technical modifications are selected. Not every technical modification that occurred throughout the whole time period analyzed is relevant, since a technical modification that was implemented months after the significant increase, cannot be the cause of the fault. Therefore, the time period from which technical modifications are selected can be limited, which is important because the number of technical modifications made over time is vast, which prevents application of intelligent methods and makes drawing meaningful conclusions difficult. In order to avoid being too strict and selecting insufficient modifications (and possibly missing the causative modification), a three-month period starting two months before a significant increase is used. This period was defined and evaluated together with quality management experts.

In the last step, the number of technical modifications is limited to those most likely to have provoked the fault (cf. Figure 2-6). Using the modifications determined in step 5 and the engine classification according to their engine material numbers, two alternatives were implemented that determine the relevant set of technical modifications. The first is a descriptive approach that identifies modifications that are covered by most of the significant engine material numbers, while the second uses association rules. More detailed information about these two methods can be found in [45].

This analysis concept, which forms the core of Q-AURA, is already in daily use by quality management experts at different engine production plants. The evaluation of the tool showed that it provides a significant benefit. The problem solving time for engines produced in the plant in Steyr was recorded in two consecutive years (before Q-AURA was applied and after its introduction). It showed that the reduction was approximately 2% [45].

B. Optimized Early Detection

This section describes the new improved concept in detail and shows the advantages over the current Q-AURA implementation. Clearly, early detection of faults that occur during development or production is crucial, since in most cases they result in negative effects for the company. As described in Section IV-A, Q-AURA is an application that identifies current problems (represented as engine faults) and automatically analyzes them in detail to gather more information about possible causes. This means that early detection is also an important task for Q-AURA. Since finding the causes of a particular fault is very time-consuming, improvements by a single day or even a week are highly beneficial. Thus, an approach was invented, which optimizes (i.e., accelerates) Q-AURA's fault detection method. The improved concept consists of four components, each fulfilling a different task: (i) assess information systems based on data quality, (ii) analyze univariate fault time series and compute forecasts, (iii) determine whether a particular fault is significant using predictions based on multiple information systems, and, (iv) evaluate the prediction accuracy to determine the quality of the forecasts.

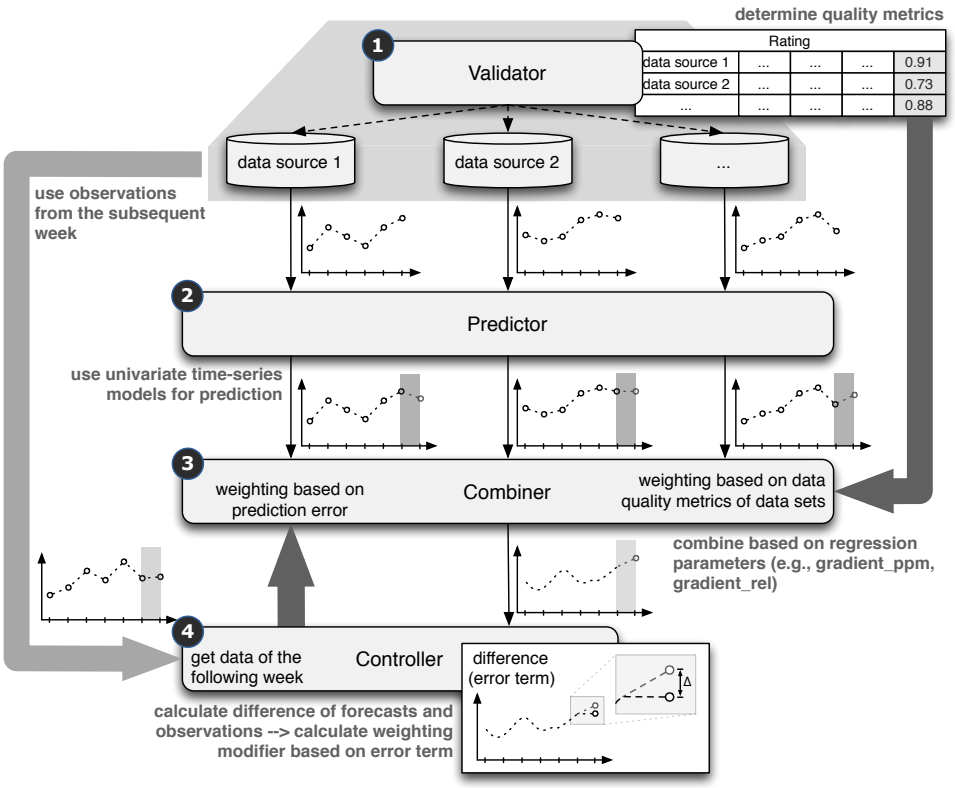


Figure 3. Concept for optimizing early detection.

The overall concept is illustrated in Figure 3. The *Validator* (cf. Figure 3-1) is responsible for determining a specific information system's data quality. Different data quality metrics are used (completeness, correctness, and consistency) to calculate the component's result, which constitutes an overall data quality metric for the particular information system. The *Predictor* (cf. Figure 3-2) analyzes the fault time series for each fault in each data source. This means that a model must be generated that describes the process underlying the time series as well as possible in order to be able to calculate a forecast (out-of-sample prediction). A single value is forecasted, which is then used to determine the significance of the particular fault. Regression analysis is applied considering a six-week period (containing the forecast value as the most recent one). In the subsequent step, the *Combiner* integrates weighting factors and the regression parameters of each data source's regression line to calculate an overall significance metric that indicates whether a particular fault is significant. Finally, the *Controller* determines the accuracy of each forecast. This is achieved by comparing new entries in the information systems from the following week. The prediction error is calculated for each data source using the new value and the predicted value of the previous week. This prediction error is then used to compute a weighting factor that is required by the combiner component.

1) Validator: The validator is responsible for determining the data quality of a particular information system. Various quality metrics from the scientific literature were compared to identify quality metrics that are relevant to the proposed approach. As described in Section III-B, the completeness, correctness, and consistency quality metrics are applied to compute the overall data quality metric.

Completeness is a data quality metric that has different interpretations in research because it can be seen from different perspectives. In the proposed concept multiple datasets exist that store partially redundant warranty and fault information. In industry, data that is used for intensive analytical processing is usually stored in an aggregated form in *data warehouses* (DWHs). Data warehouses are often designed to store historical information, while operational information systems capture only a short time period (to increase performance and throughput) [47]. In many cases, data marts are developed, which do not satisfy the third normal form of relational algebra, since they are organized to improve the performance of analytical queries and transformations. Figure 4 illustrates the DWH concept. Each intermediate step between the original information source and the data warehouse is a source of potential errors that may occur while transforming and cleansing data.

At the bottom-most level, various operational systems store the data as it is being generated. The data models support a particular business case, ensure that relevant information about real-world objects is inserted correctly, and verify the completeness at a particular level (primary key constraints, foreign key constraints, and not-NULL constraints are basic options to ensure this). At the next level, data warehouses are set up to provide an analytical basis for different business aspects. ETL processes extract, transform, and load data in preparation for DWH use cases. During the ETL process, some information may be filtered or left out due to unrequested transformation errors. Thus, completeness of the target DWH is reduced. Since different DWHs that store redundant information exist

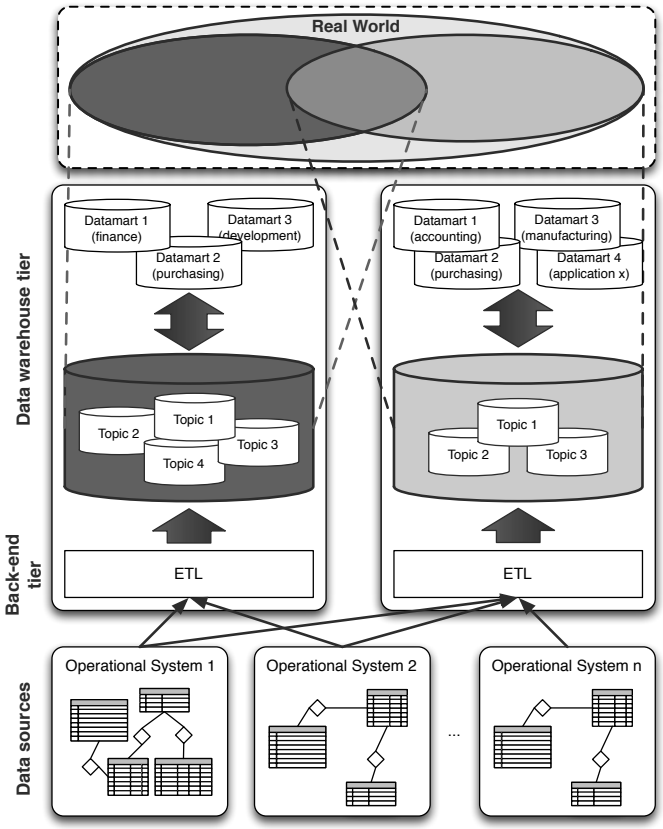


Figure 4. Completeness in data warehouses.

in the addressed application scenario, each of them may have a different view of the real world. As illustrated in Figure 4, it may be necessary to combine these views to obtain the best possible representation of the real world. This concept assumes that data in the information systems does not represent false information, since this would lead to a false representation of the real world. In the addressed application area, the processes are well supported, and in the past the most likely problem was data missed rather than false data. The resulting information base can be seen as a *reference dataset* (similar to the reference relation concept explained in Section III-B). The reference dataset is defined as shown in Equation (3).

$$d_r = \bigcup_{i=1}^{n} d_i \tag{3}$$

d_i are the instances stored in a particular data source (DWH) and d_r represents the total number of records in the reference dataset. In this case, DWHs are considered under the OWA, since it is not exactly known whether information is missing. If different DWHs store data from the same application area, a combination of these entries would lead to a better overall view (reference dataset). In order to calculate the completeness data quality metric for a single information system, the amount of information must be checked against the reference dataset. Equation (4) illustrates how the completeness metric (Q_{comp,d_i}) for a particular data source d_i can be obtained.

$$Q_{comp,d_i} = \frac{|d_i \cap d_r|}{|d_r|} \tag{4}$$

The second data quality metric used in the proposed concept is consistency, which is closely connected with integrity constraints. A perfectly designed data model would apply integrity constraints such as unique, primary key, and referential integrity to prevent inclusion of false data. Some information systems do not implement constraints and, therefore, inconsistencies may occur. An important consistency constraint is referential integrity, which guarantees the existence of a value in the corresponding database table. The consistency metric is calculated as illustrated in Equation (5).

$$Q_{cons,d_i} = \frac{|d_{i[conspos]}|}{|d_{i[all]}|} \tag{5}$$

In the mentioned equation the numerator $|d_{i[conspos]}|$ is defined as the absolute value of the entries that passed the consistency checks, and the denominator is the total number of entries of the dataset. Like the other quality metrics this calculation is applied to each information source.

The third and final data quality metric used to evaluate the data sources is correctness, which is based on semantic checks in the proposed approach. Semantic checks depend on the application scenario and the context in which the data is used. For example, if an attribute is defined as a value between 1 and 5 (e.g., indicating a grade given in Austrian schools) and a field contains the value 6, it is obvious that this information is false. Further, consider an attribute that has a strictly defined structure: the value has six signs, the first one being a letter between A and D, the next three signs between 1 and 6, and the last two signs alphanumeric. As another example, consider an application dealing with dates, where a particular attribute contains only past dates; if an entry contained a future date, it would have to be false. A check of two date attributes would have to verify that they are in sequence, meaning that one must precede the other. These examples show that considerable contextual knowledge is necessary to determine whether a particular entry is correct. More formally, the following check types can be identified:

- Range check: proves whether a particular value is in the correct range, e.g., only past dates are allowed or an integer range between 1 and 5.
- Structure check: evaluates whether entries of a particular attribute satisfy a given format, e.g., total value length is six or it must be a numeric value.

These checks need not necessarily to be static for all instances. It is very important that attributes can also depend on each other. An example is information about pupils, their residence, and their grades. If the residence of a pupil is in Austria, then the grades must be in the range between 1 and 5 (from the set of natural numbers). However, for residents of Switzerland, the range is 1 to 6 (with steps of 0.5).

Note that contextual or semantic changes (in the business process) imply that the checks for correctness must also be adapted to avoid a false correctness metric that would decrease the overall data quality metric of the data source and lead to false results of the proposed concept. Equation (6) shows the

calculation of the correctness quality metric (Q_{corr,d_i}) for a particular data source d_i.

$$Q_{corr,d_i} = \frac{|d_{i[corrpos]}|}{|d_{i[all]}|} \quad (6)$$

In the presented equation, the numerator $|d_{i[corrpos]}|$ represents the absolute value of the entries that proved correct, and $|d_{i[all]}|$ is the total number of entries in the data source.

Finally, the overall data quality metric of the data source can be calculated as the multiplication of the three quality metrics discussed (Equation (7)). The resulting quality metric of data source d_i is represented by Q_{d_i}, and the completeness, consistency, and correctness component quality metrics are denoted by Q_{comp,d_i}, Q_{cons,d_i}, and Q_{corr,d_i}, respectively. This metric is then used as a weighting factor W_{qual,d_i} for the combiner component.

$$W_{qual,d_i} = Q_{d_i} = Q_{comp,d_i} * Q_{cons,d_i} * Q_{corr,d_i} \quad (7)$$

An example output of the validator component is shown in Figure 5.

data source	Q_{comp}	Q_{cons}	Q_{corr}	Q_d
data source 1	0.85	0.91	1.00	0.77
data source 2	0.94	0.97	0.98	0.89
data source 3	0.98	0.89	0.92	0.80
...	...	...	...	...

Figure 5. Example results of the validator component.

2) Predictor: The predictor's tasks of forecasting future values based on a particular dataset's values and of performing regression analysis are implemented as two steps: (i) determining the value of the following week for the various fault types based on their number from the previous weeks and (ii) regression analysis using the previous five weeks and the calculated forecast.

In order to generate future values, the contextual requirements must be known to investigate and determine which time series method best suits the use case. In the proposed concept, the prediction of how many faults will occur in the following week is performed based on the number of faults in the after-sales market from an appropriate time period. The specific fault analysis set is bound by the particular fault type, fuel type, car brand, engine type, and the period to be used in the prediction task. In this scenario, the period was set to one year, a relevant period in the investigation process of the quality management experts. Different time series analysis methods which are capable of performing the prediction task are listed in Section III-C. An evaluation identified the most appropriate approach, which depends on the underlying process and the given time series. To this end, two types of quality checks were applied: one is based on the *Diebold-Mariano Test* [48], which compares the prediction quality of two methods, and the other calculates *Goodness-of-Fit* measures (e.g., MAPE, sMAPE, MAE). The test scenario was established as follows:

- Different time series defined as a sets of fault type, fuel type, car brand, and engine type were evaluated.
- Every possible pairwise combination of time series methods was used in the Diebold-Mariano test to obtain a matrix that shows how they perform in relation to each other. The *h* value was set to one, which specifies that only a one-point forecast was evaluated, since this is also the aim in the application scenario. The alternative hypothesis method was set to *greater*, which means testing whether method two is more accurate than method one. The loss function power was set to two, a commonly chosen value.
- To determine how good the different predictions perform using the *Goodness-of-Fit* metrics, in-sample predictions were computed, where the most recent week (observation) was left out for the comparison task. The different quality metrics were then calculated using the left-out observation and the forecast value.
- The results were ranked to see which prediction method outperforms the others in the particular use case.

The results revealed that it cannot be clearly determined which prediction method is the best, since this heavily depends on the course of the time series. The *Goodness-of-Fit* metrics could not establish a clear winner: the best methods were ARIMA, TBATS, and Croston's method. The Diebold-Mariano tests identified ARIMA and TBATS as superior methods; hence, the two are favored by the proposed concept. ARIMA is used for the prediction task, since it is also provided by a tool already in use by the business partner.

The second task of the predictor component is to perform regression analysis of data from a six-weeks period. As in the current implementation of Q-AURA, linear regression using a straight line was chosen, since this yields the best results and has been applied and evaluated for two years. The period used for regression analysis includes the most recent five weeks observed and the value predicted for the next week. The characteristic values *gradient*, *mean value*, and *coefficient of determination* are computed. The gradient and the mean value are calculated using the equation for a straight line (Equation (8)).

$$y = k * x + d \quad (8)$$

The parameters x and y represent a two-dimensional coordinate in the diagram, where x corresponds with being the time value and y is the observed (or predicted) value of the focused measure. The characteristic value k is the gradient and represents the average increase between two subsequent points in the diagram. d is the offset and describes the initial or start value y at $x = 0$. Another characteristic value of the regression line is the mean value $\bar{y}$, which is computed by averaging the data points over the time period. In the use case of the proposed concept this period consists of five observations and the predicted value. A previous version of this approach also calculated the *coefficient of determination* [46]. This value describes the steadiness of the regression line. In the proposed concept the regression line depends only on one variable, therefore the coefficient is equal to the square of *Pearson's Correlation Coefficient* r_{xy}^2 (Equation (9)) [49].

$$R^2 = r_{xy}^2 = \frac{s_{xy}^2}{s_x^2 s_y^2} \quad (9)$$

Based on the gradient, two new values are calculated, which provide a more detailed view of the course over six weeks. The first is an extension of the gradient, since it determines the relative value based on the mean value of the six weeks (of the analysis set). The mean value is interpolated based on the observations from the previous five weeks, because the most recent week is predicted, and thus has no underlying number of observed faults (Equation (10)).

$$n_{6weeks} = n_{5weeks} * \frac{6}{5} \tag{10}$$

This value is then used to determine the relative gradient of the six weeks (Equation (11)).

$$k_{rel} = \frac{k}{n_{6weeks}} \tag{11}$$

The second value is called *gradient parts-per-million* (k_{ppm}), which is also based on the gradient (k) of the regression line. The idea behind this metric is the identification of faults with a high value when compared to the particular engine type. Since the regression line is based on the analysis set consisting of fault type, fuel type, car brand, and engine type, it limits data to a fine-grained but appropriate set of faults. While k_{rel} determines the average gradient based on this analysis set, k_{ppm} takes the whole number of faults for the particular engine type $n_{enginetype}$ into account as given in Equation (12). Since the result is a very low value, it is expressed as ppm (multiplied by 1,000,000).

$$k_{ppm} = \frac{k * 1,000,000}{n_{enginetype} * \frac{6}{5}} \tag{12}$$

These computations are performed for each analysis set (fault type, fuel type, car brand, and engine type) from each dataset. An example of an output from the resulting data structure is shown in Figure 6.

data source	k	$\bar{y}$	R^2	k_{rel}	k_{ppm}
data source 1	1.92	12.34	0.32	0.69	123.23
data source 2	2.45	32.91	0.19	0.23	453.21
data source 3	0.64	24.54	0.98	0.76	91.23
...	...	...	...	...	...

Figure 6. Example results of the predictor component.

3) Combiner: The third component of the proposed concept is the combiner, which decides whether the analyzed fault is significant. As explained above, the predictor uses regression analysis and calculates the corresponding characteristic metrics, k_{rel} and k_{ppm}. The combiner uses these two parameters in addition to weighting factors from the validator and the controller component. The overall weighting factor for a particular fault is computed as the product of the data quality metric ($W_{qual,i}$) and the weighting factor based on the prediction accuracy ($W_{cont,i,ft}$) (Equation (13)).

$$W_{i,ft} = W_{qual,i} * W_{cont,i,ft} \tag{13}$$

In the proposed approach, two concepts have been developed with different granularities to determine the overall result that decides whether the fault is significant.

- *Parameter-driven approach*: In the first step the differences between defined thresholds and the characteristic parameters (k_{rel} and k_{ppm}) are calculated, which are then multiplied by the corresponding weighting factors and divided by the sum of the weighting factors over the different information sources (Equation (14) and Equation (15)). A fault is significant if both resulting values are greater than 0, and insignificant otherwise (Equation (16)).

$$R_{rel,ft} = \frac{\sum_{j=1}^{n} W_{i,ft} * (k_{rel,thr} - k_{rel,i,ft})}{\sum_{i=1}^{n} W_{i,ft}} \tag{14}$$

$$R_{ppm,ft} = \frac{\sum_{j=1}^{n} W_{i,ft} * (k_{ppm,thr} - k_{ppm,i,ft})}{\sum_{i=1}^{n} W_{i,ft}} \tag{15}$$

$$S_{ft} = \begin{cases} R_{rel,ft} > 0 \cap R_{ppm,ft} > 0, & 1 \\ R_{rel,ft} \leq 0 \cup R_{ppm,ft} \leq 0, & 0 \end{cases} \tag{16}$$

Figure 7 shows an example database table resulting from the parameter-driven approach. The result is defined on the analysis set that consists of fault type, fuel type, car brand, and engine type.

fault	fuel	brand	e_type	R_{rel}	R_{ppm}	S_{ft}
f1	d	b1	e1	0.23	0.42	1
f2	b	b1	e2	-0.12	0.18	0
f1	d	b2	e3	-0.50	-0.34	0
...	...	...	...	...	...	...

Figure 7. Example results of the combiner component based on the parameter-driven approach.

- *Result-driven approach*: This concept is based on the significance result of each information source. First, the fault must be classified as significant or not depending on the characteristic metrics. The result indicates whether based on the dataset the fault would be classified as significant (1 = significant, 0 = insignificant) (Equation (17)). Each result is multiplied with the weighting factor of the corresponding fault/data source-combination and the results of the data sources are aggregated. The last step is to divide the value by the sum of the weights of the data sources. The fault is significant if the result is greater than 0.5, and insignificant otherwise (Equation (18)).

$$S_{i,ft} = \begin{cases} k_{rel,i,ft} > k_{rel,th} \cap k_{ppm,i,ft} > k_{ppm,th}, & 1 \\ k_{rel,i,ft} \leq k_{rel,th} \cup k_{ppm,i,ft} \leq k_{ppm,th}, & 0 \end{cases} \tag{17}$$

$$S_{ft} = \begin{cases} \frac{\sum_{j=1}^{n} W_{i,ft} * S_{i,ft}}{\sum_{i=1}^{n} W_{i,ft}} > 0.5, & 1 \\ \frac{\sum_{j=1}^{n} W_{i,ft} * S_{i,ft}}{\sum_{i=1}^{n} W_{i,ft}} \leq 0.5, & 0 \end{cases} \quad (18)$$

Figure 8 shows an example output table of the combiner component based on the result-driven approach. As illustrated, the result is defined for the particular analysis set, which consists of fault type, fuel type, car brand, and engine type.

fault	fuel	brand	e_type	W_{sum}	S_{sum_w}	S_{ft}
f1	d	b1	e1	2.53	2.02	1
f2	b	b1	e2	1.45	0.58	0
f1	d	b2	e3	2.67	0.51	0
...	...	...	...	...	...	...

Figure 8. Example results of the combiner component based on the result-driven approach.

4) Controller: The controller component calculates the prediction accuracy of the fault time series' forecasts for each information source. This accuracy metric is used to obtain a weighting factor as required by the combiner component. In the proposed approach, the prediction method is a one-step out-of-sample forecast that computes a value for the following week, the new value can be observed and compared with the prediction from the previous week. In Section III-D, different prediction accuracy metrics were discussed. According to the classification proposed there, *relative errors* and *relative measures* do not meet the requirements, because they represent relative values between the accuracy of the method applied and a benchmark method. The drawback is that if the benchmark method leads to poor results, the calculated metric would possibly indicate a good accuracy. This is even more serious in the proposed approach, since the goal is to weight predictions based on different data sources. For example, if the benchmark method of a data source achieves poor results and the prediction is relatively good in comparison, the data source will be weighted more favorably than a data source where the benchmark method performs very well and the method used for the prediction is not as good in comparison. Metrics that belong to the class *scale-dependent measures* are also excluded, since they are scale dependent. For example, when a particular fault occurs more often in one data source than in another, this difference would influence the outcome, because it is not possible to compare them. Since the *MASE* metric needs more than one prediction for computation, it cannot be used in the proposed approach. Consequently, the remaining errors in the *percentage errors* category are *MAPE*, *MdAPE*, *sMAPE*, *sMdAPE*, *RMSPE*, and *RMdSPE*. Since the error metric in the proposed concept is calculated for a single forecast, there is no difference in the results between the versions using the mean and those using the median. Therefore, for this approach three different relevant metrics can be distinguished *APE*, *sAPE*, and *RSPE*.

The calculation of the *Mean Absolute Percentage Error (MAPE)* is given in Equation (19) [39].

$$e_{MAPE} = \frac{1}{n} * \sum_{i=1}^{n} \frac{|X_i - F_i|}{X_i} * 100 \quad (19)$$

The *symmetric Mean Absolute Error (sMAPE)* is defined as shown in Equation (20) [50].

$$e_{sMAPE} = \frac{1}{n} * \sum_{i=1}^{n} \frac{|X_i - F_i|}{(X_i + F_i)/2} * 100 \quad (20)$$

This equation shows that the sMAPE can take values between 0 and +200 (or - without the multiplier at the end - values between 0 and 2). A drawback of the error metric is that it is not symmetrical: Let us assume that observation X_i is the same for two information sources and has the value 50. The first data source predicts a value of 45 and the second data source predicts 55. Thus, both predictions have the same difference of 5, but one is too high and one too low. The sMAPE for the first data source is then 10.5% and for the second 9.5%. Despite this asymmetry in the results, sMAPE is used in scientific papers to determine the quality of forecasts (e.g., in the M3-Competition [50][51]).

The computation of the *Root Mean Square Percentage Error (RMSPE)* is given in Equation (21) [40].

$$e_{RMSPE} = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (\frac{|X_i - F_i|}{X_i})^2} * 100 \quad (21)$$

When dealing with a single future prediction the equation can be reduced to (M)APE, as the square root and the power of two can be eliminated.

Since sMAPE constitutes a good measure that can be transformed to the range between 0 to 1 (by removing the multiplier in the denominator), it is a good weighting factor for the proposed approach. Prediction accuracy can thus be calculated as shown in Equation (22).

$$P_{sMAPE} = 1 - \frac{|X_i - Fi|}{X_i + F_i} \quad (22)$$

Alternatively, MAPE could be used for this purpose. However, it is not ideal as a weighting factor, because it cannot be accurately transformed to the range between 0 and 1. A way of using MAPE to determine the prediction accuracy is shown in Equation (23).

$$P_{MAPE} = \begin{cases} \frac{|X_i - F_i|}{X_i} \leq 1, & \frac{|X_i - F_i|}{X_i} \\ \frac{|X_i - F_i|}{X_i} > 1, & 0 \end{cases} \quad (23)$$

Note that not only the prediction accuracy of the current week should be considered in the calculation of the weighting factor. The following example explains why: Let us assume that a specific information source achieved good prediction accuracy in recent weeks and performs poorly in the current week. If the accuracy based on a single week were used, the quality indicator of the data source would decrease drastically. Conversely, if an information source with very low prediction accuracy in previous weeks performs well in the current week, then the weighting should not be based only on this single (good) result. Therefore, the calculation in the proposed

concept of the weighting factor takes also previous prediction accuracies into account as shown in Equation (24).

$$W_{cont,d_i,fault} = \frac{P_{t-1} + P_t}{2} \quad (24)$$

Figure 9 illustrates the structure and example instances of the controller output table. An entry is defined by the dataset (represented by the data source column) and the analysis set (the attributes fault, fuel, car brand, and engine type). The remaining columns define the results of the controller component, and W_{cont} stores the final weighting factors used by the combiner component.

data source	fault	fuel	brand	e_type	P_{t-1}	P_t	W_{cont}
data source 1	f1	d	b1	e1	0.87	0.91	0.89
data source 1	f2	b	b1	e2	0.99	0.58	0.79
data source 2	f2	b	b1	e2	0.69	0.78	0.74
data source 3	f1	d	b2	e3	0.71	0.83	0.77
...	...	...	...	...	...	...	...

Figure 9. Example results of the controller component.

C. Q-AURA Integration

This section focuses on the integration of the presented concept into the Q-AURA application. Q-AURA comprises six steps: The first identifies which faults are significant and should be analyzed further. The presented concept optimizes this task by enabling earlier detection. The interface between this and the subsequent step is defined on a metric that indicates whether an analysis set is significant. Since the proposed concept uses the same representation of results, the original step can be substituted with the new approach. The improved approach including the optimization is illustrated in Figure 10.

Using an interface between the first and the second step eases this substitution. The second step needs only information about which faults are significant depending on fuel type, car brand, and engine type.

V. Conclusion and Future Work

The presented concept, which is currently evaluated, improves Q-AURA by an earlier identification of faults with negative trends. Q-AURA has been developed in cooperation with the industrial partner *BMW Motoren GmbH* (engine manufacturing plant) and is already in daily use by quality management experts at different engine manufacturing plants. It is an application that identifies significant faults, which are then examined in more detail. Bills of materials containing information about parts, components, and technical modifications are analyzed to determine modifications that are most likely the cause of a particular fault.

In this work, a concept has been proposed that addresses the challenge of earlier detection of critical faults. At the heart of the presented approach is the integration of different datasets that provide different views of warranty data. Data quality metrics are used to determine how accurate and correct the information from the different datasets is. Next, the fault course of each dataset is analyzed to predict the most likely value for the following week. Regression analysis is applied to a six-week period (using the predicted value and the last

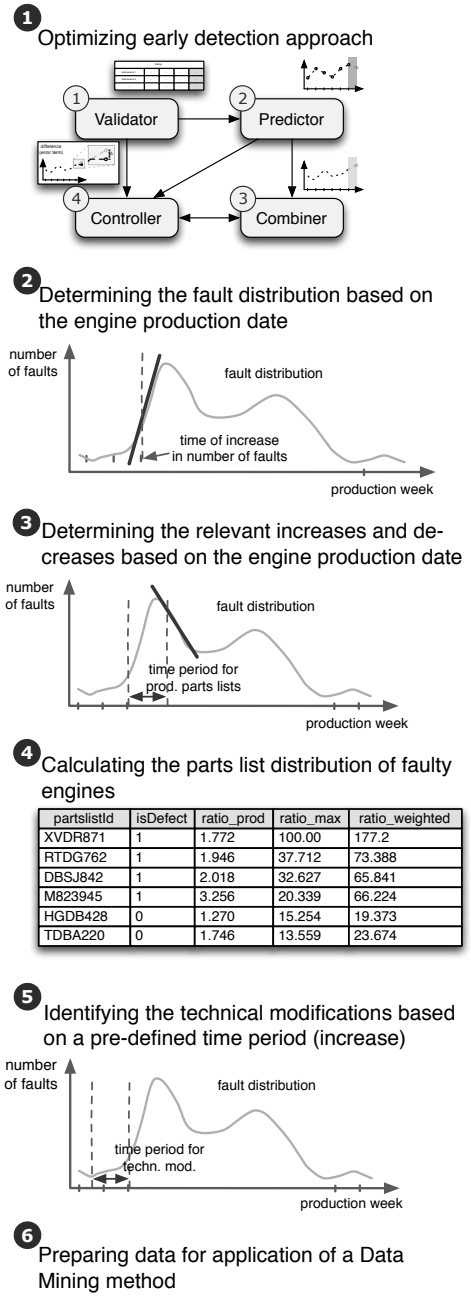


Figure 10. Integration of the proposed approach into Q-AURA.

five observations), which yields the characteristic values of the resulting regression line. The prediction accuracy is determined using predictions from the previous week and observations from the current week. In order to decide whether a fault is significant, weighting factors based on the calculated data quality metric and the prediction accuracy are used in addition to the results of the regression analysis. The approach is applied on warranty information in the automotive industry, but the concept could also be used in other application areas where time series and forecasts from different datasets must be combined to determine whether a particular course is significant. The definition of *significance* must be evaluated and determined in each application area. Depending on the use

case, other data quality metrics are potentially interesting for integration in the overall data quality metric. The three metrics used in this concept were chosen with care to be data-centric and to not take data representation or feature availability into account.

A further possible improvement would be the integration of an additional weighting factor depending on expert input. In some cases, domain experts have additional information about the datasets and would prefer an additional weighting factor that represents their view. This means that three factors would be used to determine the weighting: (i) the overall data quality metric of the dataset, (ii) the prediction accuracy of the dataset's time series analysis, and (iii) the preference metric based on expert input.

Another possible enhancement is to investigate whether applying different time series and forecasting methods for each dataset and subsequent combination of the forecasts yields more robust predictions and thus better results. Various research papers ([3][6][52][53]) have addressed such combination approaches, which are already used in the field of machine learning [54].

REFERENCES

[1] T. Leitner, C. Feilmayr, and W. Wöß, "Early Detection of Critical Faults Using Time-Series Analysis on Heterogeneous Information Systems in the Automotive Industry," in Third International Conference on Data Analytics, F. Laux, P. M. Pardalos, and C. Alain, Eds. International Academy, Research, and Industry Association (IARIA), 2014, pp. 70–75.

[2] C. K. Chan, B. G. Kingsman, and H. Wong, "The value of combining forecasts in inventory management - a case study in banking." European Journal of Operational Research, vol. 117, no. 2, 1999, pp. 199–210.

[3] A. Widodo and I. Budi, "Combination of time series forecasts using neural network," in International Conference on Electrical Engineering and Informatics (ICEEI), 2011, pp. 1–6.

[4] A. Kleyner and P. Sandborn, "A warranty forecasting model based on piecewise statistical distributions and stochastic simulation," Reliability Engineering and System Safety, vol. 88, no. 3, 2005, pp. 207–214.

[5] J. M. Montgomery, F. M. Hollenbach, and M. D. Ward, "Calibrating Ensemble Forecasting Models with Sparse Data in the Social Sciences (accepted and in press)," International Journal of Forecasting, -.

[6] J. S. Armstrong, "Combining Forecasts," in Principles of forecasting. Springer US, 2001, pp. 417–439.

[7] T. C. Redman, "The Impact of Poor Data Quality on the Typical Enterprise," Communications of the ACM, vol. 41, no. 2, 1998, pp. 79–82.

[8] B. Heinrich, M. Kaiser, and M. Klier, "How to measure Data Quality? A Metric Based Approach," in Proceedings of the 28th International Conference on Information Systems (ICIS). Montreal, Canada: Association for Information Systems, 2007.

[9] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for Data Quality Assessment and Improvement," ACM Computing Surveys, vol. 41, no. 3, 2009, pp. 1–52.

[10] C. Batini and M. Scannapieco, Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications). Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[11] R. Y. Wang and D. M. Strong, "Beyond Accuracy: What Data Quality Means to Data Consumers," Journal of Management Information Systems, vol. 12, no. 4, 1996, pp. 5–33.

[12] D. M. Strong, Y. W. Lee, and R. Y. Wang, "Data Quality in Context," Communications of the ACM, vol. 40, no. 5, 1997, pp. 103–110.

[13] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang, "Aimq: A methodology for information quality assessment," Information and Management, vol. 40, no. 2, 2002, pp. 133–146.

[14] F. Naumann and C. Rolker, "Assessment methods for Information Quality Criteria," in Fifth Conference on Information Quality (IQ 2000), Cambridge, MA, USA, 2000.

[15] D. P. Ballou and H. L. Pazer, "Modeling Completeness Versus Consistency Tradeoffs in Information Decision Contexts," IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 1, 2003, pp. 240–243.

[16] M. Ge and M. Helfert, "A review of information quality research – develop a research agenda," in International Conference on Information Quality, 2007, pp. 76–91.

[17] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," Communications of the ACM, vol. 45, no. 4, 2002, pp. 211–218.

[18] M. Bobrowski, M. Marré, and D. Yankelevich, "A Homogeneous Framework to Measure Data Quality," in Information Quality, Y. W. Lee and G. K. Tayi, Eds. MIT, 1999, pp. 115–124.

[19] B. K. Kahn, D. M. Strong, and R. Y. Wang, "Information Quality Benchmarks: Product and Service Performance," Communications of the ACM, vol. 45, no. 4, 2002, pp. 184–192.

[20] A. Motro and I. Rakov, "Estimating the Quality of Data in Relational Databases," in Proceedings of the Conference on Information Quality. MIT, 1996, pp. 94–106.

[21] A. Motro and I. Rakov, "Estimating the quality of databases," in Proceedings of the Third International Conference on Flexible Query Answering Systems (FQAS), T. Andreasen, H. Christiansen, and H. L. Larsen, Eds., vol. 1495. Springer Verlag, 1998, pp. 298–307.

[22] A. Motro, "Integrity = Validity + Completeness," ACM Transactions on Database Systems, vol. 14, no. 4, 1989, pp. 480–502.

[23] Y. Wand and R. Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations," Communications of the ACM, vol. 39, no. 11, 1996, pp. 86–95.

[24] T. C. Redman, Data Quality for the Information Age, 1st ed. Norwood, MA, USA: Artech House, Inc., 1996.

[25] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis, Fundamentals of Data Warehouses. Springer Verlag, 1995.

[26] M. Bovee, R. P. Srivastava, and B. Mak, "A conceptual framework and belief-function approach to assessing overall information quality." International Journal of Intelligent Systems, vol. 18, no. 1, 2003, pp. 51–74.

[27] F. Naumann, Quality-driven Query Answering for Integrated Information Systems. Berlin, Heidelberg: Springer-Verlag, 2002.

[28] L. Liu and L. Chi, "Evolutional data quality: A theory-specific view." in International Conference on Information Quality, C. Fisher and B. N. Davidson, Eds. MIT, 2002, pp. 292–304.

[29] D. P. Ballou and H. L. Pazer, "Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems," Management Science, vol. 31, no. 2, 1985, pp. 150–162.

[30] R. H. Shumway and D. S. Stoffer, Time Series Analysis and Its Applications: With R Examples, 3rd ed. Springer Texts in Statistics, 2011.

[31] P. S. P. Cowpertwait and A. V. Metcalfe, Introductory Time Series with R, 1st ed. Springer Publishing Company, Incorporated, 2009.

[32] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing," Journal of the American Statistical Association (JASA), vol. 106, no. 496, 2011, pp. 1513–1527.

[33] A. M. D. Livera, "Automatic forecasting with a modified exponential smoothing state space framework," Monash University, Department of Econometrics and Business Statistics, Monash Econometrics and Business Statistics Working Papers 10/10, 2010.

[34] A. B. Koehler, R. J. Hyndman, R. D. Snyder, and K. Ord, "Prediction intervals for exponential smoothing using two new classes of state space models," Journal of Forecasting, vol. 24, no. 1, 2005, pp. 17–37.

[35] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," International Journal of Forecasting, vol. 18, no. 3, 2002, pp. 439–454.

[36] J. D. Croston, "Forecasting and stock control for intermittent demands," Operational Research Quarterly, vol. 23, no. 3, 1972, pp. 289–303.

[37] L. Shenstone and R. J. Hyndman, "Stochastic models underlying

Croston's method for intermittent demand forecasting," Journal of Forecasting, 2005.

[38] H. Thome, "Univariate Box/Jenkins-Modelle in der Zeitreihenanalyse (Univariate Box/Jenkins-models in time series analysis)," Historical Social Research, vol. 19, no. 3, 1994, pp. 5–77.

[39] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," International Journal of Forecasting, 2006.

[40] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," International Journal of Forecasting, vol. 22, no. 4, 2006, pp. 679–688.

[41] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," Journal of Forecasting, vol. 1, no. 2, 1982, pp. 111–153.

[42] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," International Journal of Forecasting, vol. 8, no. 1, 1992, pp. 69–80.

[43] S. Makridakis, "Accuracy measures: theoretical and practical concerns," International Journal of Forecasting, vol. 9, no. 4, 1993, pp. 527–529.

[44] P. Goodwin and R. Lawton, "On the asymmetry of the symmetric MAPE," International Journal of Forecasting, vol. 15, no. 4, 1999, pp. 405–408.

[45] T. Leitner, C. Feilmayr, and W. Wöß, "Optimizing Reaction and Processing Times in Automotive Industry's Quality Management - A Data Mining Approach," in International Conference Data Warehousing and Knowledge Discovery (DaWaK), ser. Lecture Notes in Computer Science, L. Bellatreche and M. K. Mohania, Eds., vol. 8646. Springer-Verlag, 2014, pp. 266–273.

[46] G. U. Yule, "On the Theory of Correlation," Journal of the Royal Statistical Society, vol. 60, no. 4, 1897, pp. 812–854.

[47] A. Vaisman and E. Zimányi, Data Warehouse Systems: Design and Implementation. Springer-Verlag, 2014.

[48] F. X. Diebold and R. S. Mariano, "Comparing Predictive Accuracy," Journal of Business & Economic Statistics, vol. 13, no. 3, 1995, pp. 253–263.

[49] M. Mittlböck and M. Schemper, "Explained Variation for Logistic Regression," Statistics in medicine, vol. 15, no. 19, 1996, pp. 1987–1997.

[50] S. Makridakis and M. Hibon, "The M3-Competition: results, conclusions and implications," International Journal of Forecasting, vol. 16, no. 4, 2000, pp. 451–476.

[51] M. Hibon and T. Evgeniou, "To combine or not to combine: selecting among forecasts and their combinations," International Journal of Forecasting, vol. 21, no. 1, 2005, pp. 15–24.

[52] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," International Journal of Forecasting, vol. 5, no. 4, 1989, pp. 559–583.

[53] J. M. Bates and C. W. J. Granger, "The Combination of Forecasts," in Operational Research Society, vol. 20, no. 4, 1969, pp. 451–468.

[54] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

High-Speed Video Analysis of Ballistic Trials to Investigate Solver Technologies for the Simulation of Brittle Materials

Using the Example of Bullet-Proof Glass

Arash Ramezani and Hendrik Rothe
Chair of Measurement and Information Technology
University of the Federal Armed Forces
Hamburg, Germany
Email: ramezani@hsu-hh.de, rothe@hsu-hh.de

***Abstract*—Since computers and software have spread into all fields of industry, extensive efforts are currently made in order to improve the safety by applying certain numerical solutions. For many engineering problems involving shock and impact, there is no single ideal numerical method that can reproduce the various regimes of a problem. An approach wherein different techniques may be applied within a single numerical analysis can provide the "best" solution in terms of accuracy and efficiency. This paper presents a set of numerical simulations of ballistic tests, which analyze the effects of soda lime glass laminates, familiarly known as transparent armor. Transparent armor is one of the most critical components in the protection of light armored vehicles. The goal is to find an appropriate solver technique for simulating brittle materials and thereby improve bullet-proof glass to meet current challenges. To have the correct material model available is not enough. In this work, the main solver technologies are compared to create a perfect simulation model for soda lime glass laminates. The calculation should match ballistic trials and be used as the basis for further studies. In view of the complexity of penetration processes, it is not surprising that the bulk of work in this area is experimental in nature. Terminal ballistic test techniques, aside from routine proof tests, vary mainly in the degree of instrumentation provided and hence the amount of data retrieved. Here, the ballistic trials and the methods of analysis are discussed in detail. The numerical simulations are performed with the nonlinear dynamic analysis computer code ANSYS AUTODYN.**

Keywords-solver technologies; simulation models; brittle materials; high-performance computing; armor systems.

I. INTRODUCTION

In the security sector, the partly insufficient safety of people and equipment due to failure of industrial components are ongoing problems that cause great concern. Since computers and software have spread into all fields of industry, extensive efforts are currently made in order to improve the safety by applying certain computer-based solutions. To deal with problems involving the release of a large amount of energy over a very short period of time, e.g., explosions and impacts, there are three approaches, which are discussed in detail in [1].

As the problems are highly non-linear and require information regarding material behavior at ultra-high loading rates, which is generally not available, most of the work is experimental and may cause tremendous expenses. Analytical approaches are possible if the geometries involved are relatively simple and if the loading can be described through boundary conditions, initial conditions, or a combination of the two. Numerical solutions are far more general in scope and remove any difficulties associated with geometry [2].

For structures under shock and impact loading, numerical simulations have proven to be extremely useful. They provide a rapid and less expensive way to evaluate new design ideas. Numerical simulations can supply quantitative and accurate details of stress, strain, and deformation fields that would be very costly or difficult to reproduce experimentally. In these numerical simulations, the partial differential equations governing the basic physics principles of conservation of mass, momentum, and energy are employed. The equations to be solved are time-dependent and nonlinear in nature. These equations, together with constitutive models describing material behavior and a set of initial and boundary conditions, define the complete system for shock and impact simulations.

The governing partial differential equations need to be solved in both time and space domains (see Fig. 1). The solution over the time domain can be achieved by an explicit method. In the explicit method, the solution at a given point in time is expressed as a function of the system variables and parameters, with no requirements for stiffness and mass matrices. Thus, the computing time at each time step is low but may require numerous time steps for a complete solution.

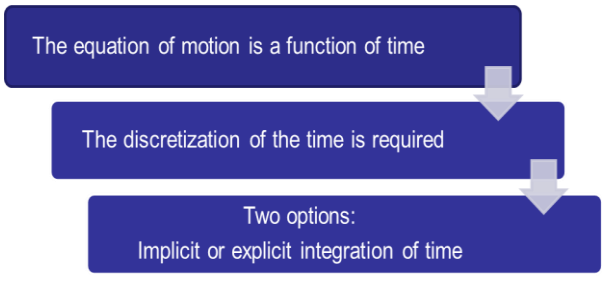


Figure 1. Discretization of time and space is required.

The solution for the space domain can be obtained utilizing different spatial discretizations, such as Lagrange [3], Euler [4], Arbitrary Lagrange Euler (ALE) [5], or mesh free methods [6]. Each of these techniques has its unique capabilities, but also limitations. Usually, there is not a single technique that can cope with all the regimes of a problem [7].

This work will focus on brittle materials and transparent armor (consisting of several layers of soda lime float glass bonded to a layer of polycarbonate to produce a glass laminate). Using a computer-aided design (CAD) neutral environment that supports direct, bidirectional and associative interfaces with CAD systems, the geometry can be optimized successively. Native CAD geometry can be used directly, without translation to IGES or other intermediate geometry formats [8]. An example is given in Fig. 2.

The work will also provide a brief overview of ballistic tests to offer some basic knowledge of the subject, serving as a basis for the comparison and verification of the simulation results. Details of ballistic trials on transparent armor systems are presented. Here, even the crack formation must precisely match later simulations. It was possible to observe crack motion and to accurately measure crack velocities in glass laminates. The measured crack velocity is a complicated function of stress and of water vapor concentration in the environment [9].

The objective of this work is to compare current solver technologies to find the most suitable simulation model for brittle materials. Lagrange, Euler, ALE, and "mesh free" methods, as well as coupled combinations of these methods, are described and applied to a bullet-proof glass laminate structure impacted by a projectile. It aims to clarify the following issue: What is the most suitable simulation model for brittle materials?

The results shall be used to improve the safety of ballistic glasses. Instead of running expensive trials, numerical simulations should be applied to identify vulnerabilities of structures. Contrary to the experimental results, numerical methods allow easy and comprehensive studying of all mechanical parameters.

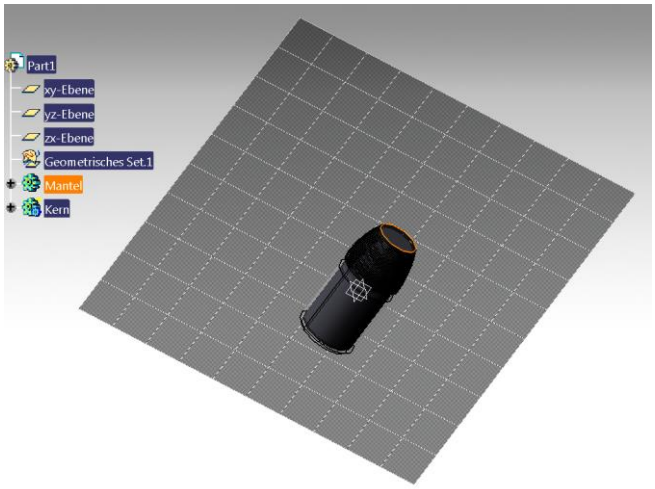


Figure 2. Native CAD geometry (.44 Remington Magnum).

Modeling will also help to understand how the transparent armor schemes behave during impact and how the failure processes can be controlled to our advantage. By progressively changing the composition of several layers and the material thickness, the transparent armor will be optimized.

After a brief introduction and description of the different methods of space discretization, there is a short section on ballistic trials, where the experimental set-up is depicted. The last section describes the numerical simulations. These paragraphs of analysis are followed by a conclusion.

II. STATE-OF-THE-ART

First approaches for optimization were already developed in 1999. Mike Richards, Richard Clegg, and Sarah Howlett investigated the behavior of glass laminates in various configurations at a constant total thickness [10]. Resulting from the experimental studies, numerical simulations were created and adjusted to the experimental results using 2D-Lagrange elements only.

Pyttel, Liebertz, and Cai explored the behavior of glass upon impact with three-dimensional Lagrange elements [11]. A failure criterion was presented and implemented in an explicit finite element solver. The main idea of this criterion is that a critical energy threshold must be reached over a finite region before failure can occur. Afterwards, crack initiation and growth is based on a local Rankine (maximum stress) criterion. Different strategies for modeling laminated glass were also discussed. To calibrate the criterion and evaluate its accuracy, a wide range of experiments with plane and curved specimens of laminated glass were done. For all experiments finite element simulations were performed. In 2011, these studies were used to analyze crash behavior.

In the same year, Zang and Wang dealt with the impact behavior on glass panels in the automotive sector [12]. In doing so, self-developed methods of numerical simulation were supposed to be compared with commercial codes. The impact process of a single glass plane and a laminated glass plane were calculated in the elastic range by the code. Furthermore, the impact fracture process of a single glass plane and a laminated glass plane were simulated respectively. The entire failure processes in detail were presented. For the first time, mesh-free methods were applied, although these were not coupled with other solver technologies.

In this study, different methods for the simulation of safety glass will be introduced. In so doing, the possibility of coupling various solver technologies will be discussed and illustrated by means of an example. For the first time, glass laminates will be modeled using coupled methods. Techniques previously applied, show considerable shortcomings in portraying the crack and error propagation in the glass. Mesh-free approaches, in turn, do not correctly present the behavior of synthetic materials. To overcome the shortcomings of these single-method approaches, this paper will present an optimal solution to the problem by combining two methods.

III. METHODS OF SPACE DISCRETIZATION

The spatial discretization is performed by representing the fields and structures of the problem using computational points in space, usually connected with each other through computational grids. Generally, the following applies: the finer the grid, the more accurate the solution. For problems of dynamic fluid-structure interaction and impact, there typically is no single best numerical method which is applicable to all parts of a problem. Techniques to couple types of numerical solvers in a single simulation can allow the use of the most appropriate solver for each domain of the problem [13].

The most commonly used spatial discretization methods are Lagrange, Euler, ALE (a mixture of Lagrange and Euler), and mesh-free methods, such as Smooth Particles Hydrodynamics (SPH) [14].

A. Lagrange

The Lagrange method of space discretization uses a mesh that moves and distorts with the material it models as a result of forces from neighboring elements (meshes are imbedded in material). There is no grid required for the external space, as the conservation of mass is automatically satisfied and material boundaries are clearly defined. This is the most efficient solution methodology with an accurate pressure history definition.

The Lagrange method is most appropriate for representing solids, such as structures and projectiles. If however, there is too much deformation of any element, it results in a very slowly advancing solution and is usually terminated because the smallest dimension of an element results in a time step that is below the threshold level.

B. Euler

The Euler (multi-material) solver utilizes a fixed mesh, allowing materials to flow (advect) from one element to the next (meshes are fixed in space). Therefore, an external space needs to be modeled. Due to the fixed grid, the Euler method avoids problems of mesh distortion and tangling that are prevalent in Lagrange simulations with large flows. The Euler solver is very well-suited for problems involving extreme material movement, such as fluids and gases. To describe solid behavior, additional calculations are required to transport the solid stress tensor and the history of the material through the grid. Euler is generally more computationally intensive than Lagrange and requires a higher resolution (smaller elements) to accurately capture sharp pressure peaks that often occur with shock waves.

C. ALE

The ALE method of space discretization is a hybrid of the Lagrange and Euler methods. It allows redefining the grid continuously in arbitrary and predefined ways as the calculation proceeds, which effectively provides a continuous rezoning facility. Various predefined grid motions can be specified, such as free (Lagrange), fixed (Euler), equipotential, equal spacing, and others. The ALE method can model solids as well as liquids. The advantage of ALE is the ability to reduce and sometimes eliminate difficulties caused by severe mesh distortions encountered by the Lagrange method, thus allowing a calculation to continue efficiently. However, compared to Lagrange, an additional computational step of rezoning is employed to move the grid and remap the solution onto a new grid [7].

D. SPH

The mesh-free Lagrangian method of space discretization (or SPH method) is a particle-based solver and was initially used in astrophysics. The particles are imbedded in material and they are not only interacting mass points but also interpolation points used to calculate the value of physical variables based on the data from neighboring SPH particles, scaled by a weighting function. Because there is no grid defined, distortion and tangling problems are avoided as well. Compared to the Euler method, material boundaries and interfaces in the SPH are rather well defined and material separation is naturally handled. Therefore, the SPH solver is ideally suited for certain types of problems with extensive material damage and separation, such as cracking. This type of response often occurs with brittle materials and hypervelocity impacts. However, mesh-free methods, such as Smooth Particles Hydrodynamics, can be less efficient than mesh-based Lagrangian methods with comparable resolution.

Fig. 3 gives a short overview of the solver technologies mentioned above. The crucial factor is the grid that causes different outcomes.

The behavior (deflection) of the simple elements is well-known and may be calculated and analyzed using simple equations called shape functions. By applying coupling conditions between the elements at their nodes, the overall stiffness of the structure may be built up and the deflection/distortion of any node – and subsequently of the whole structure – can be calculated approximately [16].

Due to the fact that all engineering simulations are based on geometry to represent the design, the target and all its components are simulated as CAD models [17]. Therefore, several runs are necessary: from modeling to calculation to the evaluation and subsequent improvement of the model (see Fig. 4).

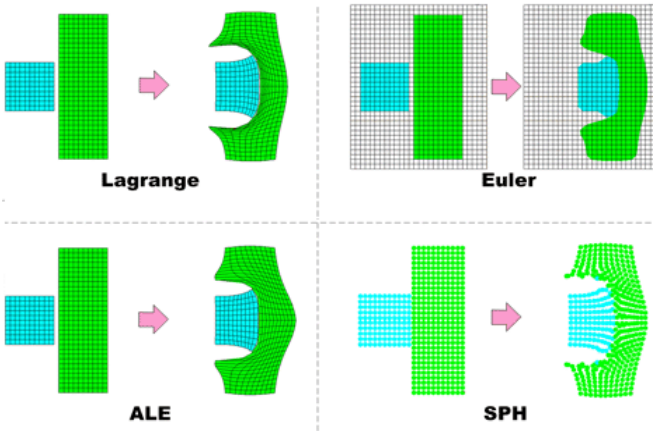


Figure 3. Examples of Lagrange, Euler, ALE, and SPH simulations on an impact problem [15].

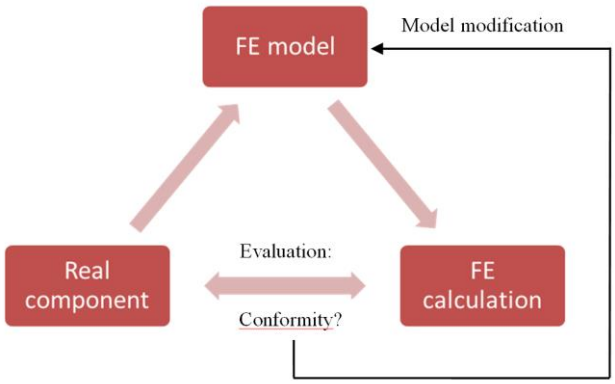


Figure 4. Iterative procedure of a typical FE analysis [16].

The most important steps during an FE analysis are the evaluation and interpretation of the outcomes followed by suitable modifications of the model. For that reason, ballistic trials are necessary to validate the simulation results. They can be used as the basis of an iterative optimization process.

IV. BALLISTIC TRIALS

Ballistics is an essential component for the evaluation of our results. Here, terminal ballistics is the most important sub-field. It describes the interaction of a projectile with its target. Terminal ballistics is relevant for both small and large caliber projectiles. The task is to analyze and evaluate the impact and its various modes of action. This will provide information on the effect of the projectile and the extinction risk.

Given that a projectile strikes a target, compressive waves propagate into both the projectile and the target. Relief waves propagate inward from the lateral free surfaces of the penetrator, cross at the centerline, and generate a high tensile stress. If the impact were normal, we would have a two-dimensional stress state. If the impact were oblique, bending stresses will be generated in the penetrator. When the compressive wave reached the free surface of the target, it would rebound as a tensile wave. The target may fracture at this point. The projectile may change direction if it perforates (usually towards the normal of the target surface). A typical impact response is illustrated in Fig. 5.

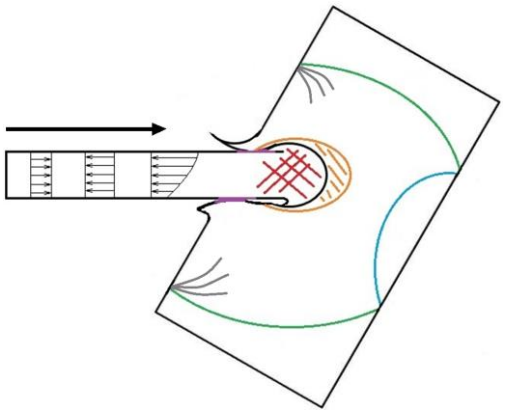


Figure 5. Wave propagation after impact.

Figure 6. Ballistic tests and the analysis of fragments.

Because of the differences in target behavior based on the proximity of the distal surface, we must categorize targets into four broad groups. A semi-infinite target is one where there is no influence of distal boundary on penetration. A thick target is one in which the boundary influences penetration after the projectile is some distance into the target. An intermediate thickness target is a target where the boundaries exert influence throughout the impact. Finally, a thin target is one in which stress or deformation gradients are negligible throughout the thickness.

There are several methods by which a target will fail when subjected to an impact. The major variables are the target and penetrator material properties, the impact velocity, the projectile shape (especially the ogive), the geometry of the target supporting structure, and the dimensions of the projectile and target.

In order to develop a numerical model, a ballistic test program is necessary. The ballistic trials are thoroughly documented and analyzed – even fragments must be collected. They provide information about the used armor and the projectile behavior after fire, which must be consistent with the simulation results (see Fig. 6).

In order to create a data set for the numerical simulations, several experiments have to be performed. Ballistic tests are recorded with high-speed videos and analyzed afterwards. The experimental set-up is shown in Fig. 7. Testing was undertaken at an indoor ballistic testing facility (see Fig. 8). The target stand provides support behind the target on all four sides. Every ballistic test program includes several trials with different glass laminates. The set-up has to remain unchanged.

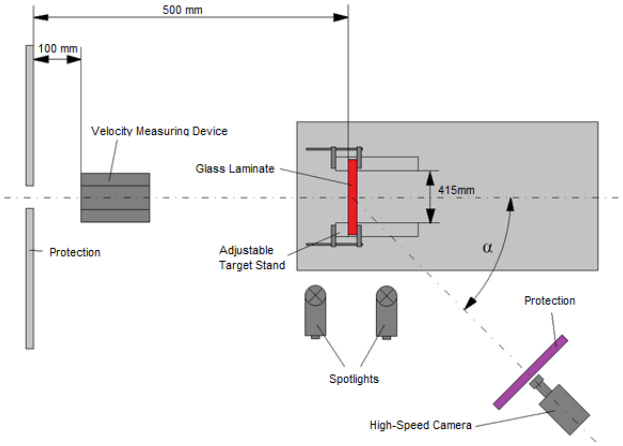


Figure 7. Experimental set-up.

The camera system is a pco.dimax that enables fast image rates of 1279 frames per second (fps) at full resolution of 2016 x 2016 pixels. The use of a polarizer and a neutral density filter is advisable, so that waves of some polarizations can be blocked while the light of a specific polarization can be passed.

Several targets of different laminate configurations were tested to assess the ballistic limit and the crack propagation for each design. The ballistic limit is considered the velocity required for a particular projectile to reliably (at least 50% of the time) penetrate a particular piece of material [18]. After the impact, the projectile is examined regarding any kind of change it might have undergone.

Fig. 9 shows a 23 mm soda lime glass target after testing. The penetrator used in this test was a .44 Remington Magnum, a large-bore cartridge with a lead base and copper jacket. The glass layers showed heavy cracking as a result of the impact.

Close to the impact point is the region of comminution. The comminuted glass is even ejected during the impact. Radial cracks have propagated away from the impact point. The polycarbonate backing layer is deformed up to the maximum bulge height when the velocity of the projectile is close to the ballistic limit. A large amount of the comminuted glass is ejected during the impact. Several targets of different laminate configurations were tested to assess the ballistic limit and the crack propagation for each design.

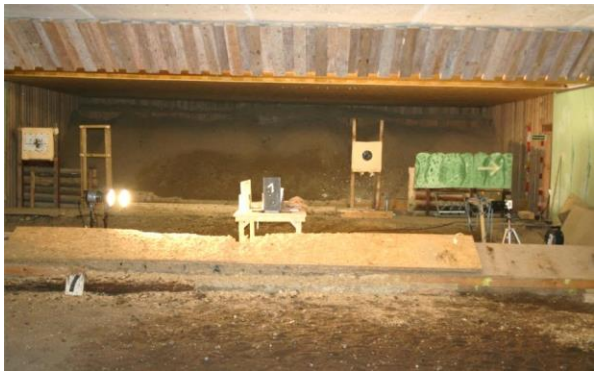


Figure 8. Indoor ballistic testing facility.

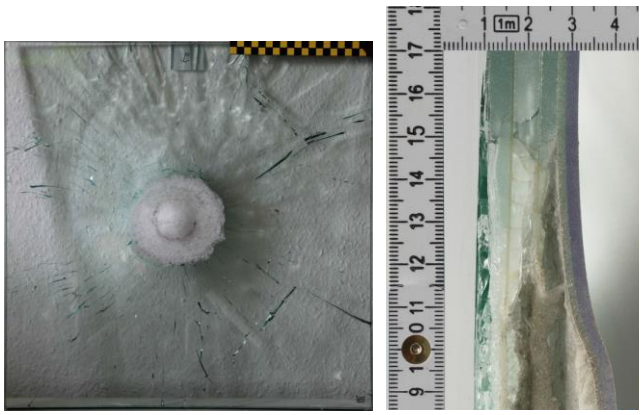


Figure 9. Trial observation with a 23mm glass laminate.

The crack propagation is analyzed using the software called COMEF [19], image processing software for highly accurate measuring functions. The measurement takes place via setting measuring points manually on the monitor. Area measurement is made by the free choice of grey tones (0…255). Optionally the object with the largest surface area can be recognized automatically as object. Smaller particles within the same grey tone range as the sample under test are automatically ignored by this filter.

Fig. 10 shows an example of measuring and analyzing cracks and Fig. 11 illustrates the propagation process in a path-time-diagram. However, caution must be taken when interpreting measurements of wave velocity from such sequences. Here, a distinction should be made between radial (red) and circular (yellow) propagation.

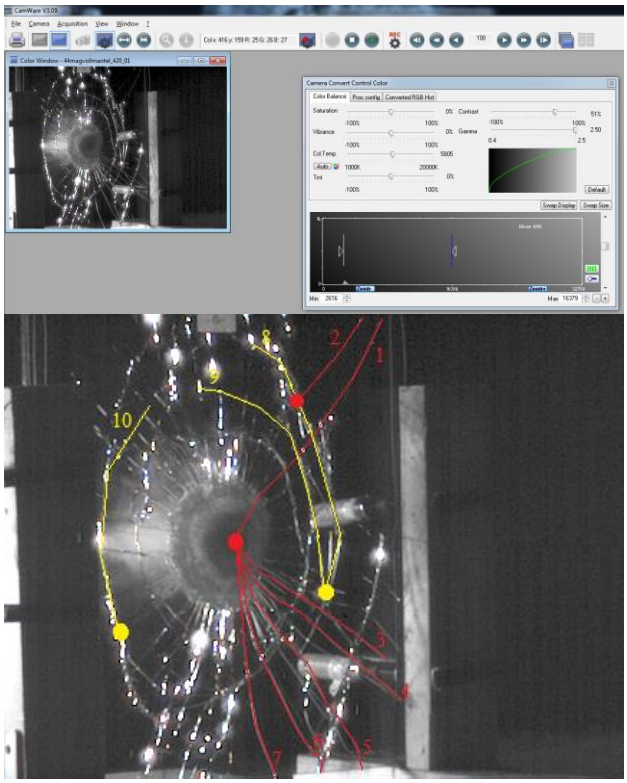


Figure 10. Analyzing crack propagation using COMEF.

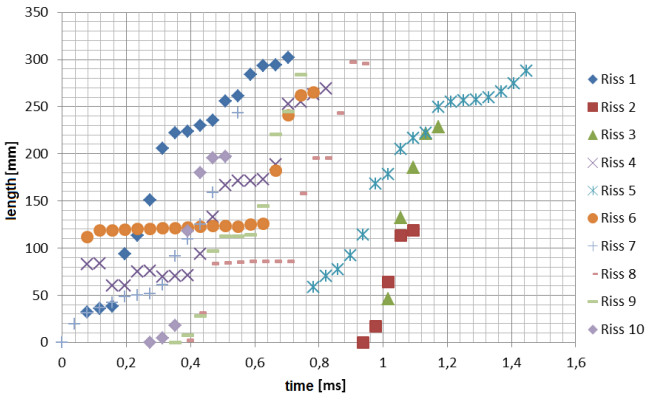


Figure 11. Analyzing the crack propagation over time.

Cracks propagate with a velocity up to 2500 m/s, which is similar to the values in the literature. The damage of a single glass layer starts with the impact of the projectile corresponding to the depth of the penetration. The polycarbonate layers interrupt the crack propagation and avoid piercing and spalling. The different types of impact are summarized in Fig. 12.

Spalling is very common and is the result of wave reflection from the rear face of the plate. It is common for materials that are stronger in compression than in tension. Scabbing is similar to spalling, but the fracture predominantly results from large plate deformation, which begins with a crack at a local inhomogeneity. Brittle fracture usually occurs in weak and lower density targets. Radial cracking is common in ceramic types of materials where the tensile strength is lower than the compressive strength, but it does occur in some steel armor. Plugging occurs in materials that are fairly ductile, usually when the projectile's impact velocity is very close to the ballistic limit. Petaling occurs when the radial and circumferential stresses are high and the projectile impact velocity is close to the ballistic limit [18]. The task is to analyze and evaluate the impact and its various modes of action. This will provide information on the effect of the projectile and the extinction risk.

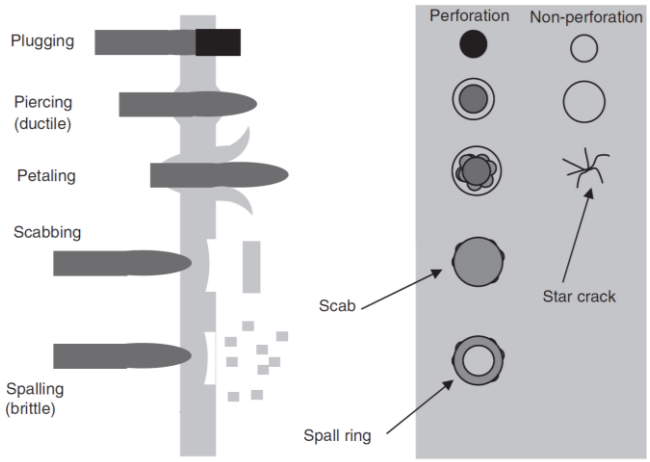


Figure 12. Target failure modes [18].

The first impact of a .44 Remington Magnum cartridge does not cause a total failure of our 23 mm soda lime glass target. Fragments of the projectile can be found in the impact hole. The last polycarbonate layer remains significantly deformed.

The results of the ballistic tests were provided prior to the simulation work to aid calibration. In this paper, a single trial will illustrate the general approach of the numerical simulations.

V. NUMERICAL SIMULATION

The ballistic tests are followed by computational modeling of the experimental set-up. Then, the experiment is reproduced using numerical simulations. Fig. 13 shows a cross-section of the ballistic glass and the projectile in a CAD model. The geometry and observed response of the laminate to ballistic impact is approximately symmetric to the axis through the bullet impact point. Therefore, a 2D axisymmetric approach was chosen.

Numerical simulation of transparent armor requires the selection of appropriate material models for the constituent materials and the derivation of suitable material model input data. The laminate systems studied here consist of soda lime float glass, polyurethane interlayer, polyvinyl butyral, and polycarbonate. Lead and copper models are also required for the .44 Remington Magnum cartridge.

The projectile was divided into two parts - the jacket and the base - which have different properties and even different meshes. These elements have quadratic shape functions and nodes between the element edges. In this way, the computational accuracy as well as the quality of curved model shapes increases. Using the same mesh density, the application of parabolic elements leads to a higher accuracy compared to linear elements (1st order elements).

Different solver technologies have been applied to the soda lime glass laminate. The comparison is presented in the following section.

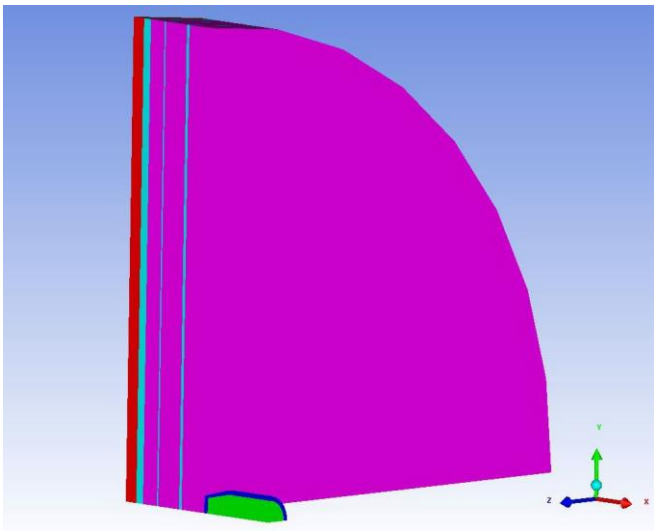


Figure 13. CAD model.

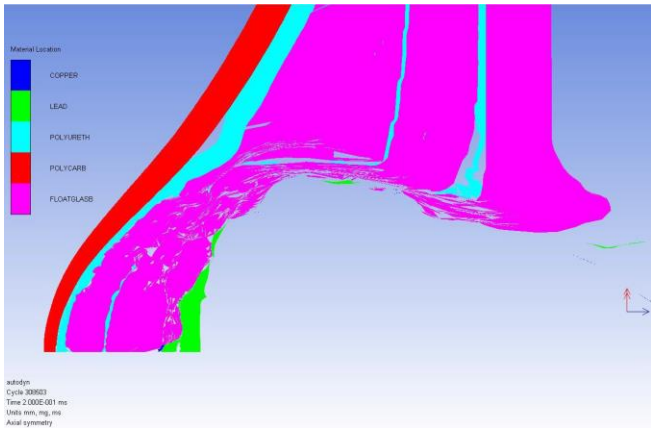


Figure 14. Lagrange method.

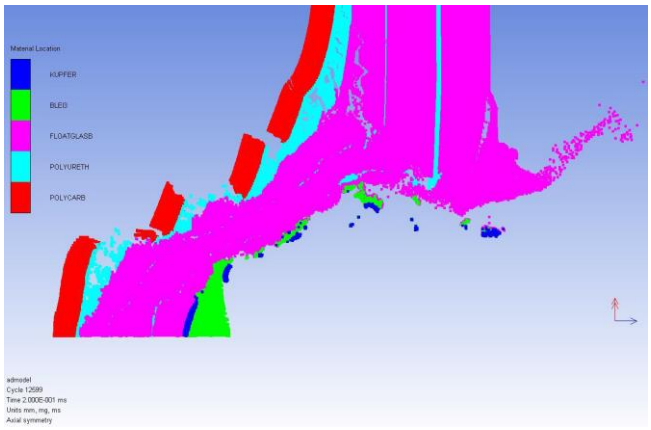


Figure 15. Mesh free Lagrangian method (SPH).

A. *Solver Evaluation*

Before the evaluation starts, it has to be noticed that the Euler method is not suitable for numerical simulations dealing with brittle materials. A major problem of Euler codes is determining the material transport. Since material flows through a fixed grid, some procedure must be incorporated in the code to move material to neighboring cells in all the coordinate dimensions. It is also necessary to identify the materials so that pressures can be calculated in cells carrying more than a single material.

Because the initial codes were designed to solve problems involving hypervelocity impact, where pressures generated on impact were orders of magnitude larger than material strength, the material was thought of as a fluid. Hence Euler codes are ideal for large deformation problems but contact is very difficult to determine without adding Lagrangian features.

Nowadays, it is generally used for representing fluids and gases, for example, the gas product of high explosives after detonation. To describe solid behavior, additional calculations are required. Cracking cannot be simulated adequately and the computation time is relatively high. For this reason, the Euler (and as a result the ALE) method will not be taken into consideration.

1) Lagrange method: Fig. 14 shows the simulation with a single Lagrange solver in the first iteration procedure. This method, as mentioned before, is well-suited for representing solids like structures and projectiles. The advantages are computational efficiency and ease of incorporating complex material models. The polyurethane interlayer, polyvinyl butyral and polycarbonate are simulated adequatly. While the soda lime glass also deforms well, the crack propagation cannot be displayed suitably with this solver.

2) Mesh free Lagrangian method (SPH): The mesh free Lagrangian method is not appropiate for simulating bullet-proof glass. The crack propagation and failure mode of the soda lime glass are very precise. The problem here however is the simulation of the layers. The particles do not provide the necessary cohesion (see Fig. 15). They break easily and then lose their function.

However, the SPH method requires some of the particles to locate current neighboring particles, which makes the computational time per cycle more expensive than mesh based Lagrangian techniques. For every increment in time, each particle must compare its position to all other particles in the computation and must build a neighbor list before the state variables can be updated. This can be a time-consuming process. Furthermore, the mesh free method is less efficient than mesh based Lagrangian methods with comparable resolution.

3) Coupled multi-solver approach (Lagrange and SPH): The coupled multi-solver approach uses SPH for the soda lime glass and Langrange for the polyurethane interlayer, polyvinyl butyral and polycarbonate. The grid consists of both SPH and Lagrange regions and transfers information from one to the other via boundary conditions. The crack propagation can be simulated precisely. The deformation of the last layer is accurately displayed and the failure mode matches the ballistic trial. Fig. 16 illustrates the simulation result for this case. This type of approach, where one body is much stiffer than the other requires a more elaborate time-step control than has a simple explicit scheme.

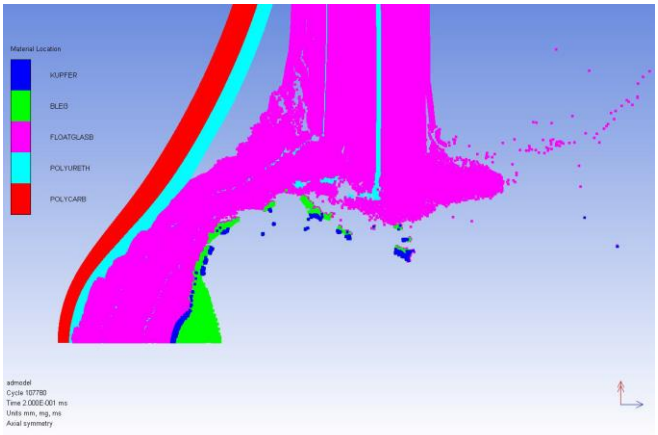


Figure 16. Coupled multi-solver approach (Lagrange and SPH).

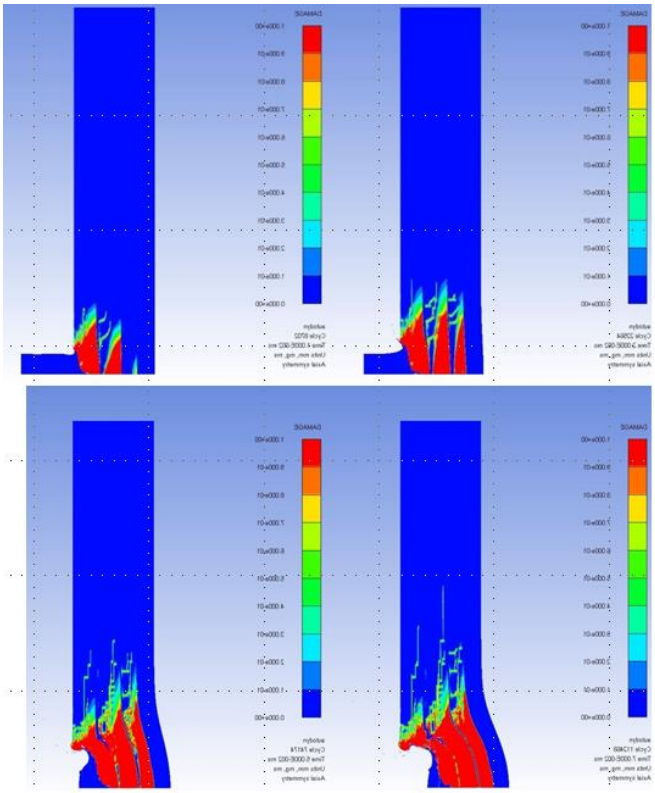


Figure 17. Crack propagation in a coupled multi-solver simulation model.

B. *Simulation Results*

With the coupled multi-solver and optimized material parameters, the simulation results adequately mirror the observations made in the ballistic experiments. Fragmentation and crack propagation are almost equal to the ballistic test shown in Fig. 9.

Fig. 17 illustrates the development of fracture after 10, 20, 50, and 70 µs due to shear induced micro-cracking (damage) in the glass during the penetration process. Note that the failure of the glass in the second and third layers spreads from the glass / polyurethane interlayers back towards the oncoming projectile. This rapid material failure is owed to a reduction in material strength as rarefaction waves from the interface reduce the confining pressure [18].

Small fragments are automatically deleted from the program to reduce computing time. Regarding the protection level of our structures, these fragments are hardly important.

The projectile is subject to a significant deformation. It gets stuck in the target and loses kinetic energy. Fig. 18 compares the numerical simulation of a .44 Remington impact with the experimental result.

A clear hole, 45-50 mm in diameter, is generated in the glass / polyurethane layers of the laminate. A comminuted region of glass, shows highly cracked and completely crushed material, of around 20 mm in diameter in the first layer which extends to around 120 mm in diameter in the last layer. Hence, the simulated diameter of comminution is almost identical to that observed experimentally.

Even the delamination of the layers can be reproduced in the simulation. The predicted height of the bulge from the flat region of the polycarbonate is 28 mm compared to approximately 8 mm observed in the ballistic trials. In the simulation, comminuted glass is caught between the bullet and the polycarbonate layer. This leads to a larger deformation. In reality, comminuted glass is ejected during the impact. The polycarbonate dishes from the edge of the support clamp to form a prominent bulge in the central region. Therefore, reducing the instantaneous geometric erosion strain of the soda lime glass will significantly improve results. Owed to the adopted calibration process, these simulation results correlate well with the experimental observations.

VI. HIGH-PERFORMANCE COMPUTING

The objective is to develop and improve the modern armor used in the security sector. To develop better, smarter constructions requires analyzing a wider range of parameters. However, there is a simple rule of thumb: the more design iterations that can be simulated, the more optimized is the final product. As a result, a high-performance computing (HPC) solution has to dramatically reduce overall engineering simulation time. HPC adds tremendous value to engineering simulation by enabling the creation of large, high-fidelity models that yield accurate and detailed insight into the performance of a proposed design. HPC also adds value by enabling greater simulation throughput. Using HPC resources, many design variations can be analyzed.

Beyond the use of HPC, the software is a key strategic enabler of large-scale simulations. The workload for the above mentioned simulations is specified in Fig. 19. The equation solver dominates the CPU time and consumes the most system resources (memory and I/O).

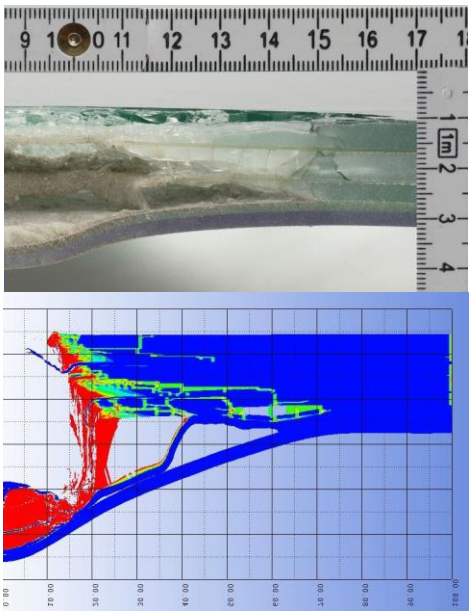


Figure 18. Comparison between simulation results and ballistic trial.

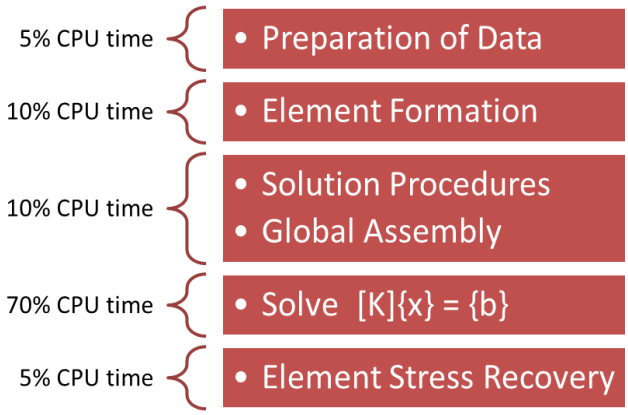


Figure 19. Comparison between simulation results and ballistic trial.

This research will evaluate the performance of the following server generations: HP ProLiant SL390s G7, HP ProLiant DL580 G7 and HP ProLiant DL380p G8.

To take into account the influence of the software, different versions of ANSYS will be applied here. Regarding the Lagrange solver and optimized material parameters in a simplified 2D simulation model (for the purpose of comparison), the following benchmark is obtained for the different simulations (see Table I).

The results indicate the importance of high-performance computing in combination with competitive simulation software to solve current problems of the computer-aided engineering sector.

VII. CONCLUSION

This work demonstrated how a small number of well-defined experiments can be used to develop, calibrate, and validate solver technologies used for simulating the impact of projectiles on complex armor systems and brittle materials.

Existing material models were optimized to reproduce ballistic tests. High-speed videos were used to analyze the characteristics of the projectile – before and after the impact. The simulation results demonstrate the successful use of the coupled multi-solver approach. The high level of correlation between the numerical results and the available experimental or observed data demonstrates that the coupled multi-solver approach is an accurate and effective analysis technique.

New concepts and models can be developed and easily tested with the help of modern hydrocodes. The initial design approach of the units and systems has to be as safe and optimal as possible. Therefore, most design concepts are analyzed on the computer.

TABLE I. BENCHMARK TO ILLUSTRATE THE INFLUENCE OF DIFFERENT SERVER AND SOFTWARE GENERATIONS

	ANSYS 14.5	**ANSYS 15.0**
SL390s G7	35m02s	18m59s
DL580 G7	27m08s	16m19s
DL380p G8	21m47s	12m55s

FEM-based simulations are well-suited for this purpose. Here, a numerical model has been developed, which is capable of predicting the ballistic performance of soda lime glass / polycarbonate transparent armor systems. Thus, estimates based on experience are being more and more replaced by software.

The gained experience is of prime importance for the development of modern armor. By applying the numerical model a large number of potential armor schemes can be evaluated and the understanding of the interaction between laminate components under ballistic impact can be improved.

The most important steps during an FE analysis are the evaluation and interpretation of the outcomes followed by suitable modifications of the model. For that reason, ballistic trials are necessary to validate the simulation results. They are designed to obtain information about

- the velocity and trajectory of the projectile prior to impact,
- changes in configuration of projectile and target due to impact,
- masses, velocities, and trajectories of fragments generated by the impact process.

Ballistic trials can be used as the basis of an iterative optimization process. Numerical simulations are a valuable adjunct to the study of the behavior of metals subjected to high-velocity impact or intense impulsive loading. The combined use of computations, experiments and high-strain-rate material characterization has, in many cases, supplemented the data achievable by experiments alone at considerable savings in both cost and engineering man-hours.

REFERENCES

[1] A. Ramezani and H. Rothe, "Investigation of Solver Technologies for the Simulation of Brittle Materials," The Sixth International Conference on Advances in System Simulation (SIMUL 2014) IARIA, Oct. 2014, pp. 236-242, ISBN 978-61208-371-1

[2] J. Zukas, "Introduction to Hydrocodes," Elsevier Science, February 2004.

[3] A. M. S. Hamouda and M. S. J. Hashmi, "Modelling the impact and penetration events of modern engineering materials: Characteristics of computer codes and material models," Journal of Materials Processing Technology, vol. 56, pp. 847–862, Jan. 1996.

[4] D. J. Benson, "Computational methods in Lagrangian and Eulerian hydrocodes," Computer Methods in Applied Mechanics and Engineering, vol. 99, pp. 235–394, Sep. 1992, doi: 10.1016/0045-7825(92)90042-I.

[5] M. Oevermann, S. Gerber, and F. Behrendt, "Euler-Lagrange/DEM simulation of wood gasification in a bubbling fluidized bed reactor," Particuology, vol. 7, pp. 307-316, Aug. 2009, doi: 10.1016/j.partic.2009.04.004.

[6] D. L. Hicks and L. M. Liebrock, "SPH hydrocodes can be stabilized with shape-shifting," Computers & Mathematics with Applications, vol. 38, pp. 1-16, Sep. 1999, doi: 10.1016/S0898-1221(99)00210-2.

[7] X. Quan, N. K. Birnbaum, M. S. Cowler, and B. I. Gerber, "Numerical Simulations of Structural Deformation under

Shock and Impact Loads using a Coupled Multi-Solver Approach," 5th Asia-Pacific Conference on Shock and Impact Loads on Structures, Hunan, China, Nov. 2003, pp. 152-161.

[8] N. V. Bermeo, M. G. Mendoza, and A. G. Castro, "Semantic Representation of CAD Models Based on the IGES Standard," Computer Science, vol. 8265, pp. 157-168, Dec. 2001, doi: 10.1007/ 978-3-642-45114-0_13.

[9] S. M. Wiederhorn, "Influence of Water Vapor on Crack Propagation in Soda-Lime Glass," Journal of the American Ceramic Society, vol. 50, pp. 407-414, Aug. 1967, doi: 10.1111/j.1151-2916.1967.tb15145.x.

[10] M. Richards, R. Clegg, and S. Howlett, "Ballistic Performance Assessment of Glass Laminates Through Experimental and Numerical Investigation," 18th International Symposium and Exhibition on Ballistics, San Antonio, Texas, Nov. 1999, pp. 1123-1131.

[11] T. Pyttel, H. Liebertz, and J. Cai, "Failure criterion for laminated glass under impact loading and its application in finite element simulation," International Journal of Impact Engineering, vol. 38, pp. 252-263, April 2011, doi: 10.1007/s00466-007-0170-1.

[12] M. Y. Zang, Z. Lei, and S. F. Wang, "Investigation of impact fracture behavior of automobile laminated glass by 3D discrete element method," Computational Mechanics, vol. 41, pp. 78-83, Dec. 2007, doi: 10.1007/s00466-007-0170-1.

[13] G. S. Collins, "An Introduction to Hydrocode Modeling," Applied Modelling and Computation Group, Imperial College London, August 2002, unpublished.

[14] R. F. Stellingwerf and C. A. Wingate, "Impact Modeling with Smooth Particle Hydrodynamics," International Journal of Impact Engineering, vol. 14, pp. 707–718, Sep. 1993.

[15] ANSYS Inc. Available Solution Methods. [Online]. Available from: http://www.ansys.com/Products/Simulation+Technology/Structural+Analysis/Explicit+Dynamics/Features/Available+Solution+Methods [retrieved: April, 2014]

[16] P. Fröhlich, "FEM Application Basics," Vieweg Verlag, September 2005.

[17] H. B. Woyand, "FEM with CATIA V5," J. Schlembach Fachverlag, April 2007.

[18] D. E. Carlucci and S. S. Jacobson, "Ballistics: Theory and Design of guns and ammunition," CRC Press, Dec. 2008.

[19] OEG Gesellschaft für Optik, Elektronik & Gerätetechnik mbH. [Online]. Available from: http://www.oeg-messtechnik.de/?p=5&l=1 [retrieved: March, 2015]

A Rare Event Method Applied to Signalling Cascades

Benoît Barbot, Serge Haddad and Claudine Picaronny
LSV, ENS Cachan & CNRS & Inria,
61, avenue du Président Wilson Cachan, France
{barbot,haddad,picaronny}@lsv.ens-cachan.fr

Monika Heiner
Brandenburg University of Technology,
Walther-Pauer-Strasse 2, Cottbus, Germany
monika.heiner@b-tu.de

***Abstract*—Formal models have been shown useful for analysis of regulatory systems. Here we focus on signalling cascades, a recurrent pattern of biological regulatory systems. We choose the formalism of stochastic Petri nets for this modelling and we express the properties of interest by formulas of a temporal logic. Such properties can be evaluated with either numeric or simulation based methods. The former one suffers from the combinatorial state space explosion problem, while the latter suffers from time explosion due to rare event phenomena. In this paper, we demonstrate the use of rare event techniques to tackle the analysis of signalling cascades. We compare the effectiveness of the COSMOS statistical model checker, which implements importance sampling methods to speed up rare event simulations, with the numerical model checker MARCIE on several properties. More precisely, we study three properties that characterise the ordering of events in the signalling cascade. We establish an interesting dependency between quantitative parameters of the regulatory system and its transient behaviour. Summarising, our experiments establish that simulation is the only appropriate method when parameters values increase and that importance sampling is effective when dealing with rare events.**

Keywords–rare event problem; importance sampling; regulatory biological systems; stochastic Petri nets.

I. INTRODUCTION

Signalling cascades. Signalling processes play a crucial role for the regulatory behaviour of living cells. They mediate input signals, i.e., the extracellular stimuli received at the cell membrane, to the cell nucleus, where they enter as output signals the gene regulatory system. Understanding signalling processes is still a challenge in cell biology. To approach this research area, biologists design and explore signalling networks, which are likely to be building blocks of the signalling networks of living cells. Among them are the type of signalling cascades which we investigate in our paper. In particular, we complete the analysis performed in [1].

A signalling cascade is a set of reactions that can be grouped into levels. At each level a particular enzyme is produced (e.g., by phosphorylation); the level generally also includes the inverse reactions (e.g., dephosphorylation). The system constitutes a cascade since the enzyme produced at some level is the catalyser for the reactions at the next level. The catalyser of the first level is usually considered to be the input signal, while the catalyser produced by the last level constitutes the output signal. The transient behaviour of such a system presents a characteristic shape, the quantity of every enzyme increases to some stationary value. In addition, the increases are temporally ordered w.r.t. the levels in the signalling cascade. This behaviour can be viewed as a signal travelling along the levels, and there are many interesting properties to be studied like the travelling time of the signal, the relation between the variation of the enzymes of two consecutive levels, etc.

In [2], it has been shown how such a system can be modelled by a Petri net, which can either be equipped with continuous transition firing rates leading to a continuous Petri net that determines a set of differential equations or by stochastic transition firing rates leading to a stochastic Petri net. This approach emphasises the importance of Petri nets that, depending on the chosen semantics, permit to investigate particular properties of the system. In this paper, we wish to explore the influence of stochastic features on the signalling behaviour, and thus we focus on the use of stochastic Petri nets.

Analysis of stochastic Petri nets can be performed either numerically or statistically. The former approach is much faster than the latter and provides exact results up to numerical approximations, but its application is limited by the memory requirements due to the combinatory explosion of the state space.

Statistical evaluation of rare events. Statistical analysis means to estimate the results by evaluating a sufficient number of simulations. However, standard simulation is unable to efficiently handle *rare events*, i.e., properties whose probability of satisfaction is tiny. Indeed, the number of trajectories to be generated in order to get an accurate interval confidence for rare events becomes prohibitively huge. Thus, *acceleration* techniques [3] have been designed to tackle this problem whose principles consist in (1) favouring trajectories that satisfy the property, and (2) numerically adjusting the result to take into account the bias that has been introduced. This can be done by *splitting* the most promising trajectories [4] or *importance sampling* [5], i.e., modifying the distribution during the simulation. In previous work [6], some of us have developed an original importance sampling method based on the design and numerical analysis of a reduced model in order to get the importance coefficients. First proposed for checking "unbounded until" properties (e.g., a quantity of enzymes remains below some threshold until a signal is produced) over models whose semantics is a discrete time Markov chain, it has been extended to also handle "bounded until" properties (e.g., a quantity of enzymes remains below some threshold until a signal is produced within 10 time units) and continuous time Markov chains [7].

Our contribution. In this paper, we complete the analysis of the signalling cascade performed in [1] with a new family of properties and we detail the algorithmic features of our importance sampling method. So, we consider here three families of properties for signalling cascades that are particularly relevant for the study of their behaviour and that are (depending on a

scaling parameter) potentially rare events. From an algorithmic point of view, this case study raises interesting issues since the combinatorial explosion of the model quickly forbids the use of numerical solvers and its intricate (quantitative) behaviour requires elaborated and different abstractions depending on the property to be checked.

Due to these technical difficulties, the signalling cascade analysis has led us to substantially improve our method and in particular the way we obtain the final confidence interval. From a biological point of view, experiments have pointed out interesting dependencies between the scaling parameter of the model and the probability of satisfying a property.

Organisation. In Section II, we present the biological background, the signalling cascade under study and the properties to be studied. Then, in Section III, after some recalls on stochastic Petri nets, we model signalling cascades by SPNs. We introduce the rare event issue and the importance sampling technique to cope with in Section IV. In Section V, we develop our method for handling rare events. Then, in Section VI, we report and discuss the results of our experiments. Finally, in Section VII, we conclude and give some perspectives to our work.

II. SIGNALLING CASCADES

In technical terms, signalling cascades can be understood as networks of biochemical reactions transforming input signals into output signals. In this way, signalling processes determine crucial decisions a cell has to make during its development, such as cell division, differentiation, or death. Malfunction of these networks may potentially lead to devastating consequences on the organism, such as outbreak of diseases or immunological abnormalities. Therefore, cell biology tries to increase our understanding of how signalling cascades are structured and how they operate. However, signalling networks are generally hard to observe and often highly interconnected, and thus signalling processes are not easy to follow. For this reason, typical building blocks are designed instead, which are able to reproduce observed input/output behaviours.

The case study we have chosen for our paper is such a signalling building block: the mitogen-activated protein kinase (MAPK) cascade [8]. This is the core of the ubiquitous ERK/MAPK network that can, among others, convey cell division and differentiation signals from the cell membrane to the nucleus. The description starts at the RasGTP complex, which acts as an enzyme (kinase) to phosphorylate Raf, which phosphorylates MAPK/ERK Kinase (MEK), which in turn phosphorylates Extracellular signal Regulated Kinase (ERK). We consider RasGTP as the input signal and ERKPP (activated ERK) as the output signal. This cascade (RasGTP $\rightarrow$ Raf $\rightarrow$ MEK $\rightarrow$ ERK) of protein interactions is known to control cell differentiation, while the strength of the effect depends on the ERK activity, i.e., concentration of ERKPP.

The scheme in Figure 1 describes the typical modular structure for such a signalling cascade, see [9]. Each layer corresponds to a distinct protein species. The protein Raf in the first layer is only singly phosphorylated. The proteins in the two other layers, MEK and ERK, respectively, can be singly as well as doubly phosphorylated. In each layer, forward reactions are catalysed by kinases and reverse reactions by phosphatases (Phosphatase1, Phosphatase2, Phosphatase3). The kinases in the MEK and ERK layers are the phosphorylated forms of the proteins in the previous layer. Each phosphorylation/dephosphorylation step applies mass action kinetics according to the pattern $A + E \rightleftharpoons AE \rightarrow B + E$. This pattern reflects the mechanism by which enzymes act: first building a complex with the substrate, which modifies the substrate to allow for forming the product, and then disassociating the complex to release the product; for details see [10].

Figure 2 depicts the evolution of the mean number of proteins with time. At time zero there is only a hundred RasGTP proteins. Then we observe the transmission of the signal witnessed by the decreasing of the number of RasGTP proteins successively followed by the increasing of the number of RafP, MEKPP and ERKPP proteins. In this figure, the temporal order between the increasing of the different types of proteins is clear. However, this figure only reports the mean number of each protein for a large number of simulations (300,000). It remains to check this correlation at the level of a single (random) trajectory.

Having the wiring diagram of the signalling cascade, a couple of interesting questions arise whose answers would shed some additional light on the subject under investigation. Among them are an assessment of the signal strength in each level, and specifically of the output signal. We will consider these properties in Sections VI-A and VI-B. The general scheme of the signalling cascade also suggests a temporal order of the signal propagation in accordance with the level order. What cannot be derived from the structure is the extent to which the signals are simultaneously produced; we will discuss this property in Section VI-C.

III. PETRI NET MODELLING

a) Stochastic Petri nets: Due to their graphical representation and bipartite nature, Petri nets are highly appropriate to model biochemical networks. When equipped with a stochastic semantics, yielding stochastic Petri nets (SPN) [11], they can be used to perform quantitative analysis.

Definition 1 (SPN): A stochastic Petri net $\mathcal{N}$ is defined by:

- a finite set of places P;
- a finite set of transitions T;
- a backward (resp. forward) incidence matrix **Pre** (resp. **Post**) from $P \times T$ to $\mathbb{N}$;
- a set of state-dependent rates of transitions $\{\mu_t\}_{t \in T}$ such that μ_t is a mapping from $\mathbb{N}^P$ to $\mathbb{R}_{>0}$.

A *marking* m of an SPN $\mathcal{N}$ is an item of $\mathbb{N}^P$. A transition t is fireable in marking m if for all $p \in P$ $m(p) \geq \mathbf{Pre}(p,t)$. Its firing leads to marking m' defined by: for all $p \in P$ $m'(p) = m(p) - \mathbf{Pre}(p,t) + \mathbf{Post}(p,t)$. It is denoted either as $m \xrightarrow{t} m'$ or as $m \xrightarrow{t}$ omitting the next marking. Let $\sigma = \sigma_1 \dots \sigma_n \in T^*$, then σ is fireable from m and leads to m' if there exists a sequence of markings $m = m_0, m_1, \dots, m_n$ such that for all $0 \leq k < n$, $m_k \xrightarrow{\sigma_k} m_{k+1}$. This firing is also denoted $m \xrightarrow{\sigma} m'$. Let m_0 be an initial marking, the reachability set $Reach(\mathcal{N}, m_0)$ is defined by: $Reach(\mathcal{N}, m_0) = \{m \mid \exists \sigma \in T^* \; m_0 \xrightarrow{\sigma} m\}$. The initialised SPNs $(\mathcal{N}, m_0)$ that we consider do not have deadlocks: for all $m \in Reach(\mathcal{N}, m_0)$ there exists $t \in T$ such that $m \xrightarrow{t}$.

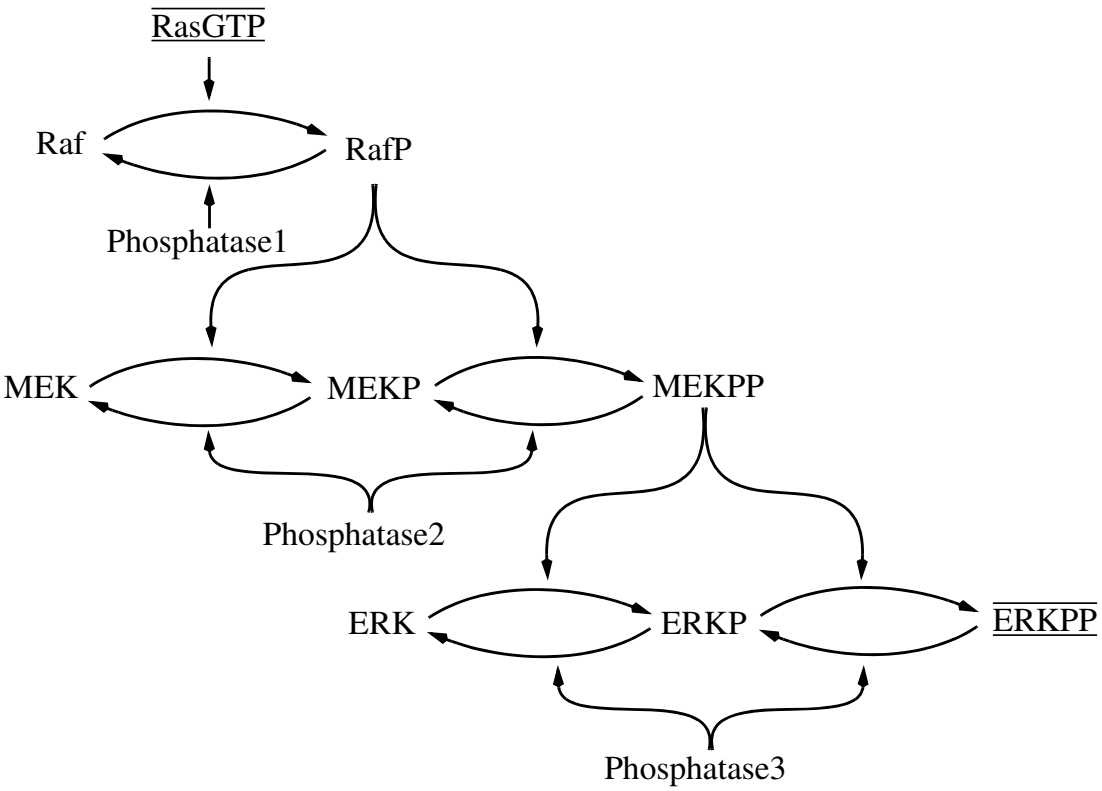


Figure 1. The general scheme of the considered three-level signalling cascade; RasGTP serves as input signal and ERKPP as output signal.

An SPN is a high-level model whose operational semantics is a continuous time Markov chain (CTMC). In a marking m, each enabled transition of the Petri net randomly selects an execution time according to a Poisson process with rate μ_t. Then the transition with earliest firing time is selected to fire yielding the new marking. This can be formalized as follows.

Definition 2 (CTMC of a SPN): Let $\mathcal{N}$ be a stochastic Petri net and m_0 be an initial marking. Then the CTMC associated with $(\mathcal{N}, m_0)$ is defined by:

- the set of states is $Reach(\mathcal{N}, m_0)$;
- the transition matrix $\mathbf{P}$ is defined by:

$$\mathbf{P}(m, m') = \frac{\sum_{m \xrightarrow{t} m'} \mu_t(m)}{\sum_{m \xrightarrow{t}} \mu_t(m)}$$

- the rate λ_m is defined by: $\lambda_m = \sum_{m \xrightarrow{t}} \mu_t(m)$

b) Running case study: We now explain how to model our running case study in the Petri net framework. The signalling cascade is made of several phosphorylation/dephosphorylation steps, which are built on mass/action kinetics. Each step follows the pattern $A+E \rightleftharpoons AE \rightarrow B+E$ and is modelled by a small Petri net component depicted in Figure 3. The mass action kinetics is expressed by the rate of the transitions. The marking-dependent rate of each transition is equal to the product of the number of tokens in all its incoming places up to a multiplicative constant given by the biological behaviour (summing up dependencies on temperature, pressure, volume, etc.).

The whole reaction network based on the general scheme of a three-level double phosphorylation cascade, as given in Figure 1, is modelled by the Petri net in Figure 4. The input signal is the number of tokens in the place RasGTP, and the output signal is the number of tokens in the place ERKPP.

This signalling cascade model represents a self-contained and closed system. It is covered with place invariants (see section VI), specifically each layer in the cascade forms a P-invariant consisting of all states a protein can undergo; thus the model is bounded. Assuming an appropriate initial marking, the model is also live and reversible; see [2] for more details, where this Petri net has been developed and analysed in the qualitative, stochastic and continuous modelling paradigms. In our paper, we extend these analysis techniques for handling properties corresponding to rare events.

We introduce a scaling factor N to parameterize how many tokens are spent to specify the initial marking. Increasing the scaling parameter can be interpreted in two different ways: either an increase of the biomass circulating in the closed system (if the biomass value of one token is kept constant), or an increase of the resolution (if the biomass value of one token inversely decreases, called level concept in [2]). The kind of interpretation does not influence the approach we pursue in this paper.

Increasing N means to increase the size of the state space and thus of the CTMC, as shown in Table I, which has been computed with the symbolic analysis tool MARCIE [12]. As expected, the explosion of the state space prevents numerical model checking for higher N and thus calls for statistical model checking.

Furthermore, increasing the number of states means to actually decrease the probabilities to be in a certain state,

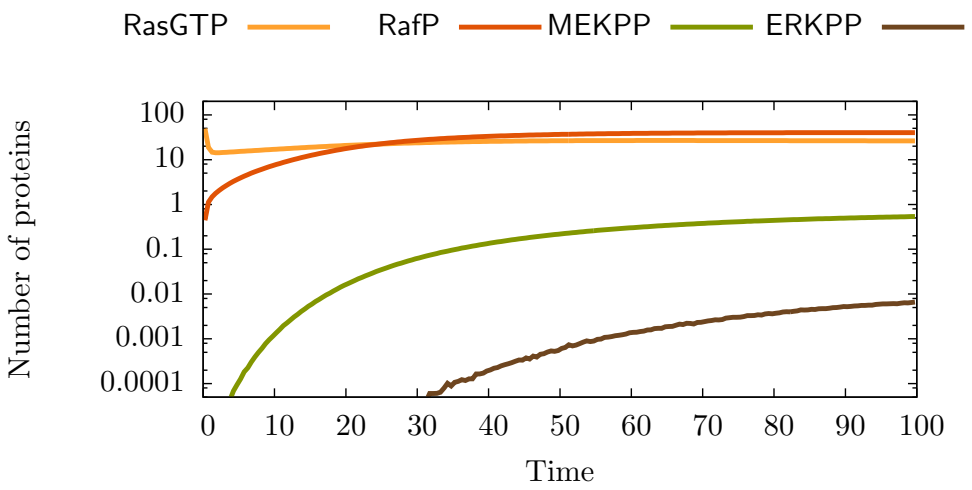


Figure 2. Transmission of the signal in the signalling cascade.

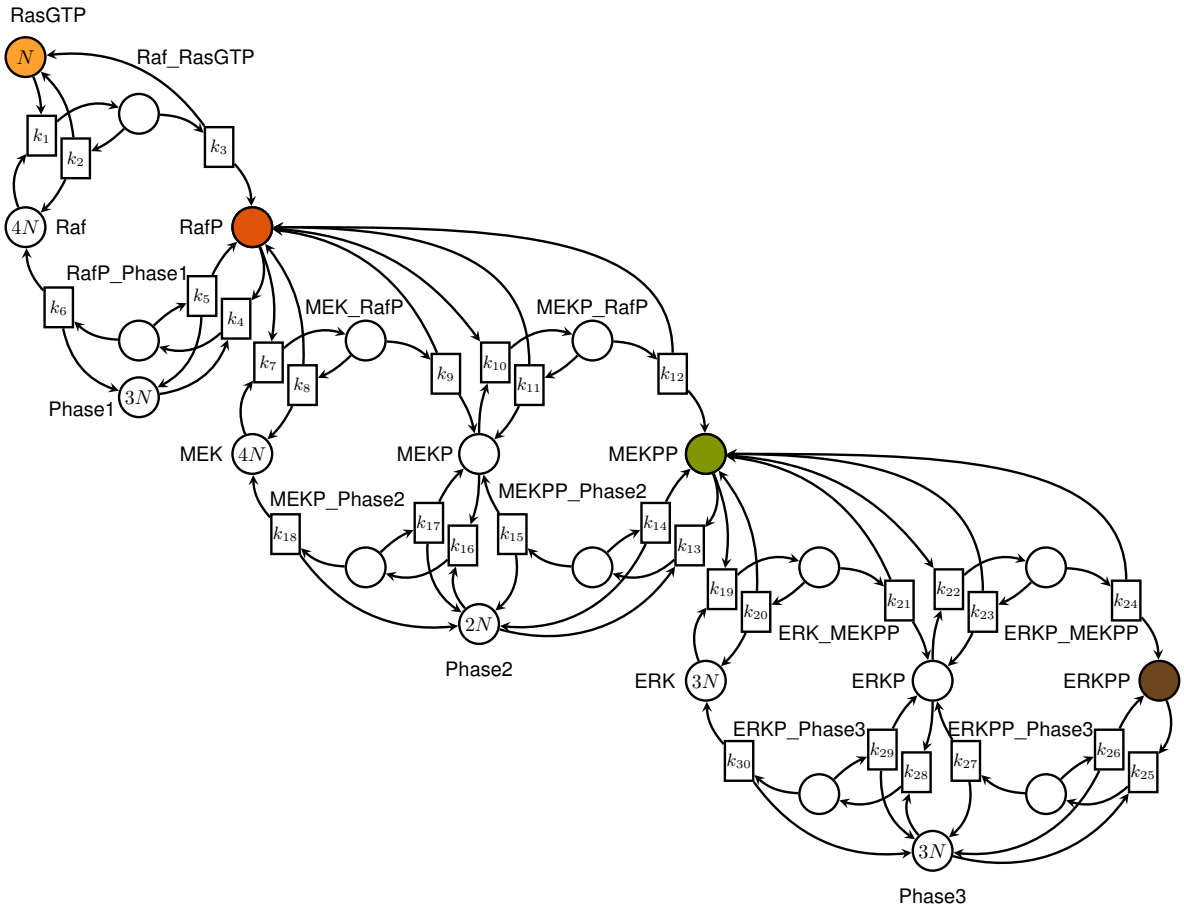


Figure 4. A Petri net modelling the three-level signalling cascade given in Figure 1; k_i are the kinetic constants for mass action kinetics, N the scaling parameter.

as the total probability of 1 is fixed. With the distribution of the probability mass of 1 over an increasingly huge number of states, we obtain sooner or later states with very tiny probabilities, and thus rare events. Neglecting rare events is usually appropriate when focusing on the averaged behaviour. But they become crucial when certain jump processes such as mutations under rarely occurring conditions are of interest.

IV. Statistical Model Checking with Rare Events

A. Statistical model checking and rare events

c) Simulation recalls: The statistical approach for evaluating the expectation $\mathbf{E}(X)$ of a random variable X related to

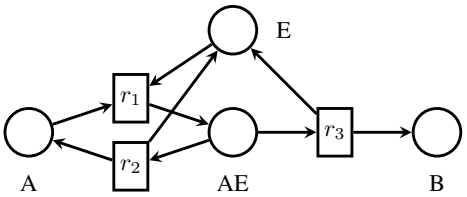


Figure 3. Petri net pattern for mass action kinetics $A + E \rightleftharpoons AE \rightarrow B + E$.

Table I. Development of the state space for increasing N.

N	number of states	N	number of states
1	24,065 (4)	6	769,371,342,640 (11)
2	6,110,643 (6)	7	5,084,605,436,988 (12)
3	315,647,600 (8)	8	27,124,071,792,125 (13)
4	6,920,337,880 (9)	9	122,063,174,018,865 (14)
5	88,125,763,956 (10)	10	478,293,389,221,095 (14)

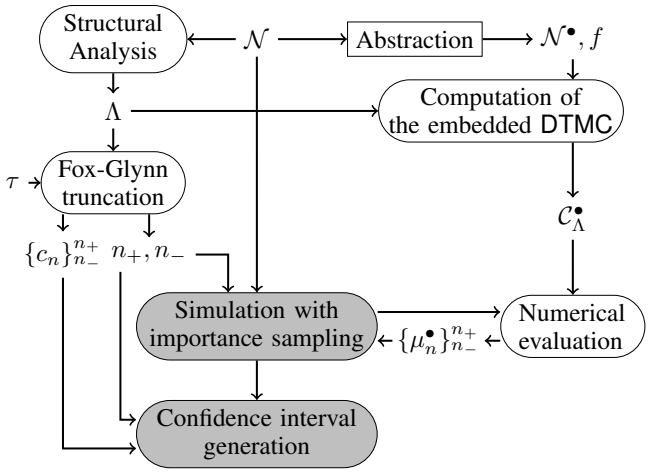


Figure 5. Principles of the methodology

a random path in a Markov chain is generally based on three parameters: the number of simulations K, the confidence level γ, and the width of the confidence interval lg (see [13]). Once the user provides two parameters, the procedure computes the remaining one. Then it performs K simulations of the Markov chain and outputs a confidence interval $[L, U]$ with a width of at most lg such that $\mathbf{E}(X)$ belongs to this interval with a probability of at least γ. More precisely, depending on the hypotheses, the confidence level has two interpretations: (1) either the confidence level is ensured, or (2) is only asymptotically valid (when K goes to infinity using central limit theorem). The two usual hypotheses for providing an exact confidence level rather than an asymptotical one are: (1) the distribution of X is known up to a parameter (e.g., Bernoulli law with unknown success probability), or (2) the random variable is bounded allowing to exploit Chernoff-Hoeffding bounds [14].

d) Statistical evaluation of a reachability probability: Let $\mathcal{C}$ be a discrete time Markov chain (DTMC) with two absorbing states s_+ or s_-, such that the probability to reach s_+ or s_- from any state is equal to 1. Assume one wants to estimate p, the probability to reach s_+. Then the simulation step consists in generating K paths of $\mathcal{C}$, which end in an absorbing state. Let K_+ be the number of paths ending in state s_+. The random variable K_+ follows a binomial distribution with parameters p and K. Thus, the random variable $\frac{K_+}{K}$ has a mean value p and since the distribution is parametrised by p, a confidence level can be ensured. Unfortunately, when $p \ll 1$, the number of paths required for a small confidence interval is too large to be simulated. This issue is known as the *rare event* problem.

e) Importance sampling: In order to tackle the rare event problem, the importance sampling method relies on a choice of a biased distribution that will artificially increase the frequency of the observed rare event during the simulation. The choice of this distribution is crucial for the efficiency of the method and usually cannot be found without a deep understanding of the system to be studied. The generation of paths is done according to a modified DTMC $\mathcal{C}'$, with the same state space, but modified transition matrix $\mathbf{P}'$. $\mathbf{P}'$ must satisfy:

$$\mathbf{P}(s, s') > 0 \Rightarrow \mathbf{P}'(s, s') > 0 \vee s' = s_- \qquad (1)$$

which means that this modification cannot remove transitions that have not s_- as target, but can add new transitions. The method maintains a correction factor called L initialised to 1; this factor represents the *likelihood* of the path. When a path crosses a transition $s \rightarrow s'$ with $s' \neq s_-$, L is updated by $L \leftarrow L\frac{\mathbf{P}(s,s')}{\mathbf{P}'(s,s')}$. When a path reaches s_-, L is set to zero. If $\mathbf{P}' = \mathbf{P}$ (i.e., no modification of the chain), the value of L when the path reaches s_+ (resp. s_-) is 1 (resp. 0). Let V_s (resp. W_s) be the random variable associated with the final value of L for a path starting in x in the original model $\mathcal{C}$ (resp. in $\mathcal{C}'$). By definition, the expectation $\mathbf{E}(V_{s_0}) = p$ and by construction of the likelihood, $\mathbf{E}(W_{s_0}) = p$. Of course, a useful importance sampling should reduce the variance of W_{s_0} w.r.t. to the one of V_{s_0} equal to $p(1-p) \approx p$ for a rare event.

V. Our methodology for importance sampling

A. Previous work

In [6], [7], we provided a method to compute a biased distribution for importance sampling: we manually design an abstract smaller model, with a behaviour close to the one of the original model, that we call the *reduced model* and perform numerical computations on this smaller model to obtain the biased distribution. Furthermore, when the correspondence of states between the original model and the reduced one satisfies a good property called the *variance reduction guarantee*, W_{s_0} is a binary random variable (i.e., a rescaled Bernoulli variable) thus allowing to get an exact confidence interval with reduced size. We applied this method in order to tackle the estimation of time bounded property in CTMCs when it is a rare event, that is the probability to satisfy a formula $aU^{[0,\tau]}b$: the state property a is fulfilled until an instant in $[0, \tau]$ such that the state property b is fulfilled. Let us outline the different steps of the method that is depicted in Figure 5.

Abstraction of the model. As discussed above, given a SPN $\mathcal{N}$ modelling the system to be studied, we manually design an appropriate reduced one $\mathcal{N}^\bullet$ and a correspondence function f from states of $\mathcal{N}$ to states of $\mathcal{N}^\bullet$. Function f is defined at the net level (see Section VI).

Structural analysis. Importance sampling was originally proposed for DTMCs. In order to apply it for CTMC $\mathcal{C}$ associated

with net $\mathcal{N}$, we need to uniformize $\mathcal{C}$ (and also $\mathcal{C}^\bullet$ associated with $\mathcal{N}^\bullet$), which means finding a bound Λ for exit rate of states, i.e., markings, considering Λ as the uniform exit rate of states and rescaling accordingly the transition probability matrices [15]. Since the rates of transitions depend on the current marking, determining Λ requires a structural analysis like invariant computations for bounding the number of tokens in places.

Fox-Glynn truncation. Given a uniform chain with initial state s_0, exit rate Λ, and transition probability matrix $\mathbf{P}$, the state distribution π_τ at time τ is obtained by the following formula:

$$\pi_\tau(s) = \sum_{n\geq 0} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!}\mathbf{P}^n(s_0, s).$$

This value can be estimated, with sufficient precision, by applying [16]. Given two numerical accuracy requirements α and β, truncation points n^- and n^+ and values $\{c_n\}_{n^-\leq n\leq n^+}$ are determined such that for all $n^- \leq n \leq n^+$:

$$c_n(1-\alpha-\beta) \leq \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq c_n$$

and

$$\sum_{n<n^-} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq \alpha \sum_{n>n^+} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq \beta$$

Computation of the embedded DTMC. Since $\mathcal{N}^\bullet$ has been designed to be manageable, we build the embedded DTMC $\mathcal{C}^\bullet_\Lambda$ of $\mathcal{N}^\bullet$ after uniformization. More precisely, since we want to evaluate the probability to satisfy formula $aU^{[0,\tau]}b$, the states satisfying a (resp. $\neg a \wedge \neg b$) are aggregated into an absorbing accepting (resp. rejecting) state. Thus, the considered probability $\mu_\tau(s^\bullet)$ is the probability to be in the accepting state at time τ starting from state $s^\bullet$.

Numerical evaluation. Matrix $\mathbf{P}'$ used for importance sampling simulation in the embedded DTMC of $\mathcal{N}$ to evaluate formulas $aU^{[0,n]}b$ for $n^- \leq n \leq n^+$, is based on the distributions $\{\mu^\bullet_n\}_{0<n\leq n^+}$, where $\mu^\bullet_n(s^\bullet)$ is the probability that a random path of the embedded DTMC of $\mathcal{N}^\bullet$ starting from $s^\bullet$ fulfills $aU^{[0,n]}b$. Such a distribution is computed by a standard numerical evaluation. However, since n^+ can be large, depending on the memory requirements, this computation can be done statically for all n or dynamically for a subset of such n during the importance sampling simulation.

Simulation with importance sampling. This is done as for a standard simulation except that the random distribution of the successors of a state depend on both the embedded DTMC $\mathcal{C}_\Lambda$ and the values computed by the numerical evaluation. Moreover, all formulas $aU^{[0,n]}b$ for $n^- \leq n \leq n^+$ have to be evaluated increasing the time complexity of the method w.r.t. the evaluation of an unbounded timed until formula.

Generation of the confidence interval. The result of the simulations is a family of confidence intervals indexed by $n^- \leq n \leq n^+$. Using the Fox-Glynn truncation, we weight and combine the confidence intervals in order to return the final interval.

Algorithmic considerations. The importance sampling simulation needs the family of vectors $\{\mu^\bullet_n\}_{0<n\leq n^+}$. They can be computed iteratively one from the other with overall time complexity $\Theta(mn^+)$ where m is the number of states of $\mathcal{N}^\bullet$. More precisely, given $\mathbf{P}^\bullet$ the transition matrix of $\mathcal{C}^\bullet_\Lambda$ (taking into account the transformation corresponding to the two absorbing states with $s^\bullet_+$ the accepting one):

$$\forall s^\bullet \neq s^\bullet_+ \ \mu^\bullet_0(s^\bullet) = 0,\ \mu^\bullet_0(s^\bullet_+) = 1 \ \text{ and } \ \mu^\bullet_n = \mathbf{P}^\bullet \cdot \mu^\bullet_{n-1}$$

Algorithm 1. One can perform this computation before starting the importance sampling simulation. But for large values of n^+, the space complexity to store them becomes intractable. However, looking more carefully at the importance sampling specification, it appears that at simulation time n one only needs two vectors $\{\mu^\bullet_n\}$ and $\{\mu^\bullet_{n-1}\}$ [7]. So depending on the memory requirements, we propose three alternative methods.

Algorithm 2. Let $l(< n^+)$ be an integer. In the precomputation stage, the second method only stores the $\lfloor \frac{n^+}{l} \rfloor + 1$ vectors $\mu^\bullet_n$ with n multiple of l in list Ls and $\mu^\bullet_{l\lfloor \frac{n^+}{l} \rfloor+1}, \ldots, \mu^\bullet_{n^+}$ in list K (see the precomputation stage of the algorithm). During the simulation stage, at time n, with $n = ml$, the vector $\mu^\bullet_{n-1}$ is present neither in Ls nor in K. So, the method uses the vector $\mu^\bullet_{l(m-1)}$ stored in Ls to compute iteratively all vectors $\mu^\bullet_{l(m-1)+i} = P^{\bullet i} \cdot \mu^\bullet_{l(m-1)}$ for i from 1 to $l-1$ and store them in K (see the computation stage of the algorithm). Then it proceeds to l consecutive steps of simulation without anymore computations. We choose l close to $\sqrt{n^+}$ in order to minimize the space complexity of such a factorization of steps.

Algorithm 3. Let $k = \lfloor \log_2(n^+) \rfloor + 1$. In the precomputation stage, the third method only stores $k+1$ vectors in Ls. More precisely, initially using the binary decomposition of n^+ ($n^+ = \sum_{i=0}^k a_{n^+,i}2^i$), the list Ls of $k+1$ vectors consists of $w_{i,n} = \mu^\bullet_{\sum_{j=i}^k a_{n,j}2^j}$, for all $1 \leq i \leq k+1$ (see the precomputation step of the algorithm). During the simulation stage at time n, with the binary decomposition of n ($v = \sum_{i=0}^k a_{n,i}2^i$), the list Ls consists of $w_{i,n} = \mu^\bullet_{\sum_{j=i}^k a_{n,j}2^j}$, for all $1 \leq i \leq k+1$. Observe that the first vector $w_{1,n}$ is equal to $\mu^\bullet_n$. We obtain $\mu^\bullet_{n-1}$ by updating Ls according to $n-1$. Let us describe the updating of the list performed by the stepcomputation of the algorithm. Let i_0 be the smallest index such that $a_{n,i_0} = 1$. Then for $i > i_0$, $a_{n-1,i} = a_{n,i}$, $a_{n-1,i_0} = 0$ and for $i < i_0$, $a_{n-1,i} = 1$. The new list Ls is then obtained as follows. For $i > i_0$ $w_{i,n-1} = w_{i,n}$, $w_{i_0,n-1} = w_{i_0-1,n}$. Then the vectors for $i_0 < i$, the vectors $w_{i,n-1}$ are stored along iterated $2^{i_0-1}-1$ matrix-vector products starting from vector $w_{i_0,n-1}$: $w(j, v-1) = P^{\bullet 2^j} w(j+1, n-1)$.

The computation at time n requires $1 + 2 + \cdots + 2^{i_0-1}$ products matrix-vector, i.e., $\Theta(m2^{i_0})$. Noting that the bit i is reset at most $m2^{-i}$ times, the complexity of the whole computation is $\sum_{i=1}^k 2^{k-i}\Theta(m2^i) = \Theta(mn^+\log(n^+))$.

Algorithm 4. The fourth method consists in computing vector $\mu^\bullet_v$ from the initial vector at each step. In this method, we only need to store two copies of the vector.

Algorithm 1

```
Precomputation(n+, μ•0, P•) Result: Ls
// List Ls fulfills Ls(i) = μ•i
Ls(0) ← μ•0
for i = 1 to n+ do
  Ls(i) ← P• Ls(i − 1)
```

Algorithm 2

```
Precomputation(n+, μ•0, P•) Result: Ls, K
// List Ls fulfills Ls(i) = μ•i·l
l ← ⌊√n+⌋   w ← μ•0
for i from 1 to ⌊n+/l⌋ l do
  w ← P• w  if i mod l = 0 then
    Ls(i/l) ← w
// List K contains μ•⌊n+/l⌋l+1, ..., μ•n+
for i from ⌊n+/l⌋ l + 1 to n+ do
  w ← P• w  K(i mod l) ← w

Stepcomputation(n, l, P•, K, Ls)   // Updates K when needed
if n mod l = 0 then
  w ← Ls(n/l − 1)
  for i from (n/l − 1)l + 1 to n − 1 do
    w ← P•0 w  K(i mod l) ← w
```

Algorithm 3

```
Precomputation(n+, μ•0, P•) Result: Ls
// Ls fulfills Ls(i) = μ•Σ_{j=i}^{k} a_{n+,j} 2^j
k ← ⌊log2(n+)⌋ + 1   v ← μ•0   Ls(k + 1) ← v
for i from k downto 0 do
  if a_{n+,i} = 1 then
    for j from 1 to 2^i do
      w ← P• w
  Ls(i) ← w

Stepcomputation(n, l, P•, Ls)   // Ls is updated accordingly to n − 1
i0 ← min(i | a_{n,i} = 1)   w ← Ls(i0 + 1)   Ls(i0) ← n
for i from i0 − 1 downto 0 do
  for j = 1 to 2^i do
    w ← P• w
  Ls(i) ← w
```

Algorithm 4

```
Stepcomputation(n, μ•0, P•) Result: v
// Vector v equal to μ•n
v ← μ•0
for i = 1 to n do
  v′ ← P• v  v ← v′
```

B. Tackling signalling cascades

The reduced net that we design for signalling cascades does not satisfy the variance reduction guarantee. This has two consequences: (1) we can perform a much more efficient importance sampling simulation and (2) we need to propose different ways of computing "approximate" confidence intervals. We now detail these issues.

Table II. Compared complexities.

Complexity	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
Space	mn^+	$2m\sqrt{n^+}$	$m \log n^+$	$2m$
Time for the precomputation	$\Theta(mn^+)$	$\Theta(mn^+)$	$\Theta(mn^+)$	0
Additional time for the simulation	0	$\Theta(mn^+)$	$\Theta(mn^+ \log(n^+))$	$\Theta(m(n^+)^2)$

Importance sampling for multiple formulas. Using uniformisation, the computation of the probability to satisfy $aU^{\tau}b$ in the CTMC, is performed by the computation of the probability to satisfy $aU^{[0,n]}b$ for all n between n^- and n^+ in the embedded DTMC. A naive implementation would require to apply statistical model checking of formulas $aU^{[0,n]}b$ for all n, but such a number can be large. A more tricky alternative consists in producing all trajectories for time horizon $n = n^+$ with the corresponding importance sampling. Simulation results are updated at the end of a trajectory for all the intervals $[0, n]$ with $n^- \leq n \leq n^+$ as follows. If the trajectory has reached the absorbing rejecting state s^- then it is an unsuccessful trajectory for all intervals. Otherwise, if it has reached the absorbing accepting state s^+ at time n_0, then for all $n \geq n_0$ it is a successful trajectory and for all $n < n_0$ it is unsuccessful. Doing this way, every trajectory contributes to all evaluations, and we significantly increase the sample size without increasing computational cost. With the same number of simulations the accuracy of the result is greatly improved. For example, the estimation of the first property (with $N = 5$) of the signalling cascade leads to $n^+ - n^- = 759$, inducing a reduction of the simulation time by three orders of magnitude. However, this requires that the importance sampling associated with time interval $[0, n^+]$ is also appropriate for the other intervals and in particular with time interval $[0, n^-]$. It is not true when the reachability probability in n^- and n^+ steps differ by several orders of magnitude. In this case, the interval $[n^-, n^+]$ must be split into several intervals such that, the reachability probability for each trajectory inside an interval is of the same order of magnitude. Figure 6 illustrates this idea by splitting the interval $[n^-, n^+]$ in subintervals of width l. For each subinterval, k trajectories are simulated. The naive algorithm corresponds to $l = 1$. In the case of our experiments (see Section VI) $l = n^+ - n^-$ was sufficient to obtain accurate results.

Confidence interval estimation. The result of each trajectory of the simulation is a realisation of the random variable $W_{s_0} = X_{s_0} L_{s_0}$ where the binary variable X_{s_0} indicates whether a trajectory starting from s_0 is succesful and the positive random variable L_{s_0} is the (random) likelihood. Observe that $\mathbf{E}(W_{s_0}) = \mathbf{E}(L_{s_0} | X_{s_0} = 1)\mathbf{E}(X_{s_0})$. Since X_{s_0} follows a Bernoulli distribution, an exact confidence interval can be produced for $\mathbf{E}(X_{s_0})$. For $\mathbf{E}(L_{s_0} | X_{s_0} = 1)$ several approaches are possible among them we have selected three possible computations ranked by conservation degree.

1) The more classical way to compute confidence inter-

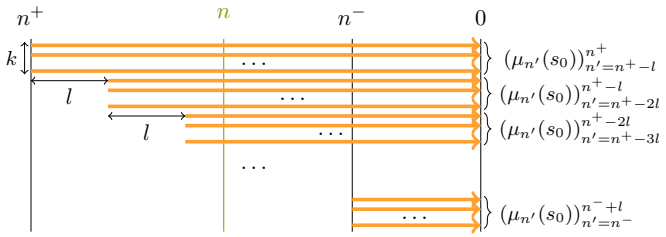


Figure 6. Parallel simulation estimating $(\mu_n(s_0))_{n=n^-}^{n^+}$ with reuse of trajectories.

vals is to suppose that the distribution is Gaussian; this is asymptotically valid if the variance is finite, thanks to the central limit theorem.

2) Another method is to use a pseudo Chernoff-Hoeffding bound. Whenever the random variable is bounded, this method is asymptotically valid. In our case we will use the minimal and maximal values observed during the simulation as the bounds of L_{s_0}.

3) The last method, which is more conservative than the previous one, consists in returning the minimal and maximal observed values as the confidence interval.

VI. Experiments

We have analysed three properties, the last two are inspired by [2]. Recall that the initial marking of the model is parametrized by a scaling factor N. For the first two properties, the reduced model is the same model but with local smaller scaling factors on the different layers of phosphorylation. Every state of the initial model is mapped (by f) to a state of the abstract model which has the "closest" proportion of chemical species. For instance, let $N = 4$, which corresponds to 16 species of the first layer, a state with 6 tokens in Raf and 10 tokens in RafP is mapped, for a reduced model with $N = 3$, to a state with $4 = \lfloor 6 \times 3/4 \rfloor$ tokens in Raf and $8 = \lceil 10 \times 3/4 \rceil$ tokens in RafP (see the later on for a specification of f).

All statistical experiments have been carried out with our tool Cosmos [17]. Cosmos is a statistical model checker for the HASL logic [18]. It takes as input a Petri net (or a high-level Petri net) with general distributions for transitions. It performs an efficient statistical evaluation of the stochastic Petri net by generating a code per model and formula. In the case of importance sampling, it additionally takes as inputs the reduced model and the mapping function specified by a C function and returns the different confidence intervals.

All experiments have been performed on a machine with 16 cores running at 2 GHz and 32 GB of memory both for the statistical evaluation of Cosmos and the numerical evaluation of Marcie.

A. Maximal peak of the output signal

The first property is expressed as a time-bounded reachability formula assessing the strength of the output signal of the last layer: "What is the probability to reach within 10 time units a state where the total mass of ERK is doubly phosphorylated?", associated with probability p_1 defined by:

$$p_1 = \Pr(\text{True } \mathbf{U}^{\leq 10}(\text{ERKPP} = 3N))$$

Table III. Computational complexity related to the evaluation of p_1.

N	Cosmos			Marcie	
	Reduction factor	time	memory	time	memory
1	-	-	-	4	514MB
2	38	20,072	3,811MB	326	801MB
3	558	15,745	15,408MB	43,440	13,776MB
4	4667	40,241	3,593MB	Out of Memory: >32GB	
5	27353	51,120	19,984MB		

Table IV. Numerical values associated with p_1.

N	Cosmos			Marcie
	Gaussian CI	Chernoff CI	MinMax CI	Output
1				2.07 E^{-12}
2	[3.75E^{-27},5.88E^{-26}]	[3.75E^{-27},4.54E^{-25}]	[3.75E^{-27},1.57E^{-23}]	8.18E^{-26}
3	[4.34E^{-42},1.72E^{-39}]	[4.34E^{-42},1.82E^{-38}]	[4.43E^{-42},1.87E^{-37}]	2.56E^{-39}
4	[1.54E^{-57},8.54E^{-56}]	[1.54E^{-57},1.98E^{-55}]	[1.78E^{-57},7.05E^{-55}]	-
5	[3.97E^{-73},2.33E^{-70}]	[3.97E^{-73},7.30E^{-70}]	[5.44E^{-73},2.24E^{-69}]	-

The inner formula is parametrized by N, the scaling factor of the net (via its initial marking). The reduced model that we design for Cosmos uses different scaling factors for the three layers in the signalling cascade. The first two layers of phosphorylation, which are based on Raf and MEK, always use a scaling factor of 1, whereas the last layer involving ERK uses a scaling factor of N. The second column of Table III shows the ratio between the number of reachable states of the original and the reduced models.

1) Experimental Results: We have performed experiments with both Cosmos and Marcie. The time and memory consumptions for increasing values of N are reported in Table III. For each value of N we generate one million trajectories with Cosmos. We observe that the time consumption significantly increases between $N = 3$ and $N = 4$. This is due to a change of strategy in the space/time trade-off in order to not exceed the machine memory capacity. Marcie suffers an exponential increase w.r.t. both time and space resources. When $N = 3$, it is slower than Cosmos and it is unable to handle the case $N = 4$.

Table IV depicts the values returned by the two tools: Marcie returns a single value, whereas Cosmos returns three confidence intervals (discussed above) with a confidence level set to 0.99. We observe that confidence intervals computed by the Gaussian analysis neither contain the result, the ones computed by Chernoff-Hoeffding do not contain it for $N = 3$, and the most conservative ones always contain it (when this result is available).

Figure 7 illustrates the dependency of p_1 with respect to the scaling factor N. It appears that the probability p_1 depends on N in an exponential way. The constants occurring in the formula could be interpreted by biologists.

2) Mapping function: We describe here formally the reduction function f. The reduction function must map each marking of the Petri net to a marking of the reduced Petri net.

First, we observe that the signalling cascades SPN contains three places invariants of interest:

- The total number of tokens in the set of places {Raf,Raf_RasGTP,RafP_Phase1,RafP, MEK_RafP, MEKP_RafP} is equal to $4N$.
- The number of tokens in the set of places {MEK, MEK_RafP, MEKP_Phase2, MEKP, MEKP_RafP,

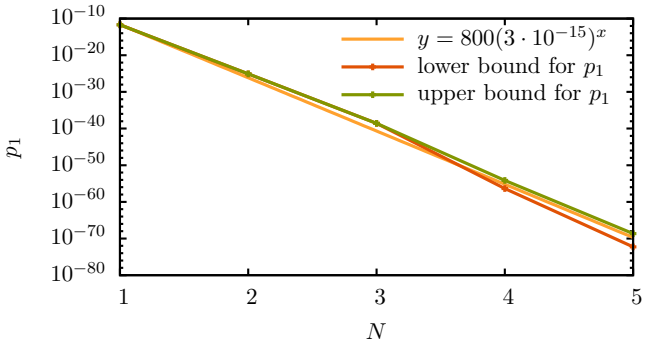


Figure 7. Highlighting an exponential dependency.

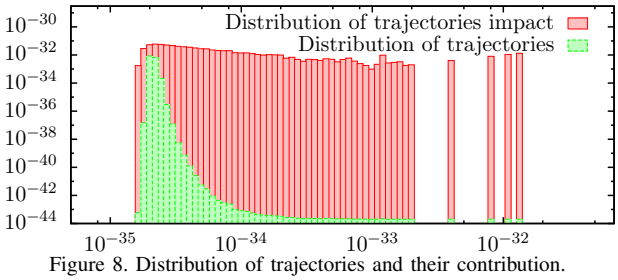


Figure 8. Distribution of trajectories and their contribution.

MEKPP_Phase2, MEKPP, ERK_MEKPP, ERKP_MEKPP} is equal to $2N$.

- The number of tokens in the set of places {ERK, ERK_RafP, ERKP_Phase2, ERKP, ERKP_RafP, ERKPP_Phase2, ERKPP} is equal to $3N$.

We also introduce three subsets of places, one per layer of phosphorylation.

- S_1 = {Raf,Raf_RasGTP,RafP_Phase1,RafP}
- S_2 = {MEK, MEK_RafP, MEKP_Phase2, MEKP, MEKP_RafP, MEKPP_Phase2,MEKPP}
- S_3 = {ERK, ERK_RafP, ERKP_Phase2, ERKP, ERKP_RafP, ERKPP_Phase2, ERKPP}

Let us remark that a marking of the SPN $\mathcal{N}$ is uniquely determined by its values on places in S_1, S_2 and S_3.

We define a function g such that: for all positive integer m, positive real number p and vector of integers of size k, $\mathbf{v} = (v_i)_1^k$, $g\,(p, m, \mathbf{v})$ is the vector of integers of size k, $\mathbf{u} = (u_i)_1^k$, defined by: for all $i > 1$,

$$u_i = \min\left(\lceil v_i \cdot p \rceil, m - \sum_{l=i+1}^{k} u_l\right) \text{ and } u_1 = m - \sum_{l=2}^{k} u_l$$

One can see that the g is properly defined and that the sum of the components of $\mathbf{u}$ are equal to m.

The reduction function f for the two properties is a mapping from the set of states of SPN $\mathcal{N}$ to the set of states of the reduced SPN $\mathcal{N}^\bullet$. This function takes as input the marking of a set of places that uniquely define the state. This set can be decomposed on the three layers of phosphorylation, that is S_1 for the first layer, S_2 for the second layer and S_3 for the last layer.

Recall that layers are not independent one from the others because proteins of one layer are used to activate the following layer; this can be seen on the invariant that contains places of the following layer. The mapping function that we construct preserve these invariants.

Roughly speaking, on each layer S_i, this function f applies a function of the form $g(p_i, m_i, -)$.

More precisely, given a scaling factor N and a scaling factor for each of the three layers of the reduced model, respectively N_1, N_2 and N_3, the reduction function f maps the marking m on the marking $m^\bullet$ defined as follow:

- $(m^\bullet(p))_{p\in S_3} = g\left(\frac{N_3}{N}, 3N_3, (m(p))_{p\in S_3}\right)$
- $(m^\bullet(p))_{p\in S_2} = g\big(\frac{N_2}{N}, 2N_2 - m^\bullet(\text{ERK\_MEKPP}) - m^\bullet(\text{ERKP\_MEKPP}), (m(p))_{p\in S_2}\big)$
- $(m^\bullet(p))_{p\in S_1} = g\big(\frac{N_1}{N}, 4N_1 - m^\bullet(\text{MEK\_RafP}) - m^\bullet(\text{MEKP\_RafP}), (m(p))_{p\in S_1}\big)$

One can see that the three invariants are preserved in the reduced model by f. We choose $N_1 = N_2 = 1$ and $N_3 = N$.

3) Experimental analysis of the likelihood: We describe here some technical details of the simulation done for evaluating probability p_1. Recall the likelihood of a trajectory requires the distribution of the random variable W_{s_0}. Proposition 6 of [6] ensures that W_{s_0} takes values in $\{0\} \cup [\mu^\bullet_{n^+}(f(s)), \infty[$. This was proven for DTMCs but can be adapted in a straightforward way for CTMCs. Values taken by L_{s_0} are taken by W_{s_0} when at the end of a successful trajectory, therefore these values are in $[\mu^\bullet_{n^+}(f(s)), \infty[$.

We simulate the system for the first formula with $N = 2$ and a discrete horizon of 615 (615 is the right truncation point given by Fox-Glynn algorithm). The result of the simulation is represented as an histogram shown in Figure 8. The total number of trajectories is 69000, 49001 of them are not successful. We observe that most of the successful trajectories end with a value close to 2.10^{-35}, and that a few trajectories have a value close to 10^{-32}. This is represented by an histogram which is shown as the green part of Figure 8 (with a logarithmic scale for the abscissa). We also represent the histogram of the contribution of the trajectories for the estimation of the mean value of L_{s_0}, that is the red part of the figure (with a logarithmic scale for the ordinate). We observe that the contribution to this mean value is almost uniform. Thus, a trajectory ending with a likelihood close to 10^{-32} have a larger impact than one ending with a likelihood close to 10^34. This means that an estimator of the mean value of $L_{(s_0,u)}$ will underestimate the expectation of $L_{(s_0,u)}$. To produce a framing of the result, one has to use a very conservative method to avoid underestimating the result.

B. Conditional maximal signal peak

The network structure of each layer in the signalling cascade presents a cyclic behaviour, i.e., phosphorylated proteins, serving as signal for the next layer, can also be dephosphorylated again, which corresponds to a decrease of the signal strength. Thus, an interesting property of the signalling cascade is the probability of a further increase of the signal strength under the condition that a certain strength has already been reached. We estimate this quantity for the first layer in the signalling cascade, i.e., RafP, and ask specifically for the probability to reach its maximal strength, $4N$: "What is the probability of the concentration of RafP to continue its

Table V. Numerical values associated with p_2.

N	L	COSMOS		MARCIE		
		confidence interval	time	result	time	memory
2	2	$[2.39{\cdot}10^{-13}, 1.07{\cdot}10^{-9}]$	31	$5.55{\cdot}10^{-10}$	90	802 MB
2	3	$[2.18{\cdot}10^{-10}, 6.92{\cdot}10^{-8}]$	110	$6.64{\cdot}10^{-8}$	136	816 MB
2	4	$[9.33{\cdot}10^{-8}, 3.54{\cdot}10^{-5}]$	256	$3.01{\cdot}10^{-6}$	276	798 MB
2	5	$[1.16{\cdot}10^{-5}, 6.08{\cdot}10^{-4}]$	1000	$7.16{\cdot}10^{-5}$	759	801 MB
2	6	$[5.42{\cdot}10^{-4}, 1.21{\cdot}10^{-3}]$	5612	$1.27{\cdot}10^{-3}$	3180	804 MB
3	5	$[1.82{\cdot}10^{-12}, 9.78{\cdot}10^{-9}]$	459	Time $>$ 48 hours		
3	6	$[3.41{\cdot}10^{-10}, 9.66{\cdot}10^{-8}]$	1428			
3	7	$[1.81{\cdot}10^{-8}, 2.23{\cdot}10^{-6}]$	7067			
3	8	$[8.72{\cdot}10^{-7}, 2.71{\cdot}10^{-6}]$	4460			
3	9	$[1.42{\cdot}10^{-6}, 4.59{\cdot}10^{-5}]$	4301			
3	10	$[2.69{\cdot}10^{-4}, 9.34{\cdot}10^{-4}]$	6420			
4	10	$[5.12{\cdot}10^{-9}, 2.75{\cdot}10^{-8}]$	8423	Memory $>$ 32GB		
4	11	$[8.23{\cdot}10^{-8}, 2.97{\cdot}10^{-7}]$	7157			
4	12	$[9.84{\cdot}10^{-7}, 1.86{\cdot}10^{-6}]$	18730			

increase and reach $4N$, when starting in a state where the concentration is for the first time at least L?". This is a special use case of the general pattern introduced in [2].

$$p_2 = \Pr_\pi((\mathsf{RafP} \geq L)\ \mathbf{U}\ (\mathsf{RafP} \geq 4N))$$

where π is the distribution over states when satisfying for the first time the state formula $\mathsf{RafP} \geq L$ (previously called a filter).

The presented method only deals with time bounded reachability and not general "Until" formula. One way to generaliqe it, is to build an automaton encoding the formula, then the product of the automaton with the Markov chain and finally compute the probability to reach an accepting state of the automaton. However, this approach has two drawbacks: first, the size of the state space increases proportionally to the number of states of the automaton. Second, most of the simulation effort will be spent to reach a state satisfying first part of the formula, which is not a rare event. We use a more efficient approach: the system is simulated without importance sampling until one reaches a state where the first part of the formula holds. Then, importance sampling is used to compute the reachability probability of the second part of the formula. This method is sound as it is equivalent to use an importance sampling only on a part of the system.

This formula is parametrised by threshold L and scaling factor N. The results for increasing N and L are reported in Table V (confidence intervals are computed by Chernoff-Hoeffding method). As before, MARCIE cannot handle the case $N = 3$, the bottleneck being here the execution time.

It is clear that p_2 is an increasing function of L. More precisely, experiments point out that p_2 increases approximatively exponentially by at least one magnitude order when L is incremented. However, this dependency is less clear than the one of the first property.

The reduced model is the one used for the first property except for the values of the following parameters: here we choose $N_1 = 1$, $N_2 = N$ and $N_3 = 0$.

C. Signal propagation

To demonstrate that the increases of the signals are temporally ordered w.r.t. the layers in the signalling cascade, and by this way proving the travelling of the signals along the layers, we explore the following property: "What is the probability

Table VI. Experiments associated with p_3.

N	L	COSMOS		MARCIE		
		confidence interval	time	result	time	memory
2	2	[0.8018,0.8024]	4112	0.8021	75	730MB
2	3	[0.4201,0.4209]	7979	0.4205	137	723MB
2	4	[0.1081,0.1086]	10467	0.1084	163	725MB
2	5	[0.0122,0.0124]	11122	0.0123	123	725MB
2	6	$[6.20{\cdot}10^{-4}, 6.61{\cdot}10^{-4}]$	11185	$6.32{\cdot}10^{-4}$	129	725MB
2	7	$[1.02{\cdot}10^{-5}, 1.61{\cdot}10^{-5}]$	11194	$1.24{\cdot}10^{-5}$	156	725MB
3	6	[0.0136,0.0138]	14648	0.0137	17420	10.3GB
3	7	$[1.45{\cdot}10^{-3}, 1.51{\cdot}10^{-3}]$	14752	$1.48{\cdot}10^{-3}$	18155	10.3GB
3	8	$[9.99{\cdot}10^{-5}, 1.17{\cdot}10^{-4}]$	14739	$1.06{\cdot}10^{-4}$	18433	10.3GB
3	9	$[3.53{\cdot}10^{-6}, 7.36{\cdot}10^{-6}]$	14734	$4.86{\cdot}10^{-6}$	18353	10.3GB
3	10	$[1.03{\cdot}10^{-8}, 9.27{\cdot}10^{-7}]$	14743	$1.29{\cdot}10^{-7}$	18355	10.3GB
3	11	$[0, 5.30{\cdot}10^{-7}]$	14766	$1.48{\cdot}10^{-9}$	18047	10.3GB
4	8	$[1.47{\cdot}10^{-3}, 1.53{\cdot}10^{-3}]$	17669	Out of Memory		
4	9	$[1.52{\cdot}10^{-4}, 1.73{\cdot}10^{-4}]$	17628			
4	10	$[9.99{\cdot}10^{-6}, 1.59{\cdot}10^{-5}]$	17656			
4	11	$[1.54{\cdot}10^{-7}, 1.57{\cdot}10^{-6}]$	17632			
4	12	$[0, 5.30{\cdot}10^{-7}]$	17664			
5	8	$[6.92{\cdot}10^{-3}, 7.06{\cdot}10^{-3}]$	20367			
5	9	$[1.13{\cdot}10^{-3}, 1.19{\cdot}10^{-3}]$	20421			
5	10	$[1.46{\cdot}10^{-4}, 1.67{\cdot}10^{-4}]$	20419			

that, given the initial concentrations of RafP, MEKPP and ERKPP being zero, the concentration of RafP rises above some level L while the concentrations of MEKPP and ERKPP remain at zero, i.e., RafP is the first species to react?". While this property has its focus on the beginning of the signalling cascade, it is obvious how to extend the investigation by further properties covering the entire signalling cascade.

$$p_3 = \Pr((\mathsf{MEKPP} = 0) \wedge (\mathsf{ERKPP} = 0))\mathbf{U}(\mathsf{RafP} > L))$$

This formula is parametrized by L. Due to the lack of space only some values of L in $[0, 4N[$ are reported. The results for increasing N and L are given in Table VI. As can be observed, the probability to satisfy this property is not a rare event thus no importance sampling is required. Instead results are obtained by a plain Monte Carlo simulation generating 10 millions of trajectories. For $N > 3$ MARCIE requires more than $32GB$ of memory thus the computation was stopped. On the other hand, the memory requirement of COSMOS is around $50MB$ for all experiments.

We also observed that as expected the probability exponentially decreases with respect to L.

VII. CONCLUSION AND FUTURE WORK

We have studied rare events in signalling cascades with the help of an improved importance sampling method implemented in COSMOS. As demonstrated by means of our scalable case study, our method has been able to cope with huge models that could not be handled neither by numerical computations nor by standard simulations. In addition, analysis of the experiments has pointed out some interesting dependencies between the scaling parameter and the quantitative behaviour of the model.

In future work, we intend to incorporate other types of quantitative properties, such as the mean time a signal needs to exceed a certain threshold, the mean travelling time from the input to the output signal, or the relation between the variation of the enzymes of two consecutive levels. We also plan to analyse other biological systems for which the evaluation of

tiny probabilities might be relevant like mutation rates in growing bacterial colonies [19]. This kind of properties requires to specify new appropriate importance sampling methods.

REFERENCES

[1] B. Barbot, S. Haddad, M. Heiner, and C. Picaronny, "Rare event handling in signalling cascades," in Proceedings of the 6th International Conference on Advances in System Simulation (SIMUL'14), A. Arisha and G. Bobashev, Eds. Nice, France: XPS, Oct. 2014, pp. 126–131.

[2] M. Heiner, D. Gilbert, and R. Donaldson, "Petri nets for systems and synthetic biology," in SFM 2008, ser. LNCS, M. Bernardo, P. Degano, and G. Zavattaro, Eds., vol. 5016. Springer, 2008, pp. 215–264.

[3] G. Rubino and B. Tuffin, Rare Event Simulation using Monte Carlo Methods. Wiley, 2009.

[4] P. L'Ecuyer, V. Demers, and B. Tuffin, "Rare events, splitting, and quasi-Monte Carlo," ACM Trans. Model. Comput. Simul., vol. 17, no. 2, 2007.

[5] P. W. Glynn and D. L. Iglehart, "Importance sampling for stochastic simulations," Management Science, vol. 35, no. 11, 1989, pp. 1367–1392.

[6] B. Barbot, S. Haddad, and C. Picaronny, "Coupling and importance sampling for statistical model checking," in TACAS, ser. Lecture Notes in Computer Science, C. Flanagan and B. König, Eds., vol. 7214. Springer, 2012, pp. 331–346.

[7] ——, "Importance sampling for model checking of continuous time Markov chains," in Proceedings of the 4th International Conference on Advances in System Simulation (SIMUL'12), P. Dini and P. Lorenz, Eds. Lisbon, Portugal: XPS, Nov. 2012, pp. 30–35.

[8] A. Levchenko, J. Bruck, and P. Sternberg, "Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties," Proc Natl Acad Sci USA, vol. 97, no. 11, 2000, pp. 5818–5823.

[9] V. Chickarmane, B. N. Kholodenko, and H. M. Sauro, "Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation," Journal of Theoretical Biology, vol. 244, no. 1, January 2007, pp. 68–76.

[10] R. Breitling, D. Gilbert, M. Heiner, and R. Orton, "A structured approach for the engineering of biochemical network models, illustrated for signalling pathways," Briefings in Bioinformatics, vol. 9, no. 5, September 2008, pp. 404–421.

[11] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, Modelling with generalized stochastic Petri nets. John Wiley & Sons, Inc., 1994.

[12] M. Heiner, C. Rohr, and M. Schwarick, " MARCIE - Model checking And Reachability analysis done effiCIEntly ," in Proc. PETRI NETS 2013, ser. LNCS, J. Colom and J. Desel, Eds., vol. 7927. Springer, 2013, pp. 389–399.

[13] L. J. Bain and M. Engelhardt, Introduction to Probability and Mathematical Statistics, Second Edition. Duxbury Classic Series, 1991.

[14] W. Hoeffding, "Probability inequalities for sums of bounded random variables," Journal of the American Statistical Association, vol. 58, no. 301, 1963, pp. pp. 13–30.

[15] A. Jensen, "Markoff chains as an aid in the study of markoff processes," Skand. Aktuarietidskr, 1953.

[16] B. L. Fox and P. W. Glynn, "Computing Poisson probabilities," Commun. ACM, vol. 31, no. 4, 1988, pp. 440–445.

[17] P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin, "HASL: An expressive language for statistical verification of stochastic models," in Proceedings of the 5th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'11), Cachan, France, May 2011, pp. 306–315.

[18] P. Ballarini, B. Barbot, M. Duflot, S. Haddad, and N. Pekergin, "HASL: A new approach for performance evaluation and model checking from concepts to experimentation," Performance Evaluation, 2015.

[19] D. Gilbert, M. Heiner, F. Liu, and N. Saunders, "Colouring Space - A Coloured Framework for Spatial Modelling in Systems Biology," in Proc. PETRI NETS 2013, ser. LNCS, J. Colom and J. Desel, Eds., vol. 7927. Springer, June 2013, pp. 230–249.

Conflict Equivalence of Branching Processes

David Delfieu, Maurice Comlan
Polytech'Nantes
Institute of Research on Communications
and Cybernetics of Nantes, France
Email: david.delfieu@irccyn.ec-nantes.fr
maurice.comlan@irccyn.ec-nantes.fr
Médésu Sogbohossou
Polytechnic School of Abomey-Calavi
Laboratory of Electronics, Telecommunications
and Applied Computer Science, Abomey-Calavi, Benin
Email: sogbohossou_medesu@yahoo.fr

***Abstract*—For concurrent and large systems, specification step is a crucial point. Combinatory explosion is a limit that can be encountered when a state space exploration is driven on large specification modeled with Petri nets. Considering bounded Petri nets, technics like unfolding can be a way to cope with this problem. This paper is a first attempt to present an axiomatic model to produce the set of processes of unfoldings into a canonic form. This canonic form allows to define a conflict equivalence.**

***Index Terms*—Petri Nets; Unfolding; Branching process; Algebra.**

I. Introduction

The complexity and the criticity of some real-time system (transportation systems, robotics), but also the fact that we can no longer tolerate failures in less critical realtime systems (smartphones, warning radar devices) enforces the use of verification and validation methods. Petri nets are a widely used tool used to model critical real-time systems. The formal validation of properties is then based on the computation of state space. But, this computation faces generally, for highly concurrent and large systems, to combinatory explosion.

The specification of parallel components is generally modeled by the interleavings of the behavior of each components. This semantics of interleaving is exponentially costly in the computing of the state space. Partial order semantics have been introduced to shunt those interleavings. This semantics prevents combinatory explosion by keeping parallelism in the model.

The objective of this approach is to pursue a theoretical aspect: to speed up the identification of the branching processes of an unfolding. The notion of equivalence can be used to make a new type of reduction of unfoldings.

Finite prefixes of net unfoldings constitute a first transformation of the initial Petri Net (PN), where cycles have been flattened. This computation produces a process set where conflicts act as a discriminating factor. A conflict partitions a process in branching processes. An unfolding can be transformed into a set of finite branching processes. These processes constitute a set of acyclic graphs - several graphs can be produced when the PN contains parallelism - built with events and conditions, and structured with two operators: causality and true parallelism. An interesting particularity of an unfolding is that, in spite of the loss of global marking, these processes contain enough information to reconstitute the reachable markings of the original Petri nets. In most of the cases, unfoldings are larger than the original Petri net. This is provoked essentially when values of precondition places exceed the precondition of non simple conflicts. This produces a lot of alternative conditions. In spite of that, a step has been taken forward: cycles have been broken and the conflicts have structured the nets in branching processes.

This paper proposes proposes an algebraic model for the definition and the reduction of the branching process of an unfolding. This paper extends [1] to reset Petri nets. Reset arcs are particularly useful, they bring expressiveness and compactness. In the example presented in the Section VI, reset arcs allow to clear the states particularly when the user has several attempts to enter its code.

A lot of works have been proposed to improve unfolding algorithms [2][3][4][5]. Is there another way to draw on recent works about unfolding? In spite of the eventual increase of the size of the net unfoldings, the suppression of conflicts and loops has decreased its structural complexity, allowing to compute the state space and to the extract of semantic information.

From a developer's point of view an unfolding can be efficiently coded by a boolean table of events. This table describes every pair to pair relation between events. This table has been the starting point of our reflection: it stresses the point that a new connector can be defined to express that a set of events belong to the same process. This connector allows to aggregate all the events of a branching process. For example, a theorem is proposed to compute all the branching processes, in canonic form, for chains of conflicts of the kind illustrated in Figure 1.

The work presented in this paper takes place in the context

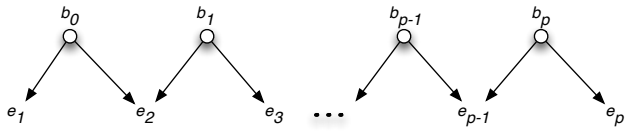


Figure 1. Chain of conflicts.

TABLE I. Process syntax.

$Capacity$	α	:=	$\bar{x} \mid x \mid \tau$
$Proces$	p	::=	$\alpha.p \mid p \mid\mid q \mid p+q \mid D(\tilde{x}) \mid p\backslash x \mid 0$

of combining process algebra [6][7] and Petri nets [8].

The axiomatic model of Milner's process with Calculus of Communicating Systems (CCS) is compared with the branching processes and related to other works in Section II. Then, after a brief presentation of Petri nets and unfoldings in Section III, Section IV presents our contribution with the definition of an axiomatic framework and the description of properties. The last section presents examples, in particular, illustrating a conflict equivalence.

II. Related Work

Process algebra appeared with Milner [7] on the Calculus of Communicating Systems (CCS) and the Communicating Sequential Processes (CSP) of Hoare [6]. These approaches are not equivalent but share similar objectives. The algebra of branching process proposed in this paper is inspired by the process algebra of Milner. CCS is based on two central ideas: The notion of observability and the concept of synchronized communication; CCS is as an abstract code that corresponds to a real program whose primitives are reduced to simple send and receive on channels. The terms (or agents) are called processes with interaction capabilities that match requests communication channels. The elements of the alphabet are observable events and concurrent systems (processes). They can be specified with the use of three operators: sequence, choice, and parallelism. A main axiom of CCS is the rejection of distributivity of the sequence upon the choice. Let p and q be two processes, the complete syntax of process is described in the Table I.

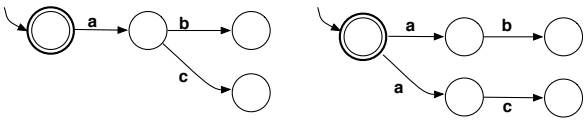


Figure 2. Milner: rejection of distributivity of sequence on choice.

Consider an observer. In the left automaton of Figure 2, after the occurrence of the action a, he can observe either b or c. In right automaton, the observation of a does not imply that b and c stay observable. The behavior of the two automata are not equivalent.

In CCS, Milner defines the observational equivalence. Two automata are observational equivalent if there are bisimular. On a algebraic point of view, the distributivity of the sequence on the choice is rejected in equation (1):

$$a.(b+c) \neq_{behaviorally} a.b + a.c \quad (1)$$

The key point of our approach is based on the fact that this distributivity is not rejected in occurrence nets. The timing of the choices in a process is essential [9]. The nodes of occurrence nets are events. An event is a fired transition of the underlying Petri net. In CCS, an observer observes possible futures. In occurrence nets, the observer observes arborescent past. This controversy in the theory of concurrency is an important topic of linear time versus branching time. In the model, equation (2) holds:

$$a \prec (b \perp c) \equiv (a \prec b) \perp (a \prec c) \quad (2)$$

Equation (2) is a basic axiom of our algebraic model. The equivalence relation differs then from bisimulation equivalence. This relation will be defined in the following with the definition of the canonic form of an unfolding.

Branching process does not fit with process algebra on numerous other aspects. For example, a difference can be noticed about parallelism. While unfolding keeps true parallelism, process algebra considers a parallelism of interleaving. Another difference is relative to events and conditions, which are nodes of different nature in an unfolding. Conditions and events differ in term of ancestor. Every condition is produced by at most one event ancestor (none for the condition standing for m_0, the initial marking), whereas every event may have 1 or n condition ancestor(s).

In CCS, there is no distinction between conditions and events. Moreover, conditions will be consumed defining processes as set of events. However, a lot of works [5][9][10] have shown the interest of an algebraic formalization: it allows the study of connectives, the compositionally and facilitates reasoning (tools like [11]). Let have two Petri nets; it is questionable whether they are equivalent. In principle, they are equivalent if they are executed strictly in the same manner. This is obviously a too restrictive view they may have the same capabilities of interaction without having the same internal implementations. These works resulted to find matches (rather flexible and not strict) between nets. Mention may be made among other the occurrence net equivalence [12], the bisimulation equivalence [13], the partial order equivalence [14], or the ST-bisimulation equivalence [15]. These different equivalences are based either on the isomorphism between the unfolding of nets or on observable actions or traces of the execution of Petri nets or other criteria.

The approach developed in this paper proposes a new equivalence, which is weaker than a trace equivalence; it does not preserves traces but preserves conflicts. The originality of the approach is to encapsulate causality and concurrency in a new operator, which "aggregates" and "abstracts" events in a process. This new operator reduces the representation and accelerates the reduction process. This paper intends first, to give an algebraic model to an unfolding, and second,

to establish a canonic form leading to the definition of an equivalence conflict.

III. Unfolding a Petri Net

In this section, Petri nets and unfolding of Petri nets are presented.

A. Petri Net

A Petri net [8] $\mathcal{N} =< P, T, \mathcal{W} >$ is a triple with: P, a finite set of places, T, the finite set of transitions, $P \cup T$ are nodes of the net; ($P \cap T = \emptyset$ signifies that P and T are disjoint), and $\mathcal{W} : (P \times T) \cup (T \times P) \to \mathcal{N}$, the flow relation defining arcs (and their valuations) between nodes of $\mathcal{N}$. A *marking* of $\mathcal{N}$ is a multiset M: $P \to \{0, 1, 2, ...\}$ and the *initial marking* is denoted M_0.

The pre-set (resp. post-set) of a node x is denoted ${}^\bullet x = \{y \in P \cup T \mid \mathcal{W}(y, x) > 0\}$ (resp. $x^\bullet = \{y \in P \cup T \mid \mathcal{W}(x, y) > 0\}$). A transition $t \in T$ is said *enabled* by m iff: $\forall p \in {}^\bullet t,\ m(p) \geq \mathcal{W}(p, t)$. This is denoted: $m \xrightarrow{t}$ Firing of t leads to the new marking m' ($m \xrightarrow{t} m'$): $\forall p \in P,\ m'(p) = m(p) - \mathcal{W}(p, t) + \mathcal{W}(t, p)$. The initial marking is denoted m_0.

A Petri net is *k-bounded* iff $\forall m$, reachable from $m_0, m(p) \leq k$ (with $p \in P$). It is said *safe* when *1-bounded*. Two transitions are in a *structural conflict* when they share at least one pre-set place; a conflict is *effective* when these transitions are both enabled by a same marking. The considered Petri nets in this paper are k-bounded.

Reset arcs constitute an extension of Petri nets. These arcs does not change the enabling rules of transitions [16]. If $Rst(p, t)$ represents the set of reset arcs from a transition t to a place p. If $M \xrightarrow{t} M'$ then $\forall p \in P$ such as $Rst(p, t) = 0$, $M'(p) = 0$. But if $W(t, p) > 0$ then $M'(p) = W(t, p)$. The firing rule is defined by the following relation

$$\forall p \in P, \quad M' = (M - Pre(p, t)) \ . \ R(p, t) + Post(p, t)$$

where "." is the Hadamard matrix product.

Definition 1 (Reset arc Petri Nets)**.** *A reset arc Petri Nets is a tuple $N_R =< P, T, W, R >$ with $< P, T, W >$ a Petri nets and $Rst : P \times T \to \{0, 1\}$ is the set of reset arcs ($Rst(p, t) = 0$ is there exists a reset arc binding p to t, else $Rst(p, t) = 1$).*

B. Unfolding

In [3], the notion of *branching process* is defined as an initial part of a run of a Petri net respecting its partial order semantics and possibly including non deterministic choices (conflicts). This net is acyclic and the largest branching process of an initially marked Petri net is called *the* unfolding of this net. Resulting net from an unfolding is a labeled occurrence net, a Petri net whose places are called *conditions* (labeled with their corresponding place name in the original net) and transitions are called *events* (labeled with their corresponding transition name in the original net).

An occurrence net [17] is a net $\mathcal{O} =< \mathcal{B}, \mathcal{E}, \mathcal{F} >$, where $\mathcal{B}$ is the set of *conditions* (places), $\mathcal{E}$ is the set of *events* (transitions), and $\mathcal{F}$ the flow relation (1-valued arcs), such that:

- for every b $\in \mathcal{B}$, $|{}^\bullet b| \leq 1$;
- $\mathcal{O}$ is acyclic;
- for every e $\in \mathcal{E}$, ${}^\bullet e \neq \emptyset$;
- $\mathcal{O}$ is finited preceded;
- no element of $\mathcal{B} \cup \mathcal{E}$ is in conflict with itself;
- $\mathcal{F}^+$, the transitive closure of $\mathcal{F}$, is a strict order relation.

$Min(\mathcal{O}) = \{b \mid b \in \mathcal{B}, |{}^\bullet b| = 0\}$ is the minimal conditions set: the set of conditions with no ancestor can be mapped with the initial marking of the underlying Petri net. Also, $Max(\mathcal{O}) = \{x \mid x \in \mathcal{B} \cup \mathcal{E}, |x^\bullet| = 0\}$ are maximal nodes. A *configuration C* of an occurence net is a set of events satisfying:

- if $e \in C$ then $\forall e' \prec e$ implies $e' \in C$ (C is causally closed);
- $\forall e, e' \in C : \neg(e \perp e')$ (C is conflict-free).

A *local configuration* [e] of an event e is the set of event e', such that $e' \prec e$.

Three kinds of relations could be defined between the nodes of $\mathcal{O}$:

- The strict causality relation noted $\prec$: for $x, y \in \mathcal{B} \cup \mathcal{E},\ x \prec y$ if $(x, y) \in \mathcal{F}^+$ (for example $e_3 \prec e_6$, in Figure 3.b).
- The conflict relation noted $\perp$: $\forall b \in \mathcal{B}$, if $e_1, e_2 \in b^\bullet$ ($e_1 \neq e_2$), then e_1 and e_2 are in *conflict relation*, denoted $e_1 \ \perp \ e_2$ (for example $e_4 \perp e_5$, in Figure 3.b).
- The concurrency relation noted $\wr$: $\forall x, y \in \mathcal{B} \cup \mathcal{E}$ ($x \neq y$), $x \wr y$ ssi $\neg((x \prec y) \vee (y \prec x) \vee (x \ \perp \ y))$ (for example $e_2 \wr e_3$, in Figure 3.b).

Remark 1. *The transitive aspect of $\mathcal{F}^+$ implies a transitive definition of strict causality.*

A set $B \subseteq \mathcal{B}$ of conditions such as $\forall b, b' \in B, b \neq b' \Rightarrow b \wr b'$ is a *cut*. Let B be a *cut* with $\forall b \in B, \nexists b' \in \mathcal{B} \backslash B,\ b \wr b'$, B is the *maximal cut*.

Definition 2. The unfolding $Unf_F \stackrel{\text{def}}{=} < \mathcal{O}_F, \lambda_F >$ *of a marked net $< \mathcal{N}, m_0 >$, with $\mathcal{O}_F \stackrel{\text{def}}{=} < \mathcal{B}_F, \mathcal{E}_F, \mathcal{F}_F >$ an occurrence net and $\lambda_F : \mathcal{B}_F \cup \mathcal{E}_F \to \mathcal{P} \cup \mathcal{T}$ (such as $\lambda(\mathcal{B}_F) \subseteq \mathcal{P}$ and $\lambda(\mathcal{E}_F) \subseteq \mathcal{T}$) a labeling function, is given by:*

1) *$\forall p \in \mathcal{P}$, if $m_0(p) \neq \emptyset$, then $B_p \stackrel{\text{def}}{=} \{b \in \mathcal{B}_F \mid \lambda_F(b) = p \wedge {}^\bullet b = \emptyset\}$ and $m_0(p) = |B_p|$;*
2) *$\forall B_t \subseteq \mathcal{B}_F$ such as B_t is a cut, if $\exists t \in \mathcal{T}, \lambda_F(B_t) = {}^\bullet t \wedge |B_t| = |{}^\bullet t|$, then:*
   a) *$\exists! e \in \mathcal{E}_F$ such as ${}^\bullet e = B_t \wedge \lambda_F(e) = t$;*
   b) *if $t^\bullet \neq \emptyset$, then $B'_t \stackrel{\text{def}}{=} \{b \in \mathcal{B}_F \mid {}^\bullet b = \{e\}\}$ is as $\lambda_F(B'_t) = t^\bullet \wedge |B'_t| = |t^\bullet|$;*
   c) *if $t^\bullet = \emptyset$, then $B'_t \stackrel{\text{def}}{=} \{b \in \mathcal{B}_F \mid {}^\bullet b = \{e\}\}$ is as $\lambda_F(B'_t) = \emptyset \wedge |B'_t| = 1$;*

3) $\forall B_t \subseteq \mathcal{B}_F$, *if* B_t *is not a cut , then* $\nexists e \in \mathcal{E}_F$ *such as* ${}^\bullet e = B_t$.

Definition 2 represents an *exhaustive* unfolding algorithm of $< \mathcal{N}, m_0 >$. In 1., the algorithm for the building of the unfolding starts with the creation of conditions corresponding to the initial marking of $< \mathcal{N}, m_0 >$ and in 2., new events are added one at a time together with their output conditions (taking into account sink transitions). In 3., the algorithm requires that any event is a possible action: there are no adding nodes to those created in items 1 and 2. The algorithm does not necessary terminate; it terminates if and only if the net $< \mathcal{N}, m_0 >$ does not have any infinite sequence. The sink transitions (ie $t \in \mathcal{T}, t^\bullet = \emptyset$) are taken into account in 2.(c).

Let be $\mathcal{E} \subset \mathcal{E}_F$. The occurrence net $\mathcal{O} \stackrel{\text{def}}{=} < \mathcal{B}, \mathcal{E}, \mathcal{F} >$ associated with $\mathcal{E}$ such as $\mathcal{B} \stackrel{\text{def}}{=} \{b \in \mathcal{B}_F \mid \exists e \in \mathcal{E}, b \in {}^\bullet e \cup e^\bullet\}$ and $\mathcal{F} \stackrel{\text{def}}{=} \{(x,y) \in \mathcal{F}_F \mid x \in \mathcal{E} \vee y \in \mathcal{E}\}$ is *a prefix* of $\mathcal{O}_F$ if $Min(\mathcal{O}) = Min(\mathcal{O}_F)$. By extension, $Unf \stackrel{\text{def}}{=} < \mathcal{O}, \lambda >$ (with λ, the restriction of λ_F to $\mathcal{B} \cup \mathcal{E}$) is a prefix of unfolding Unf_F.

It should be noted that, according to the implementation, the names (the elements in the sets $\mathcal{E}$ and $\mathcal{B}$) given to nodes in the same unfolding can be different. A name can be independently chosen in an implementation using a tree formed by its causal predecessors and the name of the corresponding nodes in $\mathcal{N}$ [3].

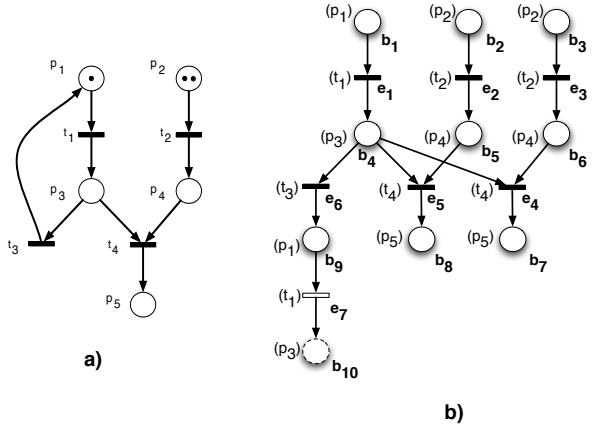


Figure 3. a) Petri net, b) Unfolding.

Definition 3. *A causal net* $\mathcal{C}$ *is an occurrence net* $\mathcal{C} \stackrel{\text{def}}{=} < \mathcal{B}, \mathcal{E}, \mathcal{F} >$ *such as:*

1) $\forall e \in \mathcal{E} : e^\bullet \neq \emptyset \ \wedge \ {}^\bullet e \neq \emptyset$;
2) $\forall b \in \mathcal{B} : |b^\bullet| \leq 1 \ \wedge \ |{}^\bullet b| \leq 1$.

Definition 4. $\mathcal{P}_i = (\mathcal{C}_i, \lambda_F)$ *is a* process *of* $< \mathcal{N}, m_0 >$ *iff:* $\mathcal{C}_i \stackrel{\text{def}}{=} < \mathcal{B}_i, \mathcal{E}_i, \mathcal{F}_i >$ *is a causal net and* $\lambda : \mathcal{B}_i \cup \mathcal{E}_i \rightarrow P \cup T$ *is a labeling fonction such as:*

1) $\mathcal{B}_i \subseteq \mathcal{B}_F$ *and* $\mathcal{E}_i \subseteq \mathcal{E}_F$
2) $\lambda_F(\mathcal{B}_i) \subseteq P$ *and* $\lambda_F(\mathcal{E}_i) \subseteq T$;
3) $\lambda_F({}^\bullet e) = {}^\bullet \lambda_F(e)$ *and* $\lambda_F(e^\bullet) = \lambda_F(e)^\bullet$
4) $\forall e_i \in \mathcal{E}_i, \ \forall p \in P : \ \mathcal{W}(p, \lambda_F(e)) = |\lambda^{-1}(p) \cap {}^\bullet e| \ \wedge \ \mathcal{W}(\lambda_F(e), p) = |\lambda^{-1}(p) \cap e^\bullet|$
5) *If* $p \in Min(P) \Rightarrow \exists b \in \mathcal{B}_i : \ {}^\bullet b = \emptyset \ \wedge \ \lambda_F(b) = p$

$Max(\mathcal{C}_i)$ is the *state* of $\mathcal{N}$. $Min(\mathcal{C}_i)$ and $Max(\mathcal{C}_i)$ are (resp. minimum) maximum cuts. Generally, any maximal cut $B \subseteq \mathcal{B}_i$ corresponds to a reachable marking m of $< \mathcal{N}, m_0 >$ such as $\forall p \in \mathcal{P}, m(p) = |B_p|$ avec $B_p = \{b \in B \mid \lambda(b) = p\}$.

The *local configuration* of an event e is defined by: $[e] \stackrel{\text{def}}{=} \{e' \mid e' \prec e\} \cup \{e\}$ and is a process. For example of unfolding in Figure 3.b: $[e_4] \stackrel{\text{def}}{=} \{e_1, e_3, e_4\}$.

The conflicts in an unfolding derive from the fact that there is a reachable marking (a cut in an unfolding) such as two or many transitions of a labelled net $< \mathcal{N}, m_0 >$ are *enabled* and the firing of one transition disable other. Whence the proposition:

Proposition 1. *Let be* $e_1, e_2 \in \mathcal{E}_F$. *If* $e_1 \ \perp \ e_2$, *then there* $\exists (e'_1, e'_2) \in [e_1] \times [e_2]$ *such as* ${}^\bullet e'_1 \cap {}^\bullet e'_2 \neq \emptyset$ *et* ${}^\bullet e'_1 \cup {}^\bullet e'_2$ *is a cut.*

IV. Branching Process Algebra

The Section III-B showed how unfolding exhibits causal nets and conflicts. Otherwise, every couple of events that are not bounded by a causal relation or the same conflict set are in concurrency. Then, an unfolding allows to build a $2D$-table making explicit every binary relations between events. Practically, this table establishes the relations of causality and exclusion. If a binary relation is not explicit in the table, it means that the couple of events are in a concurrency relation.

Let $\mathcal{EB} = \mathcal{E} \cup \mathcal{B}$ a finite alphabet, composed of the events and the conditions generated by the unfolding. The event table (produced by the unfolding) defines for every couple in $\mathcal{EB}$ either a causality relation $\mathcal{C}$, either a concurrency relation $\mathcal{I}$ or an exclusive relation $\mathcal{X}$. These sets of binary relations dot not intersect and the following expressions can be deduced:

$$Unf/_{\mathcal{X}} = \mathcal{C} \cup \mathcal{I} \tag{3}$$
$$Unf/_{\mathcal{C}} = \mathcal{X} \cup \mathcal{I} \tag{4}$$
$$Unf/_{\mathcal{I}} = \mathcal{C} \cup \mathcal{X} \tag{5}$$

To illustrate these relation sets, the negation operator noted $\neg$ can be introduced. Then, equations (3), (4), (5) lead to (6), (7), (8):

$$\neg((e_1, e_2) \in \mathcal{I}) \ \Leftrightarrow \ (e_1, e_2) \in \mathcal{C} \cup \mathcal{X} \tag{6}$$
$$\neg((e_1, e_2) \in \mathcal{C}) \ \Leftrightarrow \ (e_1, e_2) \in \mathcal{I} \cup \mathcal{X} \tag{7}$$
$$\neg((e_1, e_2) \in \mathcal{X}) \ \Leftrightarrow \ (e_1, e_2) \in \mathcal{C} \cup \mathcal{I} \tag{8}$$

Equation (8) expresses that if two events are not in conflict they are in the same branching process. Let us now define the union of binary relations $\mathcal{C}$ and $\mathcal{I}$: $\mathcal{P} = \mathcal{C} \cup \mathcal{I}$. For every couple $(e_1, e_2) \in \mathcal{P}$, either (e_1, e_2) are in causality or in concurrency: $\mathcal{P}$ is the union of every branching process of an unfolding.

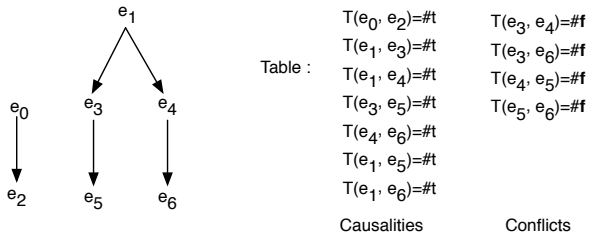


Figure 4. Unfolding.

a) Example: Figure 4 represents an unfolding (in the left part) and a Table T (right part), which defines the event relations of the unfolding.

In Figure 4, the Table T contains 7 causal relations and 4 conflict relations. (e_0, e_4) is not (negation) in the table, it means that e_0 and e_4 are concurrent. Moreover, if two events are not in conflict (consider e_0 and e_6): (e_0, e_6) is not a key of the table, (e_0, e_6) are in concurrency and thus, those events belongs to the same branching process.

A. Definition of the Algebra

The starting point of this work is based on the fact that the logical negation operator articulates the relation between two sets: the process set $\mathcal{P}$ and the exclusion set $\mathcal{X}$. As mentioned in Section IV, $\mathcal{C}$, $\mathcal{I}$ *and* $\mathcal{P}$ does not intersect, then semantically, if a couple of events is not in a relation of exclusion (noted $\perp$), the events are in $\mathcal{P}$. $\mathcal{P}$ contains binary relations between events that are in branching process.

To express that events are in the same branching process, a new operator noted $\oplus$ is introduced. An algebra describing branching process can be defined as follow:

$$\{\mathcal{U}, \prec\ ,\ \wr\ ,\ \perp\ , \oplus,\ \neg\}$$

Let us note; $* = \oplus, \prec,$ or $\perp$, $\#t$ the void process, and $\#f$ the false process. Here is the formal signature of the language:

- $\forall e \in \mathcal{EB}, e \in \mathcal{U}, \#t \in \mathcal{U}, \#f \in \mathcal{U}$
- $\forall e \in \mathcal{U}, \neg e \in \mathcal{U}$
- $\forall (e_1, e_2) \in \mathcal{U}^2, e_1 * e_2 \in \mathcal{U}$.

B. Definition of operators

1) Causality: $\mathcal{C}$ is the set of all the causalities between every elements of $\mathcal{EB}$. $e_1 \prec e_2$ if e_1 is in the local configuration of e_2, i.e., the Petri net contains a path with at least one arc leading from e_1 to e_2:

$$e_1 \prec e_2 \text{ if } e_1 \in [e_2] \tag{9}$$

- $\prec$ is associative: $e_1 \prec (e_3 \prec e_5) \equiv (e_1 \prec e_3) \prec e_5$;
- $\prec$ is transitive: $(e_1 \prec e_3) \vee (e_3 \prec e_5) \equiv e_1 \prec e_5$;
- $\prec$ is not commutative: $e_1 \prec e_3$ but $e_3 \neg \prec e_1$;
- $\#t$ is the neutral element for $\prec$: $\#t \prec e \equiv e$;
- every element of $\mathcal{EB}$ has an opposite: $\#f \prec e \equiv \neg e$.

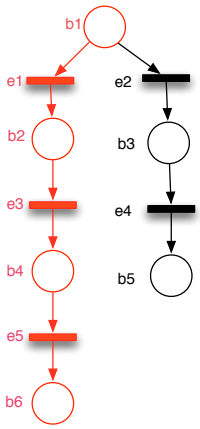


Figure 5. Causalite.

2) Exclusion: $\mathcal{X}$ is the set of all the exclusion relations between every elements of $\mathcal{EB}$. Two events e and e' are in exclusion if the net contains two paths $b\ e_1\ ...\ e$ and $b\ e_2\ ...\ e'$ starting at the same condition b and $e_1 \neq e_2$:

$$e_1 \perp e_2 \equiv ((\bullet e_1 \cap \bullet e_2 \neq \emptyset) \text{ or } (\exists e_i, e_i \prec e_2 \text{ and } e_1 \perp e_i)) \tag{10}$$

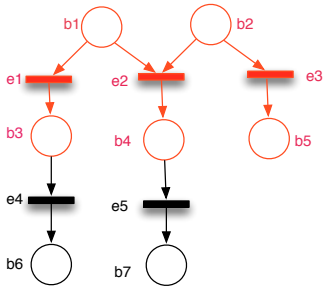


Figure 6. Exclusion.

- $\perp$ is commutative: $e_1 \perp e_2 \equiv e_2 \perp e_1$;
- $\perp$ is associative: $e_1 \perp (e_2 \perp e_3) \equiv (e_1 \perp e_2) \perp e_3$;
- $\perp$ is not transitive: $(e_1 \perp e_2) \vee (e_2 \perp e_3)$ but $e_1 \neg \perp e_3$;
- $\#f$ is the neutral element for $\perp$: $e \perp \#f \equiv e$;
- $\#t$ is the absording element for $\perp$: $e \perp \#t \equiv \#t$.

3) Concurrency: $\mathcal{I}$ is the set of every couple of element of $\mathcal{EB}$ in concurrency. e_1 and e_2 are in concurrency if the occurrence of one is independent of the occurrence of the other. So, $e_1 \wr e_2$ iff e_1 and e_2 are neither in causality neither in exclusion.

$$e_1 \wr e_2 \equiv \neg((e_1 \perp e_2) \text{ or } (e_1 \prec e_2) \text{ or } (e_2 \prec e_1)) \tag{11}$$

- $\wr$ is commutative: $e_1 \wr e_5 \equiv e_5 \wr e_1$;
- $\wr$ is associative: $e_1 \wr (e_5 \wr e_7) \equiv (e_1 \wr e_5) \wr e_7$;
- $\wr$ is not transitive: $(e_1 \wr e_5) \vee (e_5 \wr e_2)$ but $e_1 \perp e_2$;
- $\#t$ is the neutral element for $\wr$: $e \wr \#t \equiv e$;
- $\#f$ is an absorbing element for $\wr$: $e \wr \#f \equiv \#f$.

4) Process: $\oplus$ aggregates events in one process. Two events e_1 and e_2 are in the same process if e_1 causes e_2 or if e_1 is

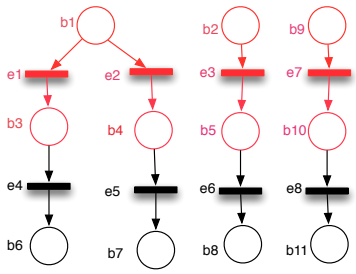


Figure 7. Concurrency.

concurrent with e_2:

$$e_1 \oplus e_2 \equiv (e_1 \prec e_2) \text{ or } (e_2 \prec e_1) \text{ or } (e_1 \wr e_2) \quad (12)$$

This operator constitutes an abstraction that hides in a black box causalities and concurrencies. The meaning of this operator is similar to the linear connector $\oplus$ of MILL [18]. It allows to aggregates resources. But, in the context of unfolding, events or conditions are unique and then they cannot be counted. Thus, this operator is here idempotent.

The expression $e_1 \oplus e_2$ defines that e_1 and e_2 are in the same process.

Note that $(\oplus\ e_1\ e_2\ ...\ e_{n-1}\ e_n)$ will abbreviate $(e_1 \oplus\ e_2 \oplus e_3 \oplus ...e_{n-1} \oplus\ e_n)$

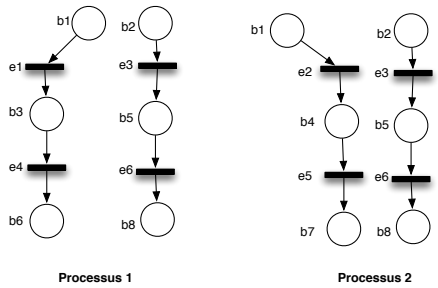


Figure 8. Process.

- $\oplus$ is commutative, associative, and transitive (definition of $\oplus$);
- Idempotency: $e \oplus e \equiv e$
- Neutral element: $e \oplus \#t \equiv e$
- Absorbing element: $e \oplus \#f \equiv \#f$
- $e \oplus \neg e \equiv \#f$

C. Axioms

The following axioms stem directly from previous assumptions and definitions made upon the algebraic model:

Axiom 1 (Distributivity of $\prec$)**.**

$$e \prec (e_1 \perp e_2) \equiv_{def} (e \prec e_1) \perp (e \prec e_2)$$

This first axiom constitutes the basis of our approach. As discussed in the Section II, on the contrary of CCS, e is distributed onto two expressions, giving alternative processes.

Axiom 2 (Definition of $\oplus$)**.**

$$e_1 \oplus e_2 \equiv_{def} (e_1 \prec e_2) \perp (e_2 \prec e_1) \perp (e_1 \wr e_2)$$

$\oplus$ aggregates two elements in a process. Two elements are in a process if they are concurrent or in a causality relation.

Axiom 3 ($\prec$)**.**

$$e_1 \prec e_2 \equiv_{def} \neg e_1 \perp (e_1 \oplus e_2)$$

A causality can be expressed by two processes in exclusion: either $\neg e_1$: e_1 has not occurred either $e_1 \oplus e_2$: e_1 and e_2 within the same process.

Axiom 4 (Duality between $\oplus$ and $\perp$)**.**

$$e_1 \oplus e_2 \equiv_{def} e_1 \neg\perp e_2 \qquad e_1 \neg\oplus e_2 \equiv_{def} e_1 \perp e_2$$

This axiom comes from the introduction of the operator $\neg$ discussed in the beginning of the Section IV. It expresses that $\mathcal{P}$ and $\mathcal{X}$ are complementary sets.

Axiom 5 (Exclusion)**.**

$$e_1 \perp e_2 \equiv_{def} (\neg e_1 \oplus e_2) \perp (e_1 \oplus \neg e_2)$$

The fifth axiom expresses that a conflict can be considered as two processes in conflict.

D. Distributivities

The distributivities over $\perp$ are used in the transformation of an expression in the canonical form (Section V). The other distributivities will be used in the reduction process.

1) Distributivities over $\wr$:

- $\prec$ is distributive over $\wr$:

$$e \prec (e_1 \wr e_2) \equiv (e \prec e_1) \wr (e \prec e_2)$$

- $\perp$ is distributive over $\wr$:

$$e \perp (e_1 \wr e_2) \equiv (e \perp e_1) \wr (e \perp e_2)$$

- $\oplus$ is distributive over $\wr$:

$$e \oplus (e_1 \wr e_2) \equiv (e \oplus e_1) \wr (e \oplus e_2)$$

2) Distributivities over $\perp$:

- $\prec$ is distributive over $\perp$ (Axiom 1):

$$e \prec (e_1 \perp e_2) \equiv (e \prec e_1) \perp (e \prec e_2)$$

- $\wr$ is distributive over $\perp$:

$$e \wr (e_1 \perp e_2) \equiv (e \wr e_1) \perp (e \wr e_2)$$

- $\oplus$ is distributive over $\perp$:

$$e \oplus (e_1 \perp e_2) \equiv (e \oplus e_1) \perp (e \oplus e_2)$$

3) Distributivities over $\oplus$:

- $\perp$ is distributive over $\oplus$:

$$e \perp (e_1 \oplus e_2) \equiv (e \oplus e_1) \perp (e \oplus e_2)$$

- $\wr$ is distributive over $\oplus$:

$$e \wr (e_1 \oplus e_2) \equiv (e \oplus e_1) \wr (e \oplus e_2)$$

E. Derivation Rules

This section gives a set of rules, which transform branching processes toward a canonical form. These transformations preserve conflicts whereas $\prec$ and $\wr$ are transformed in $\oplus$.

Let us note b a condition, e an event and E a well formed formula on the algebra. These rules allow to reduct process:

1) Modus Ponens:

$$\frac{\vdash \oplus\ b\ ... \quad \vdash \oplus\ b\ ... \prec e}{\vdash e}\ MP_1$$

$$\frac{\vdash e \quad \vdash e \prec \oplus\ b\ ...}{\vdash \oplus\ e\ b\ ...}\ MP_2$$

Where $\oplus\ b\ ...$ stands for the general form for $(\oplus\ b_1\ b_2\ ...\ b_n)$. MP_1 expresses that the set of conditions $\oplus\ b\ ...$ are consumed by the causality, whereas in MP_2, e stays in the conclusion.

2) Dual form:

$$\frac{\vdash \neg e_1 \quad \vdash e_1 \prec e_2}{\vdash \neg e_1 \oplus \neg e_2}\ MP'$$

3) Simplification:

$$\frac{\vdash \neg e_1 \oplus E}{\vdash E}\ S_1$$

$$\frac{\vdash \oplus\ b\ ...\ E}{\vdash E}\ S_2$$

Those rules are applied, *in fine*, to clear not pertinent informations in the process. S_1 rule is applied, to clear the negations, whereas S_2 is applied to clear the conditions, which have not been consumed.

4) Reduction of $\wr$:

$$\frac{\vdash e_1 \wr e_2}{\vdash e_1 \oplus e_2}\ Par$$

This rule corresponds to the definition of $\oplus$
These rules have been defined to lead to a canonic form.

V. Canonic Form and Conflict Equivalence

A *canonic form* is a relation expressed on elements of $\mathcal{EB}$ and with the operators $\oplus$ and $\perp$ ordered by an alphanumeric sort on the name of its symbol. This definition of the canonic form allows to define an equivalence called a "conflict equivalence".

Theorem 1 (Canonical form)**.** *Let us consider an unfolding U, this form can be reduced in the following form:*

$$U = (\perp\ P_1\ P_2\ ...\ P_n), \text{where} \quad P_i = (\oplus\ e_{i_1}... \ e_{i_n})$$

This form is canonic and exhibits every processes P_i of the unfolding.

Proof. In an unfolding every causality ($\prec$) and every partial order ($\wr$) can be reduced in $\oplus$ by deduction rules Modus Ponens (MP, MP_1, MP_2), Simplification rule (S) and Par (see Section IV-E).

Moreover, $\oplus$ and $\perp$ are mutually distributive, so $\perp$ can be factorized in every sub-formula to reach the higher level of the formula. In fine, an alphanumeric sort on symbols of the processes can be applied to assure the unicity of the form. □

This canonic form preserves conflicts, let us now define a *conflict equivalence*:

Definition 5 (Conflict Equivalence)**.** *Let us U_1, U_2 unfoldings of Petri nets:*

$$U_1 \approx_{conf} U_2 \qquad \textit{iff they have the same canonic form.}$$

Remark 2. *A process is an aggregate set of events, where $\prec$ and $\wr$ are hidden. This equivalence is lower than a trace equivalence: each process P_i is an abstraction of a set of traces.*

A. Theorems

The properties of operators (definitions, axioms and distributivites) allow to define theorems, which are congruences.

Theorem 2 (Conflict)**.**

$$e_1 \prec (e_2 \perp e_3) \equiv (e_1 \prec (e_2 \oplus \neg e_3)) \perp (e_1 \prec (\neg e_2 \oplus e_3))$$

Proof.

$$\begin{aligned} e_1 \prec (e_2 \perp e_3) \quad &\equiv_{Ax_5} \quad e_1 \prec ((e_2 \oplus \neg e_3) \perp (\neg e_2 \oplus e_3) \\ &\equiv_{dist} \quad (e_1 \prec (e_2 \oplus \neg e_3)) \perp (e_1 \prec (\neg e_2 \oplus e_3)) \end{aligned}$$

□

This theorem expresses how to develop a conflict and the following theorem allows to reduce processes:

Theorem 3 (Absorption)**.** *Let E, F some processes:*

$$E \perp (E \oplus F) \equiv E \oplus F$$

Proof.

$$\begin{array}{rcl} E \perp (E \oplus F) & \equiv & (E \oplus \#t) \perp (E \oplus F) \\ & \equiv_{Neutral} & E \oplus (\#t \perp F) \\ & \equiv & E \oplus F \end{array}$$

□

B. Chain of conflicts

This section presents a theorem that computes the branching process in canonic form of a chain of conflict illustrated in Figure 9.

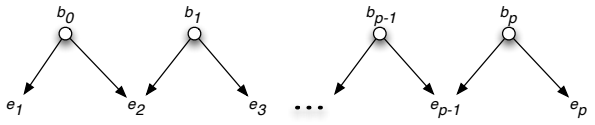


Figure 9. Chain of conflicts.

The axiomatic representation of the unfolding is:

$$U = ((\oplus\ b_0\ b_1\ ...\ (b_0 \prec (e_1 \perp e_2))(b_1 \prec (e_2 \perp e_3))...)$$

After some steps of reduction $(MP + S)$:

$$U = (e_1 \perp e_2 \perp ... \perp\ e_p)$$

Let us note:

- $l^1 = (e_1, e_2, ...e_n)$, $l^2 = (e_2, ...e_n)$
- l_i the i^{th} element of a list l.
- If e_i is an element of the list l, let us note $indice(e_i)$ the position of e_i in l.

Remark 3. *In the list of event constituting a chain of conflict ($l = (e_1, e_2, ...e_n)$), for every event e_i, the next (resp. previous) event in the same branching process is e_{i+2} or e_{i+3} (resp. e_{i-2} or e_{i-3})*

The next definition defines two processes U_n and V_n, which are aggregation of events, where the possible successor of an event e_i is either $l_{(indice(e_i)+2)}$ either $l_{(indice(e_i)+3)}$.

Definition 6. *Let us consider that $n <= p$,*

$$\left\{ \begin{array}{l} U_0 = e_1 \\ U_n^1 = l_{n+2}^1 \oplus U_{n+2}^2 \\ U_n^2 = l_{n+3}^1 \oplus U_{n+3}^2 \\ U_n = U_n^1 \ \oplus U_n^2 \end{array} \right. \qquad \left\{ \begin{array}{l} V_0 = e_2 \\ V_n^1 = l_{n+2}^2 \oplus V_{n+2}^2 \\ V_n^2 = l_{n+3}^2 \oplus V_{n+3}^2 \\ V_n = V_n^1 \ \oplus V_n^2 \end{array} \right.$$

U_n: processes beginning by e_1 V_n: processes beginning by e_2 where p is the index of the last event implied in the chain of conflict

Theorem 4. *The canonic form of a chain of conflict C is $U_n \oplus V_n$:*

$$(e_1 \perp\ e_2 \perp\ ...\perp\ e_p) \equiv U_n \oplus V_n$$

Proof. Correctness: let us consider an incorrect process $q \in L_p$:

$$q = (\oplus\ e_{q_1}\ e_{q_2}\ ...\ e_{q_p})$$

An incorrect process contains two event in conflict. Thus, this incorrectness implies the existence of two events in q such as $e_{q_i} \perp e_{q_{i+1}}$ and $e_{q_i}, e_{q_{i+1}}$ corresponding to two successive events of l. This is in contradiction with the definition of the functions $(U_n^1, U_n^2, V_n^1, V_n^2)$ for which events are added with either l_{n+2} either l_{n+3}. For a correct process, indices cannot be consecutive.

Completeness: let us consider a valid process:

$$q = (\oplus\ e_{q_1}\ e_{q_2}\ ...\ e_{q_p})$$

which is not included in L_p. $\forall e \in q$, if q is valid then $\forall (e_i, e_j) \in q, \neg(e_i \perp e_j)$, so it implies that e_i and e_j are not successive in l and every enabled event is in q. Moreover, as q is not included in L_p, thus, it exists at least one couple (e_{q_i}, e_{q_j}), which does not correspond to the construction defined by the functions $(U_n^1, U_n^2, V_n^1, V_n^2)$, which define the possible successor of an event. This means that $indice(e_{q_j}) > indice(e_{q_i} + 3)$.

For every $n = indice(e_{q_j}) - indice(e_{q_i})$ greater than 3, let us note $i_2 = indice(e_{q_i}) + 2$ the event $e_{q_{i_2}}$ is a possible event, which is not in q (contradiction). □

VI. Examples

Examples VI-A and VI-B illustrate conflit equivalence, whereas the example VI-C contains reset arcs.

A. Example 1

Figure 10 gives a Petri net, which represents a chain of conflicts and its unfolding.

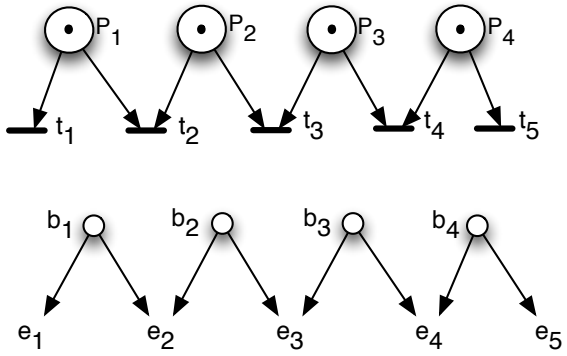


Figure 10. PN and unfolding of a chain of conflicts.

The unfolding gives a table of binary relations on events (see Section IV), which is represented by the following algebraic expression U_2:

$$U_1 = (\oplus\ b_1\ b_2\ b_3\ b_4\ b_5\ (b_1 \prec (e_1 \perp e_2))\ (b_2 \prec (e_2 \perp e_3))\ ...)$$

After some steps of reduction $(MP + S)$, U_1 becomes:

$$(e_1 \perp\ e_2 \perp\ e_3 \perp\ e_4 \perp\ e_5) \tag{13}$$

Theorem 4 allows to compute from (13) its following canonic form:

$$(\perp (\oplus\ e_1\ e_3\ e_5)(\oplus\ e_1\ e_4\ )(\oplus\ e_2\ e_4)(\oplus\ e_2\ e_5))$$

B. Example 2

Let us consider the following Unfolding of Figure 11. The

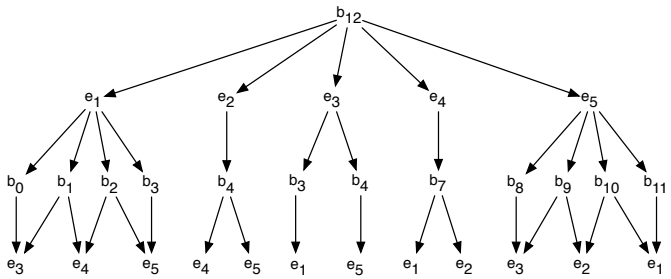


Figure 11. U_2.

table has been computed and the set of binaries relations between events leads to the following algebraic expression U_2:

$$\begin{aligned} U_2 \quad = (\oplus\ b_{12} \quad & (b_{12} \prec (e_1 \perp\ e_2 \perp\ e_3 \perp\ e_4 \perp\ e_5)) \\ & (e_1 \prec (\oplus\ b_0\ b_1\ b_2\ b_3))(e_2 \prec b_4)(e_3 \prec (\oplus b_5\ b_6)) \\ & (e_5 \prec (\oplus b_8\ b_9\ b_{10}\ b_{11}))((\oplus\ b_0\ b_1) \prec e_3) \\ & ((\oplus\ b_1\ b_2) \prec e_4)\ (e_4 \prec b_7)\ ((\oplus\ b_2\ b_3) \prec e_5) \\ & (b_4 \prec (\perp e_4\ e_5))(b_5 \prec e_1)\ (b_6 \prec e_5) \\ & (b_7 \prec (\perp e_1\ e_2))\ ((\oplus\ b_8\ b_9) \prec e_3) \\ & ((\oplus\ b_9\ b_{10}) \prec e_2)\ ((\oplus\ b_{10}\ b_{11}) \prec e_1)) \end{aligned} \tag{14}$$

Let us note P the aggregation of the five first lines of the previous Equation (14) becomes:

$$U_2 \quad = (\oplus\ b_{12} \quad (b_{12} \prec (\perp\ e_1\ e_2\ e_3\ e_4\ e_5))\ P \tag{15}$$

Rules $MP1$, MP_2 and theorem 1 reduce (15) in:

$$\begin{aligned} U_2 \quad = (\perp\ & (\oplus\ e_1\ P)\ (\oplus\ e_2\ P)\ (\oplus\ e_3\ P) \\ & (\oplus\ e_4\ P)\ (\oplus\ e_5\ P)\ ) \end{aligned}$$

Distributivity of $perp$:

$$\begin{aligned} U_2 \quad = (\oplus \quad & (\perp (\oplus\ e_1\ b_0\ b_1\ b_2\ b_3)(\oplus\ e_2\ b_4)(\oplus\ e_3\ b_5\ b_6) \\ & (\oplus\ e_4\ b_7)(\oplus\ e_5\ b_8\ b_9\ b_{10}\ b_{11}))\ ((\oplus\ b_0\ b_1) \prec e_3) \\ & ((\oplus\ b_1\ b_2) \prec e_4)((\oplus\ b_2\ b_3) \prec e_5)\ (b_4 \prec (\perp e_4\ e_5)) \\ & (b_5 \prec e_1)\ (b_6 \prec e_5)(b_7 \prec (\perp e_1\ e_2)) \\ & ((\oplus\ b_8\ b_9) \prec e_3)\ ((\oplus\ b_9\ b_{10}) \prec e_2) \\ & ((\oplus\ b_{10}\ b_{11}) \prec e_1)) \end{aligned}$$

Distributivity of $\perp$ and MP_1:

$$\begin{aligned} U_2 \quad = (\perp \quad & (\oplus\ e_1\ e_3\ e_5\ b_1\ b_2)(\oplus\ e_1\ e_4\ b_0\ b_3)(\oplus\ e_2\ e_4) \\ & (\oplus\ e_2\ e_5)\ (\oplus\ e_3\ e_1)(\oplus\ e_3\ e_5)(\oplus\ e_4\ e_1) \\ & (\oplus\ e_4\ e_2)(\oplus\ e_5\ e_1\ e_3\ b_9\ b_{10})\ (\oplus\ e_5\ e_2\ b_8\ b_{11})) \end{aligned}$$

Theorem 2 : absorption of $(\oplus\ e_3\ e_1)$ and $(\oplus\ e_3\ e_5)$ in $(\oplus\ e_1\ e_3\ e_5\ b_1\ b_2)$, idempotency of $\perp$:

$$\begin{aligned} U_2 \quad = (\perp \quad & (\oplus\ e_1\ e_3\ e_5\ b_1\ b_2)(\oplus\ e_1\ e_4\ b_0\ b_3)(\oplus\ e_2\ e_4) \\ & (\oplus\ e_2\ e_5)\ (\oplus\ e_4\ e_1)(\oplus\ e_5\ e_1\ e_3\ b_9\ b_{10}) \\ & (\oplus\ e_5\ e_2\ b_8\ b_{11})) \end{aligned}$$

Rules of simplification S_1 and S_2 and theorem 2:

$$U_2 = (\perp (\oplus\ e_1\ e_3\ e_5)(\oplus\ e_1\ e_4)(\oplus\ e_2\ e_4)(\oplus\ e_2\ e_5))$$

The two unfoldings of examples 1 and 2 have the same canonic form, they are *conflict-equivalent*: $U_1 \approx_{conf} U_2$

1) Reasoning about processes: Let us consider all the process p of $U_2 : (\oplus\ e_1\ e_3\ e_5), (\oplus\ e_1\ e_4), ...$

- $\forall p \in U_2$ whenever e_3 is present, e_1 is present.
- $\forall p \in U_2, \neg e_3 \perp (e_1 \oplus\ e_3 \oplus e_5)$
  This is the algebraic definition of $\prec$. Finally, from this chain of conflicts, the following causality can be deduced:

$$e_3 \prec (e_1 \oplus e_5) \tag{16}$$

- A similar reasoning can be made:

$$\forall p \in U_2, \neg(e_1 \oplus e_5) \perp (e_1 \oplus\ e_3 \oplus e_5)$$

  This is the algebraic definition of:

$$(e_1 \oplus e_5) \prec e_3 \tag{17}$$

  Equations (16) and (17) express that there is a *strong link* between e_3 and the process $(e_1 \oplus e_5)$ but $\prec$ is no well suited to encompass this relation. These two processes are like "intricated".
- In the same manner:

$$\begin{aligned} & \neg e_2 \perp (e_2 \oplus e_4) \perp (e_2 \oplus e_5) \\ \equiv_{dist}\ & \neg e_2 \perp (e_2 \oplus (e_4 \perp e_5)) \\ \equiv_{def}\ & e_2 \prec (e_4 \perp e_5) \end{aligned} \tag{18}$$

  e_2 leads to a conflict

$$\begin{aligned} & \neg e_1 \perp ((\oplus e_1 e_3 e_5) \perp (e_1 \oplus e_4) \\ \equiv_{dist}\ & \neg e_1 \perp (e_1 \oplus ((e_3 \oplus e_5) \perp e_4)) \\ \equiv_{def}\ & e_1 \prec ((e_3\ \oplus\ e_5) \perp e_4) \end{aligned} \tag{19}$$

Equations (18) and (19) show that e_1 and e_2 transform the chain of conflict in a unique conflict. New relations between events or processes can be introduced:

- Alliance relation: e_1, e_3 and e_5 are in "an alliance relation". Every event of this set is enforced by the occurrence of the other events: $e_1 \oplus e_3$ enforces e_5, $e_1 \oplus e_5$ enforces e_3 and $e_3 \oplus e_5$ enforces e_1.
- Intrication: the occurrence of e_3 forces $e_1 \oplus e_5$ and reciprocally $e_1 \oplus e_5$ forces e_3.
- Resolving conflicts (liberation):
  - e_1 resolves 3 conflicts on 4 (as e_2, e_4 and e_5)
  - e_3 resolves every conflicts.

Semantically, e_3 can be identified as an important event in the chain. Moreover, $(\oplus e_1\ e_3\ e_5)$ is a process aggregated with "associated events". This chain of conflict can be seen as two causalities in conflicts: $(e_1 \prec (e_4 \perp (e_3 \oplus e_5)))\ \perp\ (e_2 \prec (e_4 \perp e_5))$

C. Example 3 (Cash dispenser)

Let us consider a cash dispenser illustrated in Figure 12. The user has three tries (3 tokens are generated in place WaitEnterCode) to enter a valid code ($OKcode$), then he can get $Cash$ or can $Consult$ its account. In this example, a reset arc from $OKCode$ allow to clear the tokens that have not be consumed (for example when the user has entered a valid code at its first or second try) and two reset arcs have been added from $getConsult$ and $getCash$ to clear $ReadyToConsult$ or $ReadyToGetCash$.

It could be useful to prove that if the events $GetCash$ implies that $Okcode$ belongs to the same process.

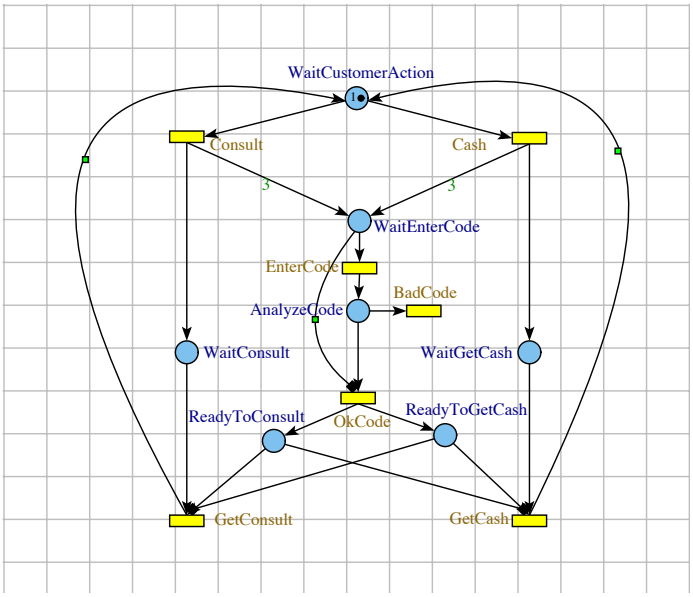


Figure 12. Cash dispenser.

The unfolding of cash dispenser is given in Figure 13. A combinatory inflation of the net is caused by to the reset arcs and by the transitions, $Consult$ and $Cash$, which produces 3 tokens each.

The reset arcs introduces for each events e_9, e_{10}, e_{11}, e_{18}, e_{19}, and e_{20} (events relative the transition $OKcode$) two arcs, which consumes adding conditions. The translation of reset arcs have been defined manually and is not yet implemented in reduction rules. The computing of the canonical form of the processes is following expression:

$U3 = (\perp (\oplus\ Consult\ EnterCode\ OKcode\ GetConsult)$
$(\oplus\ Consult\ EnterCode\ BadCode\ OKcode\ GetConsult)$
$(\oplus\ Consult\ EnterCode\ BadCode\ \ BadCode\ OKcode\ GetConsult)$
$(\oplus\ Consult\ EnterCode\ BadCode\ BadCode\ BadCode)$
$(\oplus\ Cash\ EnterCode\ OKcode\ Getcash)$
$(\oplus\ Cash\ EnterCode\ BadCode\ OKcode\ Getcash)$
$(\oplus\ Cash\ EnterCode\ BadCode\ BadCode\ OKcode\ Getcash)$
$(\oplus\ Cash\ EnterCode\ BadCode\ BadCode\ BadCode)$

This expression formally proves that if $GetCash$ is in a process then $OkCode$ belongs to the same process.

VII. IMPLEMENTATION ASPECTS

A program [19] has been developed. It takes Petri Nets as inputs Romeo [20] unfolds and computes the canonical form. This program has been written in Lisp. The algebraic definitions and the reduction rules has been described with *redex*, a formal package introduced in [11].

A. Syntax of the language

The redex package allows to implement the syntactic rules of the language with an abstract and conceive way:

```
; Nodes                                                    1
   [bool t f]                                              2
   [n variable bool b e (¬ ⊕ n)]                           3
   [e variable (¬ e)]                                      4
   [b variable (¬ b)]                                      5
; n−ary or binary operators                                6
   [on ⊕ ⊥ ≀]                                              7
   [o2 ≺]                                                  8
; Process                                                  9
   [P variable (⊕ Q ... )]                                 10
   [Q variable P n]                                        11
   [C−P (⊕ C−P P) (⊕ P C−P) hole]                          12
;  Conflicts                                               13
   [X variable (⊥ Y ... )]                                 14
   [Y variable X n ]                                       15
   [C−X (⊥ C−X P) (⊥ P C−X) hole]                          16
;  Expression                                              17
   [E variable (on F ...)                                  18
                    (b o2 e) (P o2 X)]                     19
   [F variable E P ]                                       20
   [C−E (on C−E E) (on E C−E)                              21
          (E o2 C−E) (C−E o2 E) hole]                      22
```

- The lines 2 to 5 define the basics nodes, which are boolean, b conditions and e the events. The term *variable* in lines 3 to 5 allows to use in the language every symbols denoted as n_i, b_i or e_i. These symbols are the terminal symbols of the alphabet.
- The lines 7 and 8 group the n-ary and the binary operators.
- Lines 10 to 12 define the process. A process P is constitued with $\oplus$ operator on Q, where Q is defined as a node n or a process P. Every non terminal symbol P_i is a process.
- Lines 14 to 16 define conflicts in a similar way.
- Finally, lines 18 to 22 define expressions that are built from conflicts, process and causality.
- For every term: Process, Conflicts and Expression, contexts are defined. The contexts capture prefixes

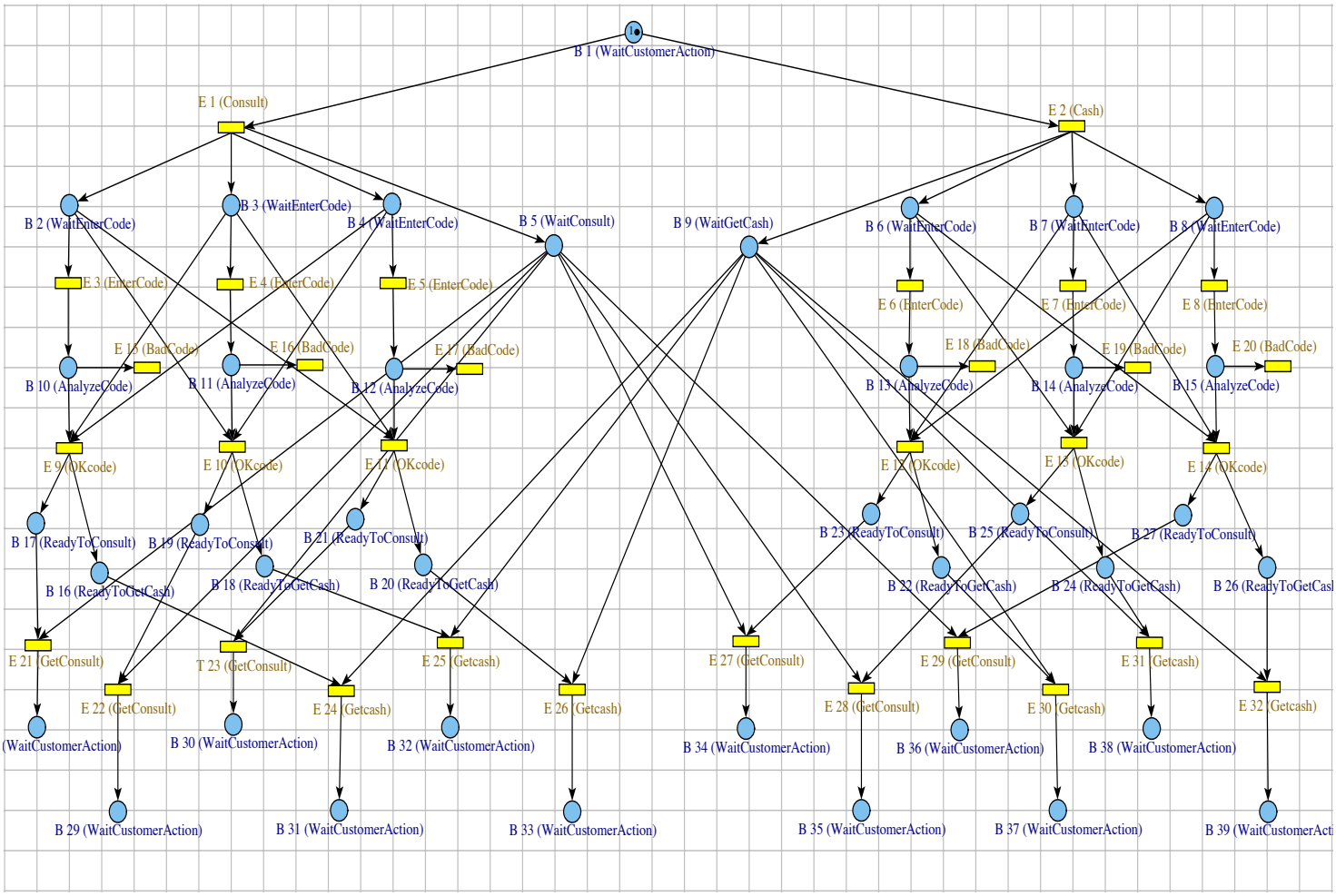


Figure 13. Unfolding of Cash dispenser.

and suffixes of an expression and put them into a hole.

B. Reductions rules

Definitions have been implemented as reduction rules:

```
(-->  (in-hole C-P
        (⊕ Q_1 ... f Q_2 ...))
      (in-hole C-P f)  "A⊕" )
(-->  (in-hole C-E
        (⊕ Q_1 ... e_1 Q_2 ... (¬ e_1) Q_4 ...))
      (in-hole C-E f)  "F⊕" )
```

The particularities of this syntax are:

- Q_i... is equivalent to the regular expression Q_i^*, which represents an ordered list of symbol Q_i, which is eventually empty, finite or infinite.
- The contexts C-P or C-E allows to capture every sub-expression with every prefix and suffixe.

The first rule, labelled $A_{\oplus}$, illustrates that f is an absorbing element. In this rule, C-P captures the context of a Process P and put into a hole. This reduction rule expresses that every sub-expression of the type $(\oplus Q_1 ... f Q_2 ...)$, which can be reduced to the node f. This rule is named and thus, its use can be traced in a future proof.

The second rule $F_{\oplus}$ states the property defined in Section IV-B4 : $e_1 \oplus \neg e_1 \equiv \#f$. This reduction rules defines that every expression (for every context C-E) containing e_1 and $\neg e_1$ in an $\oplus$ operator can be reduced to f.

C. Theorems

This section describes the implementation and the coding of theorems.

1) Theoreme 4: Theorem 4 has been stated from definition 6, which corresponds to the following statements:

```
(define (U1n n 1)
  (if (>= (- (maxi 1) n) 2)
      (cons (list-ref 1 (+ n 2))
            (Rn (+ n 2) 1)) empty))

(define (V2n n 1)
  (if (>= (- (maxi 1) n) 3)
      (cons (list-ref 1 (+ n 3))
            (Rn (+ n 3) 1)) empty))
```

Finally, the implementation is coded like the union of the previous definitions:

```
(define (Rn n l)
  (if (>= (- (maxi l) n) 1)
      (Union (U1n n l) (U2n n l))
      empty))
```

Note that the implementation of the definitions and the theorems are closed to their formal expression.

2) Theoreme 3: $E \perp (E \oplus F) \equiv (E \oplus F)$ has been implemented has a reduction rule:

```
(--> (in-hole C-E
         (⊥ E_1 ... E E_2 ...
              (⊕ E E_3 ...) E_4 ...))
     (in-hole C-E
          (⊥ E_1 ... E_2 ...
               (⊕ E E_3 ... ) E_4 ...)) "T3")
```

This code means that if E is in a "$\perp$ expression:" ($\perp$ E_1... EE_2...), then if a sub expression in $\oplus$ contains E, then E can be suppressed of the "$\perp$ expression" for any context.

VIII. Conclusion and future work

This work is a first attempt to present an axiomatic framework to the analyze of the processes issued of an unfolding. From a set of axioms, distributivities, and derivation rules, theorems have been established and a reduction process can lead to a canonic form. The unfolding process, definitions, theorems, and reduction rules have been coded in LISP[21] with a package named PLT/Redex[11][22]. This canonic form assets an equivalence conflicts ($\equiv_{conf}$) between unfoldings and then Petri nets.

Several perspectives are into progress. First, new theorems have to be established allowing to speed up the procedure of canonic reduction and to extend extraction of knowledge on relationship between events. Different kinds of relationship between events can be defined and formalized: Alliance relation, Intrication, etc. Moreover, as already outlined in the examples, algebraic reasoning can raise semantic informations about events from the canonic form. Another perspective is to extend the approach to Petri nets with inhibitor or drain arcs.

References

[1] d. Delfieu, M. Comlan, and M. Sogbohossou, "Algebraic analysis of branching processes," in *Sixth International Conference on Advances in System Testing and Validation Lifecycle*, 2014, pp. 21–27, best paper award.

[2] J. Esparza and K. Heljanko, "Unfoldings - a partial-order approach to model checking," *EATCS Monographs in Theoretical Computer Science*, 2008.

[3] Engelfriet and Joost, "Branching processes of petri nets," *Acta Informatica*, vol. 28, no. 6, pp. 575–591, 1991.

[4] J. Esparza, S. Römer, and W. Vogler, *An Improvement of McMillan's Unfolding Algorithm*. Mit Press, 1996.

[5] McMillan and Kenneth, "Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits," in *Computer Aided Verification*. Springer, 1993, pp. 164–177.

[6] C. A. R. Hoare, *Communicating sequential processes*. Prentice-hall Englewood Cliffs, 1985, vol. 178.

[7] R. Milner, *Communication and concurrency*. Prentice-hall Englewood Cliffs, 1989.

[8] C. A. Petri, "Communication with automata," *PhD thesis, Institut fuer Instrumentelle Mathematik*, 1962.

[9] R. Glabbeek and F. Vaandrager, "Petri net models for algebraic theories of concurrency," in *PARLE Parallel Architectures and Languages Europe*, ser. Lecture Notes in Computer Science, J. Bakker, A. Nijman, and P. Treleaven, Eds. Springer Berlin Heidelberg, 1987, vol. 259, pp. 224–242.

[10] E. Best, R. Devillers, and M. Koutny, "The box algebra=petri nets+process expressions," *Information and Computation*, vol. 178, no. 1, pp. 44 – 100, 2002.

[11] M. Felleisen, R. Findler, and M. Flatt, *Semantics Engineering With PLT Redex*. Mit Press, 2009.

[12] M. Nielsen, G. Plotkin, and G. Winskel, "Petri nets, event structures and domains," in *T. Theor. Comp. Sci.*, vol. 13(1), 1981, pp. 89–118.

[13] J. Baeten, J. Bergstra, and J. Klop, "An operational semantics for process algebra," in *CWI Report CSR8522*, 1985.

[14] G. Boudol and I. Castellani, "On the semantics of concurrency: partial orders and transitions systems," in *Rapports de Recherche No 550, INRIA, Centre Sophia Antipolis*, 1986.

[15] V. Glaabeek and F. Vaandrager, "Petri nets for algebraic theories of concurrency," in *CWI Report SC-R87*, 1987.

[16] C. Dufourd, P. Jančar, and Ph. Schnoebelen, in *Proceedings of the 26th ICALP'99*, ser. Lecture Notes in Computer Science, J. Wiedermann, P. van Emde Boas, and M. Nielsen, Eds., vol. 1644. Prague, Czech Republic: Springer, Jul. 1999, pp. 301–310.

[17] T. Chatain and C. Jard, "Complete finite prefixes of symbolic unfoldings of safe time petri nets," in *Petri Nets and Other Models of Concurrency - ICATPN 2006*, ser. Lecture Notes in Computer Science, S. Donatelli and P. Thiagarajan, Eds. Springer Berlin Heidelberg, 2006, vol. 4024, pp. 125–145. [Online]. Available: http://dx.doi.org/10.1007/11767589

[18] J.-Y. Girard, "Linear logic," *Theoretical computer science*, vol. 50, no. 1, pp. 1–101, 1987.

[19] D. Delfieu and M. Comlan, "Penelope," http://penelope.rts-software.org/svn, Oct. 2013, tools for editing, unfolding and and to obtain canonical form for Petri Nets.

[20] G. Gardey, D. Lime, M. Magnin *et al.*, "Romeo: A tool for analyzing time petri nets," in *Computer Aided Verification*. Springer Berlin Heidelberg, 2005, pp. 418–423.

[21] G. L. Steele, *Common LISP: the language*. Digital press, 1990.

[22] D. Delfieu and S. Mdssu, "An algebra for branching processes," in *Control, Decision and Information Technologies (CoDIT), 2013 International Conference on*, May 2013, pp. 625–634.

Design and Implementation of Ambient Intelligent Systems using Discrete Event Simulations

Souhila Sehili
University of Corsica
SPE UMR CNRS 6134
Corte, France
sehili@univ-corse.fr

Laurent Capocchi
University of Corsica
SPE UMR CNRS 6134
Corte, France
capocchi@univ-corse.fr

Jean-François Santucci
University of Corsica
SPE UMR CNRS 6134
Corte, France
santucci@univ-corse.fr

***Abstract*—The Internet of Things (IoT) project enables rapid innovation in the area of Internet connected devices and associated cloud services. An IoT node can be defined as a flexible platform for interacting with real world objects and making data about those objects accessible through the Internet. Communication between nodes is discrete Event-oriented and the simulation process play an important role in defining assembly of nodes in such ambient systems. One of today's challenges in the framework of ubiquitous computing concerns the design of such ambient systems. The main problem is to propose a management adapted to the composition of applications in ubiquitous computing. In this paper, we propose the definition of a modeling and simulation scheme based on a discrete-event formalism in order to specify at the very early phase of the design of an ambient system: (i) the behavior of the components involved in the ambient system to be implemented; (ii) the possibility to define a set of strategies that can be implemented in the execution machine. A pedagogical example concerning a concurrent access to a switchable on/off light has been modeled into the Python DEVSimPy environment in order to validate our approach.**

Keywords–IoT; Discrete-event; Simulation; Formalism; Assembly; Smart; Environment.

I. INTRODUCTION

Technological advances in recent years around mobile communication and miniaturization of computer hardware have led to the emergence of ubiquitous computing. In our previous paper [1] we have presented how the DEVS formalism can be used in order simulate the behavior of ambient intelligent components before any implementation using the WComp environment. The interest of this approach has been pointed out on a pedagogical example which allowed to show that using the DEVS formalism conflicts can be detected using simulation before any implementation. The word "ubiquitous computing"was first used in 1988 by Mark Weiser to describe his vision of future [2] - computing at the twenty-first century as he had imagined. In his idea, computing tools are embedded in objects of everyday life. The objects are used both at work and at home. The user has at its disposal a range of small computing devices such as smartphone or PDA, and their use is part of ordinary daily life. These devices make access to information easier for everyone, anywhere and anytime. Users then have the opportunity to exchange data easily, quickly and effortlessly, regardless of their geographic position. The definition of such complex systems involving sensors, smartphones, interconnected objects, computers, etc. results in what is called ambient systems.

One of today's challenges in the framework of ubiquitous computing [3] concerns the design of such ambient systems. One of the main problems is to propose a management adapted to the composition of applications in ubiquitous computing [4]. Ambient systems applications design involves the management of many varied devices integrated in objects of everyday life. The unpredictability of availability of the features of these devices makes the need for explicit adaptation for this type of system. The specificity of this adaptation is that it will meet all the constraints imposed by the context of ambient computing. The difficulty is to propose a compositional adaptation, which aims to integrate new features that were not foreseen in the design, remove or exchange entities that are no longer available in a given context. Mechanism to address this concern must then be proposed by middleware for ubiquitous computing.

We have being focused on the WComp environment, which is a prototyping and dynamic execution environment for Ambient Intelligence applications. WComp [5] is created by the Rainbow research team of the I3S laboratory, hosted by University of Nice - Sophia Antipolis and CNRS. It uses lightweight components to manage dynamic orchestrations of Web service for devices, like UPnP [6] or DPWS services [7], discovered in the software infrastructure. In the framework of the WComp, it has been defined a management mechanism allowing extensible interference between devices. This is particularly important in the context definition of new coordination logic. In WComp it has been proposed a methodology for interference management mechanism to be dynamically and automatically extensible. In order to deal with the asynchronous nature of the real world, WComp has defined an execution machine for complex connections. In a real case, the assumption of zero reaction time is not realistic. It is essential to check that the system is fast enough according to the dynamics of the environment. It is also essential to make the link between the logical time and physical time and the relationship between the actual events of the environment and those used in the definition of synchronous processes [8]. The entity that is responsible for ensuring these approximations is the execution machine and is used to treat the interface between synchronous and asynchronous process environments [9].

In this paper, we propose the definition a modeling and simulation scheme based on the DEVS formalism in order to specify at the very early phase of the design of an ambient system: (i) the behavior of the components involved in the ambient system to be implemented; (ii) the possibility to define

a set of strategies, which can be implemented in the execution machine. The interest of such an approach is twofold: (i) the behavior will be used to write the methods required; (ii) to check the different strategies (to be implemented in the execution machine) before implementation. The rest of the paper is as follows: Section II concerns with the background of the study by presenting the traditional approach for the design of IoT systems. It briefly introduces a set of middleware framework before focusing on the WComp Framework. The DEVS formalism and the DEVSimPy environment are also presented. In Section III, the proposed approach based on the DEVS formalism is given. An overview of the approach as well as the interest in using DEVS simulation is detailed. Section IV deals with the validation of the approach through a case study The conclusion and future work are given in Section V.

II. RELATED WORK

There have been a some approaches dedicated to the management of ubiquitous systems. In this section, we highlight several kinds of middleware tools have been proposed in the recent years such as:

Roman et al. [10] proposed a middleware software infrastructure Gaia, which assist humans in the development of applications for ubiquitous computing buildings and homes intelligent by interacting with devices simultaneously.

Seung et al. [11] proposed a new approach in middleware architecture HOMEROS, which adopts a hybrid-network model to efficiently manage enormous resources, context, location, allowing high flexibility in the environment of heterogeneous devices and users.

Lopes et al. [12] proposed a middleware software infrastructure EXEHDA, which manages and implements the follow-me semantics in which the applications code is installed on-demand on the devices and this installation is adaptive to context of each device.

Ferry et al. [13] proposed a middleware WComp based on a software infrastructure, a service composition architecture, and a compositional adaptation mechanism used in prototyping and executing the Ambient Intelligence applications.

III. BACKGROUND

A. IoT Design and WComp

The ubiquitous computing is a new form of computing that has inspired many works in various fields such as the embedded system, wireless communication, etc. Embedded systems offer computerized systems having sizes smaller and integrated into objects everyday life. An ambient system [14] is a set of physical devices that interact with each other (e.g., a temperature sensor, a connecting lamp, etc.). The design of an ambient system should be based on a software infrastructure and any application to be executed in such an ambient environment must respect the constraints imposed by this software infrastructure.

Devices and software entities provided by the manufacturers are not provided to be changed: they are black boxes. This concept can limit the interactions to use the services they provide and prevents direct access to their implementation. The creation of an ambient system can not under any circumstances pass by a modification of the internal behavior of these entities but simply facilitate the principle reusability, since an entity chooses for its functionality and not its implementation. In the vision of ubiquitous computing, users and devices operate in an environment variable and potentially unpredictable in which the entities involved appear and conveniently disappear (a consequence of mobility, disconnections, breakdowns, etc.). It is not possible to anticipate the application design when we do not have information about availability of any devices. As a result a set of tools have been dedicated in developing software infrastructure allowing the design of applications with the constraint unpredictability availability of component entities [15].

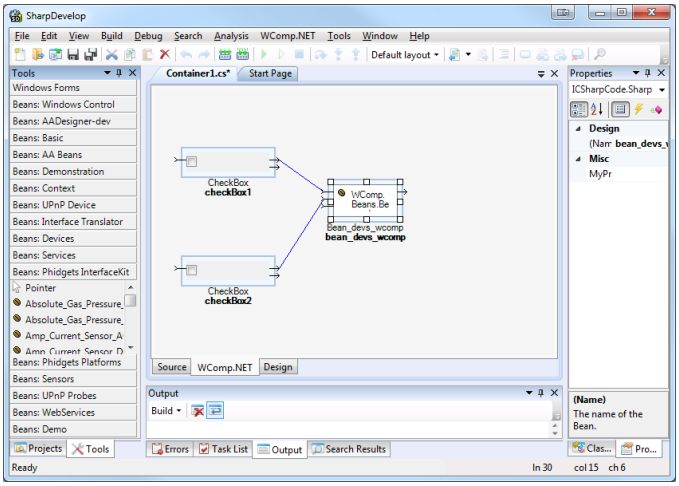


Figure 1. WComp platform.

In this paper, we deal with the WComp framework, which is used in order to design ambient systems. The WComp architecture is organized around containers and designers [16] (Figure 1). The purpose of containers is to take over the management of the dynamic structure such as instantiation, destruction of components and connections.

The Designer runs the Container for instantiation and for the removal of components or connections between components in the Assembly, which has to be created. A component belonging to the WComp platform is an instance of the Bean class implemented in a hight level object language [17] to use properties at runtime and to calibrate some variables to refine the interaction.

An application is created by a WComp component assembly in a container, according to SLCA model [18]. WComp allows to implement an application from an orchestration of services available in the platform and/or other off-the-shelves components.

Whatever the tool that may be used, the design of a IoT component leans on the definition of:

- A set of methods allowing to describe the behaviors of the component
- The execution machine associated with the considered component

The design of ambient computing systems involves a different technique from those used in conventional computing. Applications are designed dynamically by *smart*

devices (assembly components) of different nature.

The smart device is an identified component, which is generalized as a class of objects defining a data as property and containing distinct logic sequences that can manipulate it, known as methods that are executed when the component receives an event from others components. The manner of executing these methods (*state automaton* [19]) depending on some inputs is called the *execution machine* (Figure 2).

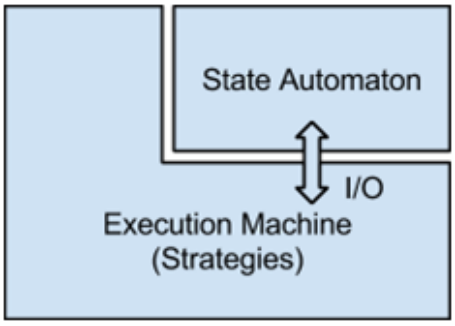


Figure 2. Component state automaton with execution machine.

The construction of an ambient system requires the definition of:

- The state automaton (methods)
- The execution machine

Several ways to manage the execution machine are known as strategies; the description of the strategies are defined manually in the methods of the Bean class (object oriented class) of WComp framework. Figure 3 describes the traditional way to design an ambient system using WComp. The behavior and the components involved in the ambient system, as well as the Bean classes describing the execution machine, are coded using the C# language.

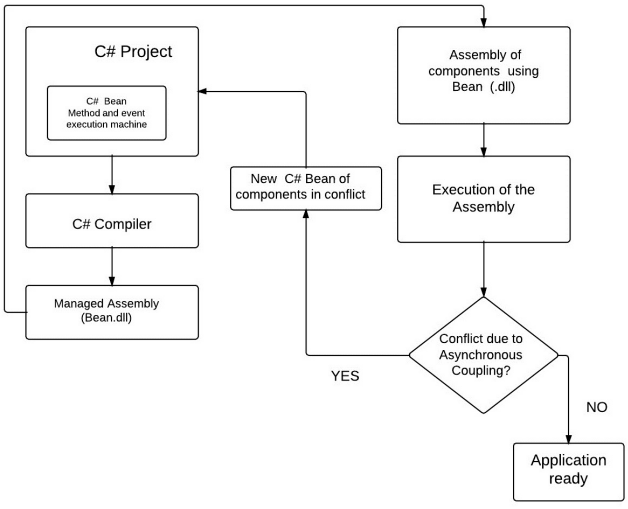


Figure 3. Traditional IoT component design with WComp.

The compilation allows to derive the corresponding dynamic assembling binary files (.dll) of the Bean classes involved in the resulting Assembly [20]. The Assembly can then be executed. Conflicts are checked: if conflicts (generally due to asynchronous couplings) are detected the designer has to write a new behavior of the execution machine by recoding the Bean classes in order to solve the coupling conflicts while if no conflict are detected the application is ready. In this paper, we choose to propose a new approach for a computer aided design of ambient systems using the DEVS formalism by developing DEVS simulation concepts and tools for the WComp platform. The goal is to use the DEVS formalism and the DEVSimPy framework in order to perform DEVS modeling and simulations: (i) to detect the potential conflicts without waiting to implementation and execution phases as in the traditional approach of Figure 3; (ii) to offer the designer to choose between different executions strategies and to test them using DEVs simulations; (iii) to propose a way to automatically generate the coded of the methods involved in the execution machine strategies. The DEVS formalism and the DEVSimPy environment are briefly introduced in the next two sub-sections while the proposed approach is introduced in Section III.

B. The DEVS formalism

Since the seventies, some formal works have been directed in order to develop the theoretical basements for the modeling and simulation of dynamical discrete event systems [21]. DEVS (Discrete EVent system Specification) [22], [23] has been introduced as an abstract formalism for the modeling of discrete event systems, and allows a complete independence from the simulator using the notion of abstract simulator.

DEVS defines two kinds of models: *atomic models* and *coupled models*. An atomic model is a basic model with specifications for the dynamics of the model. It describes the behavior of a component, which is indivisible, in a timed state transition level. Coupled models tell how to couple several component models together to form a new model. This kind of model can be employed as a component in a larger coupled model, thus giving rise to the construction of complex models in a hierarchical fashion. As in general systems theory, a DEVS model contains a set of states and transition functions that are triggered by the simulator.

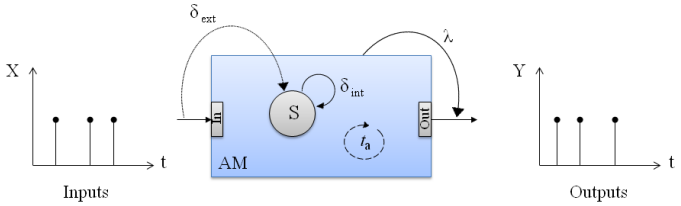


Figure 4. Atomic model in action.

A DEVS atomic model AM (Figure 4) with the behavior is represented by the following structure:

$$AM =< X, Y, S, \delta_{int}, \delta_{ext}, \lambda, t_a >$$

where:

- $X : \{(p, v)|(p \in inputports,\ v \in X_p^h)\}$ is the set of input ports and values,

- $Y : \{(p, v)|(p \in outputports,\ v \in Y_p^h)\}$ is the set of output ports and values,
- S: is the set of states,
- $\delta_{int} : S \rightarrow S$ is the internal transition function that will move the system to the next state after the time returned by the time advance function,
- $\delta_{ext} : Q \times X \rightarrow S$ is the external transition function that will schedule the states changes in reaction to an external input event,
- $\lambda : S \rightarrow Y$ is the output function that will generate external events just before the internal transition takes places,
- $t_a : S \rightarrow R_{\infty}^{+}$ is the time advance function that will give the life time of the current state.

The dynamic interpretation is the following:

- $Q = \{(s, e)|s \in S^h, 0 < e < t_a(s)\}$ is the total state set,
- e is the elapsed time since last transition, and s the partial set of states for the duration of $t_a(s)$ if no external event occur,
- δ_{int} : the model being in a state s at t_i , it will go into s' , $s' = \delta_{int}(s)$, if no external events occurs before $t_i + t_a(s)$,
- δ_{ext} : when an external event occurs, the model being in the state s since the elapsed time e goes in s', The next state depends on the elapsed time in the present state. At every state change, e is reset to 0.
- λ : the output function is executed before an internal transition, before emitting an output event the model remains in a transient state.
- A state with an infinite life time is a passive state *(steady state),* else, it is an active state *(transient state)*. If the state s is passive, the model can evolve only with an input event occurrence.

The DEVS coupled model CM is a structure:

$$CM =< X, Y, D, \{M_d \in D\}, EIC, EOC, IC >$$

where:

- X is the set of input ports for the reception of external events,
- Y is the set of output ports for the emission of external events,
- D is the set of components (coupled or basic models),
- M_d is the DEVS model for each $d \in D$,
- EIC is the set of input links that connects the inputs of the coupled model to one or more of the inputs of the components that it contains,
- EOC is the set of output links that connects the outputs of one or more of the contained components to the output of the coupled model,
- IC is the set of internal links that connects the output ports of the components to the input ports of the components in the coupled models.

In a coupled model, an output port from a model $M_d \in D$ can be connected to the input of another $M_d \in D$ but cannot be connected directly to itself.

The DEVS abstract simulator is derived directly from the model. A simulator is associated with each atomic model and a coordinator is associated with each coupled model. In this approach, simulators allows to control the behavior of each model, and coordinators allows the global synchronization between each of them. The communication between all these elements is performed using four kinds of messages. The initialization messages (i, t) are used to achieve an initial temporal synchronization between all actors. The internal transition messages $(*, t)$ allow the processing of an internal event, while the external transition messages (x, t) allow the processing of an external event. Finally, the output messages (y, t) allow the transportation of the output values to the parent elements and is the result of an $(*, t)$ message.

C. The DEVSimPy environment

DEVSimPy [24] (DEVS Simulator in Python language) is an open Source project (under GPL V.3 license) supported by the SPE team of the university of Corsica Pasquale Paoli. This aim is to provide a GUI for the modeling and simulation of PyDEVS [25] models. PyDEVS is an Application Programming Interface (API) allowing the implementation of the DEVS formalism in Python language. Python is known as an interpreted, very high-level, object-oriented programming language widely used to quickly implement algorithms without focusing on the code debugging [26]. The DEVSimPy environment has been developed in Python with the wxPython [27] graphical library without strong dependences other than the Scipy [28] and the Numpy [29] scientific python libraries. The basic idea behind DEVSimPy is to wrap the PyDEVS API with a GUI allowing significant simplification of handling PyDEVS models (like the coupling between models or their storage).

Figure 5 depicts the general interface of the DEVSimPy environment. A left panel (bag 1 in Figure 5) shows the libraries of DEVSimPy models. The user can instantiate the models by using a drag-and-drop functionality. The bag 2 in Figure 5 shows the modeling part based on a canvas with interconnection of instantiated models. This canvas is a diagram of atomic or coupled DEVS models waiting to be simulate.

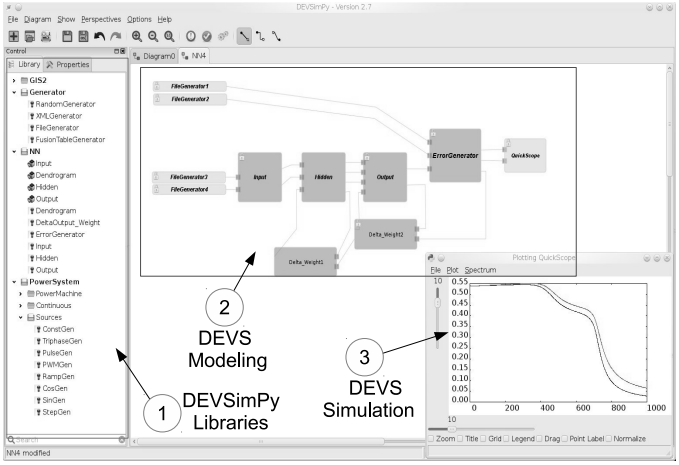


Figure 5. DEVSimPy general interface.

A DEVSimPy model can be stored locally in the hard disk or in cloud through the web in the form of a compressed file including the behavior and the graphical view of the model

separately. The behavior of the model can be extended using specific plug-ins embedded in the DEVSimPy compressed file. This functionality is powerful since it makes it possible to implement new algorithms above the DEVS code of models in order to extend their handling in DEVSimPy (exploit behavioral attributes, overriding of DEVS methods, etc.). A plug-in can also be global in order to manage several models through an generic interface embedded in DEVSimPy. In this case, the general plug-in can be enabled/disabled for a family of selected models. An interesting global plug-in called Blink has been implemented to facilitate the debugging in DEVSimPy. This plug-in is based on successive steps of the simulation and blink the models to indicate their activity with a color code corresponding to the nature of the DEVS transition function (internal, external, time advance, output).

DEVSimPy capitalizes on the intrinsic qualities of DEVS formalism to simulate automatically the models. Simulation is carried out in pressing a simple button, which invokes an error checker before the building of the simulation tree. The simulation algorithm can be selected among hierarchical simulator (default with the DEVS formalism) or direct coupling simulator (most efficient when the model is composed with DEVS coupled models). A plug-in manager is proposed in order to expand the properties of DEVSimPy allowing their enabling/disabling through a dialog window. For example, a plug-in called ”Blink” is proposed to visualize the activity of models during the simulation. It is based on a step by step approach and illuminates each active model with a color, which depends on the executed transition function. In this paper, a plug-in is used to allow the transposition of the execution machine strategies validated with DEVS simulation to WComp environment.

This paper shows how the DEVS formalism is suitable to model synchronous automatons and check the strategies of the execution machine in a context of IoT system design. It also presents the power of WComp to design IoT component based on the strategies defined with DEVSimPy, which is a framework dedicated to DEVS M&S. Furthermore, the strategies defined using DEVSimPy are fully integrated in WComp. The behavior of a DEVS model is expressed through specifications of a finite state automaton. However, this DEVS specifications represent both the state automation and the execution machine. The interest of using DEVS is the ability to define as many strategies as DEVS model specifications. In the following section, background information as the DEVS formalism, DEVSimPy framework and WComp are outlined.

IV. PROPOSED APPROACH

As pointed in Section II-A, the traditional way to design ambient systems described in Figure 3 has the following drawback: the creation of Bean class components using the WComp Platform is performed by the definition of methods (both implementing the behavior of a device and its execution machine) in the object oriented language C#. The compilation allows to obtain a set of library components, which are used in a given Assembly (which corresponds to the designed ambient system). However, eventual conflicts due to the connections involved by the Assembly can be detected only after execution. This means that the Designer has to modify the execution machine of some components and restart the design at the beginning. We propose a quite different way to proceed, which is described in Figure 6.

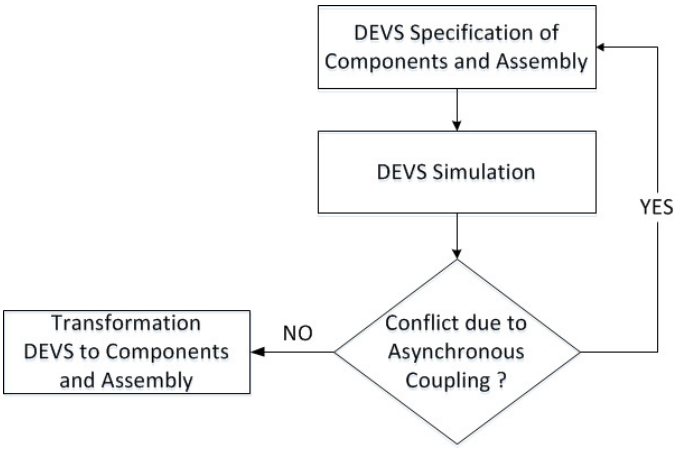


Figure 6. IoT component design using DEVS.

The idea is to use the DEVS formalism in order to help the Designer to:

- Validate different strategies for execution machines involved in an Assembly.
- Write the methods corresponding to the strategy of the execution machine he wants to implement.

For that the Designer has first to write the specifications the components as well as the coupling involved in an Assembly (corresponding to an ambient system to implement). Then simulations can be performed. According to the results of the simulation, conflicts can be highlighted: if some conflicts exist the DEVS specifications have to be modified if not the design process goes on with C# implementation as in Figure 3. The DEVS specifications can be used to help the Designer to write the methods of the Bean classes in the C# language Figure 6 and then compile them and execute the resulting Assembly being assured that there will be no coupling conflict. Section IV details the proposed approach using a pedagogical example. Two different execution machine strategies will be implemented using WComp and using the DEVS formalism. We will point out how DEVS can be used to simulate execution machines strategies before compilation and execution of the C Bean classes. Furthermore, we also point out how the designer can use the DEVS specifications in order to write the methods involved in an execution machine strategy.

V. CASE STUDY: SWITCHABLE ON/OFF LIGHT

A. Description

We choose to validate the proposed approach on a pedagogical case study: realization of an application to control the lighting in a room. The case study involved three components to be assembled: a light component with an input (ON / OFF) and two switches components with an output (ON / OFF) as shown in Figure 7.

Two different behaviors concerning the connections between the switch and the light component are envisioned (corresponding to the implementation of two different execution machines):

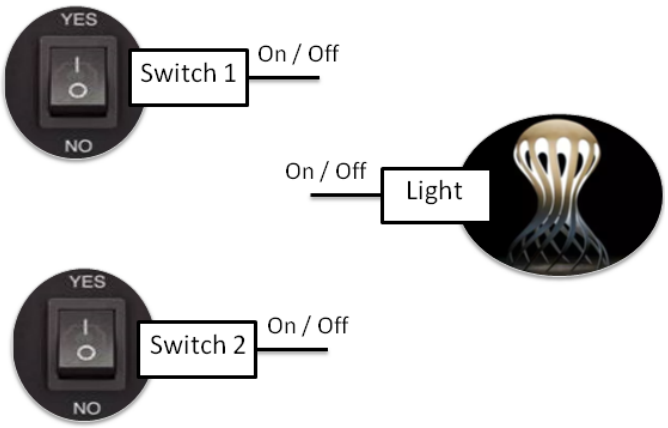


Figure 7. Assembly of Light and Switches components.

- First behavior: the light is controlled by toggle switches, which rest in any of their positions
- Second behavior: the light is controlled by push button switches, which have two-position devices actuated with a button that is pressed and released

In this part, we will present first how we have implemented these two previous behaviors using the WComp platform. Then we will give the DEVS approach involving the DEVS specifications of the two behaviors of the case study and the way DEVS can be used for WComp design of the ambient components.

B. WComp implementation

The behaviors corresponding to the toggle switch and push button switch have been implemented using two different Bean classes in WComp platform in order to be assembled separately with a light component.

The Bean class (Figure 8) in WComp platform is a self-contained class enabling the reuse of the component and facilitate the sharing of it component by other systems. This class is introduced in a specific category of the graphic interface (Container WComp) and the references (#Category in Figure 8) are added in the class. The implementation of the Bean class requires the definition of the name of the Bean class, which is the name of the component in the Designer (#Bean Name in Figure 8). The Properties of the Bean class contain the setter and getter of the class attributes. The Methods implement the behaviors of the component (#Propriety, #Methods in Figure 8) and the EventHandler activate the methods when events are emitted.

Looking at the structure of the Bean class we identify the part that involves a set of actions to follow in a given situation (Methods). These actions define the behaviours of the component that have been identified by the programmer early in the design process.

To illustrate this point, we choose to clarify the observed behaviors by implementing them in the methods of different classes.

```
using System;
using WComp.Beans;
[Bean(Category=" ")]                                   #Category

public class MyBean                                    #Bean Name

public int Myprop {                                    #Propriety
        get {
            return myprop; }
        set {
            myprop = value; }

public void MyMethod() {                               #Methods
        # A method sending an event

        }

public delegate void stringValueEventHandler(string val);

public event stringValueEventHandler PropertyChanged;  #Events

private void MyEvent(int i) {
        if (PropertyChanged != null)
            PropertyChanged(i);
}
```

Figure 8. Bean class structure in WComp.

1) First behavior implementation: corresponding to the toggle switch described in the Figure 9, the line 2 is used to check the position of the toggle switch: if ON is true the line 3 ensures that there are subscribers before calling the event PropertyChanged. In the lines 4 and 5 the event is raised and a resulting string is transmitted. The Bean class returns the String once The ControlMethod method is invoked.

```
1   public void ControlMethod(bool on) {
2   if(on)
3   {if (PropertyChanged != null)
4   PropertyChanged("Light_On");
5   }else{PropertyChanged("Light_Off");}
6   }
```

Figure 9. First light method implementation in WComp.

2) Second behavior implementation: Corresponding to the push button switch described in the Figure 10. The initialization of the lightstate variable of component Light is performed through line 1. Line 3 allows to switch the value of the lightstate variable while line 4 allows to initialize the message to be returned. Line 5 is dedicated to check the lightstate variable and to eventually change to returned message. Lines 6 and 7 allow to ensure that there are subscribers before calling the event Property-Changed and transmit the returned message.

The compilation step is performed for each Bean class. The compiler produces modules that are the traditional executable files (DLL) reusable and manipulated in the WComp platform. After this process each bean class is instantiated and connected with two check-box representing the respective switches in order to realize the required assembly in the WComp platform (Figure 11).

```
1    public bool lightstate = false;
2   public void ControlMethod() {
3      lightstate =! lightstate;
4      string msg = "light_off";
5      if (lightstate) { msg ="light_on";}
6      if (PropertyChanged != null)
7        PropertyChanged(msg);
```

Figure 10. Second light method implementation in WComp.

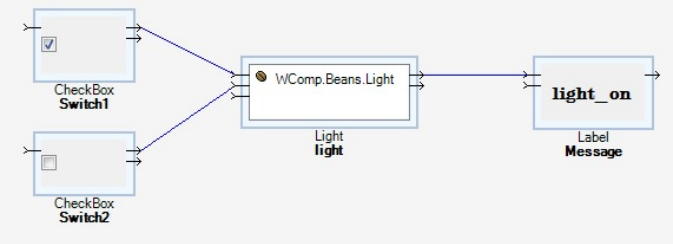


Figure 11. WComp assembly components.

```
Import os
Import platform
Import DomainInterface                          #Specific import

class MyComponent                               #DEVS Class

Def __init__():                                 #Initialization

Def extTransition(self):                        # DEVS external transition function
    pass

Def outFonction(self):                          # DEVS output function
    pass

def intTransition(self):                        # DEVS internal transition function
    pass

def timeAdvance(self):                          # DEVS Time Advance function
    return self.state['sigma']
```

Figure 13. Aotmic Model structure in DEVSimPy.

C. DEVS Specifications

In order to highlight the interest of the DEVS formalism in the management of conflicts between the interconnected components in WComp platform, we defined a DEVS atomic model for each component in DEVSimPy Framework. The behaviors of the light component are implemented in the atomic model Light (Figure 12). The assembly is a DSP diagram (DSP stands for DEVSimPy) and is easy to reuse in DEVSimPy.

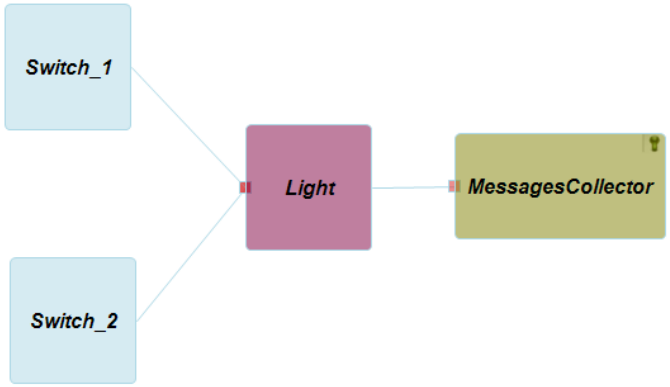


Figure 12. Object interaction diagram for the light component.

Figure 13 depicts the template of an atomic model class in Python language into DEVSimPy.

The implementation of this class needs some specific imports when the model inherits another module or library (#Specific import in Figure 13). The class has a constructor (__init __ ()) with a particular attribute "self.state" that allow to define the state variables (#Initialization in Figure 13). The transition functions like δ_{int} and δ_{ext} are implemented through intTransition(self) and extTransition(self) methods (#DEVS external transition function and #DEVS internal transition function in Figure 13). The output function λ is implemented in the outFunction(self) method and the time advance function t_a in timeAdvance(self) method.

In the structure of the atomic model class, the different actions related to the component behaviors are defined in the external transition that we have chosen to clarify below in the two cases defined in the Section IV-B.

The specifications of the behaviors are achieved using finite-state automaton (Figures 14 and 16) that allows to specify the component behaviors formally [30] and facilitate the deployment in DEVSimPy as an atomic model.

1) The toggle switch behavior: in the transition graph "automaton" given in Figure 14, each state is represented by a pair (state/output). This means that the states are "state1 and state2" and the associated output are "Set_On and Set_Off". The input value is given by the transition between one state and the next state. The system can remain in the same state (loop) as stationary state.

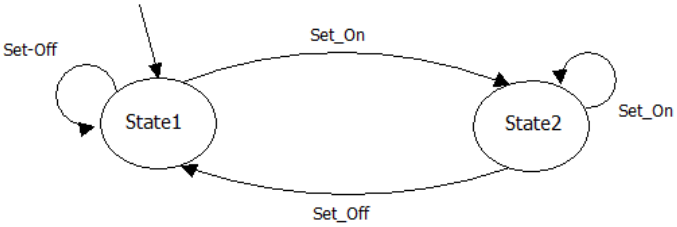


Figure 14. Automaton of the toggle switch behavior.

The corresponding DEVSimPy implementation of the automaton is given in Figure 15 and expressed through the external transition of the atomic model *Light* (Figure 12).

```
1     self.intstate= "OFF"
2     def extTransition(self):
3     for i in xrange(len(self.IPorts)):
4       msg=self.peek(self.IPorts[i])
5       if msg :
6         self.result[i]=msg.value[1]
7       if self.result[i]==self.intstate :
8           self.finstate=self.intstate
9           else:
10          self.finstate=self.result[i]
11            self.state['sigma']=0
```

Figure 15. External transition function of the light atomic model.

The initialization of the state variable *instate* is done in line 1 (initial value is OFF). Line 3 and line 4 allow to assign the variable *msg* with the value of the events on the input ports. From line 5 to line 10 the code allows to assign the value of the state variable *instate* according to the value of the variable *msg*: if the message on the port is equal to the initial state then the state variable remains on the same state else the value of the *instate* variable is changed. By setting the variable *sigma* to 0, line 11 allows to activate the output function.

2) The push-button switch behavior: the transition graph "automaton" given in Figure 16 is a different one that in Figure 14, where the system cannot remain in the same state for each input. The system move to the other state.

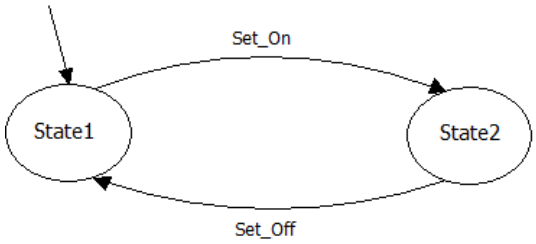


Figure 16. Automaton of the push-button switch behavior.

The corresponding DEVSimPy implementation of the automaton is given in Figure 17 and expressed through the external transition of the atomic model *Light* (Figure 12).

```
1  def extTransition(self):
2      for i in xrange(len(self.IPorts)):
3        msg=self.peek(self.IPorts[i])
4  if msg :
5    self.result[i]=msg.value[1]
6    if self.intstate == "ON":
7    self.finstate= "OFF"
8    else:
9    self.finstate="ON"
10   self.intstate=self.finstate
```

Figure 17. External transition function of the atomic model Light.

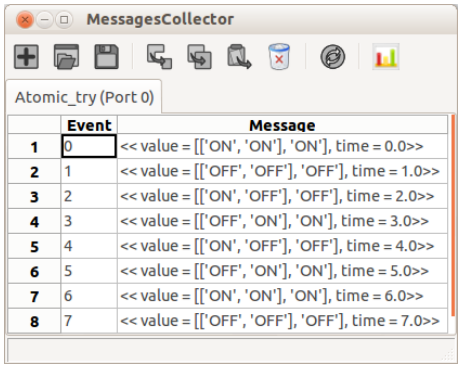
Line 2 and line 3 allow to assign the variable msg with the value of the events on the input ports. From line 4 to line 10, the code allows to switch the value of the state variable *intstate* and *finstate* from ON to OFF or OFF to ON according to the values of the input ports.

D. Simulation results

In both cases, once the modeling scheme has be realized using the DEVSimPy environment, we are able to perform simulations that correspond to the behavior of the ambient system according to the two different execution machines that have been defined. The simulation results obtained with DEVSimPy are illustrated in a *MessageCollector* model, which is often used to store messages received during the simulation. The *MessageCollector* model organizes its results in a table (see Figure 18 and Figure 19).

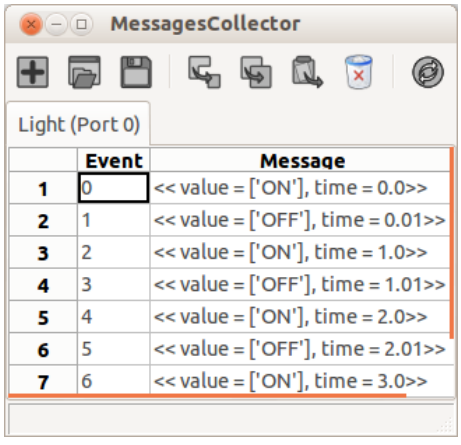
In Figure 18, we show several lines which highlight the result of events from two toggle switches, in the first line we describe the position of toggle switches ['ON', 'ON'] the resulting event is the Lamp 'ON'. The simulation results of the first case express the fact that the execution machine allows the ambient system under study remains in the initial position ("ON" or "OFF") until we will actuate another position using one of the switches.

In Figure 19, we show a several lines which highlight the result of events from two push button switches. The simulation results of the second case express the fact that the execution machine allows the ambient system under study to alternately "ON" and "OFF" with every push of one of the switches.



	Event	Message
1	0	<< value = [['ON', 'ON'], 'ON'], time = 0.0>>
2	1	<< value = [['OFF', 'OFF'], 'OFF'], time = 1.0>>
3	2	<< value = [['ON', 'OFF'], 'OFF'], time = 2.0>>
4	3	<< value = [['OFF', 'ON'], 'ON'], time = 3.0>>
5	4	<< value = [['ON', 'OFF'], 'OFF'], time = 4.0>>
6	5	<< value = [['OFF', 'ON'], 'ON'], time = 5.0>>
7	6	<< value = [['ON', 'ON'], 'ON'], time = 6.0>>
8	7	<< value = [['OFF', 'OFF'], 'OFF'], time = 7.0>>

Figure 18. First simulation results captured with *MessagCollector*.



	Event	Message
1	0	<< value = ['ON'], time = 0.0>>
2	1	<< value = ['OFF'], time = 0.01>>
3	2	<< value = ['ON'], time = 1.0>>
4	3	<< value = ['OFF'], time = 1.01>>
5	4	<< value = ['ON'], time = 2.0>>
6	5	<< value = ['OFF'], time = 2.01>>
7	6	<< value = ['ON'], time = 3.0>>

Figure 19. Second simulation results captured with *MessageCollector*.

E. Integration of DEVS implementation in WComp

As depicted in Figure 20, the integration of strategies in WComp starts by defining the DEVS atomic model (AM) corresponding to the component in which strategies are identified (using functions) in DEVSimPy environment (*Light* in the case study).

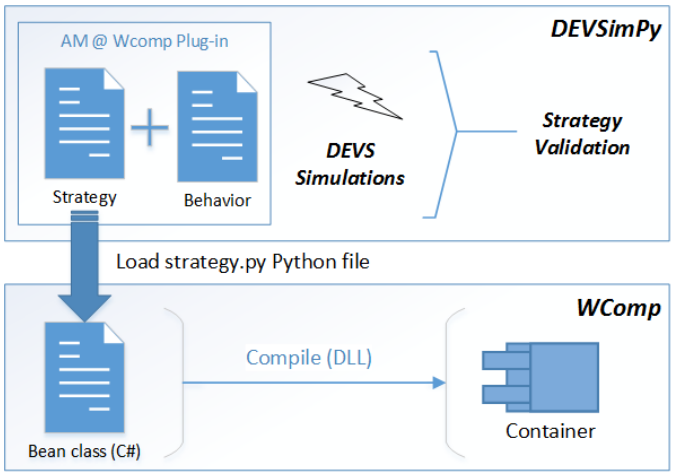


Figure 20. DEVSimPy/WComp integration process.

These strategies will be defined in a dedicated interface from a DEVSimPy local plug-in. The access to the local plug-in will be through the context menu of the atomic model only when the general plug-in called *WComp* of strategies is activated (Figure 21).

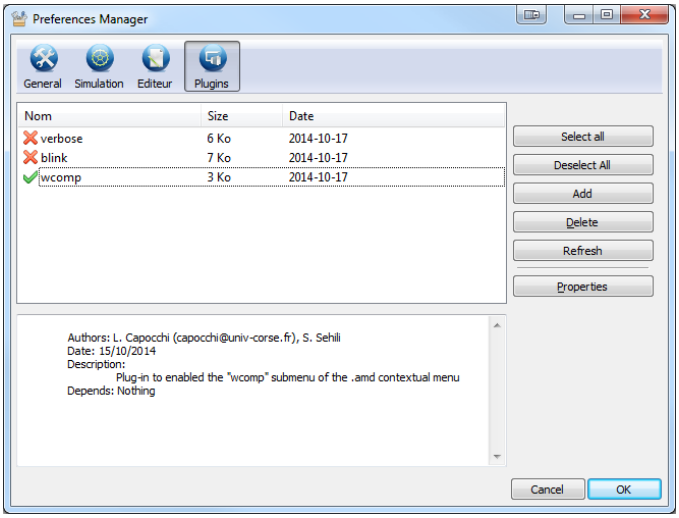


Figure 21. General plug-in *WComp* in DEVSimpy.

Once simulations are performed and strategies are validated in the DEVSimPy framework, we load the strategies file (strategy.py in Figure 20) that contains strategies into WComp. This is done to through IronPython [31], which is an implementation of Python for .NET allowing us to leverage the .NET framework using Python syntax and coding styles.

For that, the class Bean of the component *Light* has been created in WComp and the references (line 1-2) have been added as illustrated in Figure 22 in order to insert Python statements into C# code.

```
1 using IronPython.Hosting;
2 using IronPython.Runtime;
3 using Microsoft.Scripting.Hosting;
4 using Microsoft.CSharp;
```

Figure 22. C# importing to use Python functions.

As illustrated in Figure 23, the IronPython runtime (line 1) and the Dynamic Type (line 2) have been created and the strategy Python file (line 3) has been loaded.

```
1 ScriptRuntime python=Python.CreateRuntime();
2 dynamic pyfile = python.UseFile(@"Path");
3 String string = pyfile.Strategies();
```

Figure 23. C# code to insert Python *Strategies* function.

After the compilation of the Bean class, the corresponding binary file (dll) is inserted in the resulting assembly to be interconnected with other components.

F. Interest of the approach

As described in Sections IV-C and IV-D, the proposed approach allows to study the behavior of an ambient system using DEVS simulations before any WComp implementation. This will allow a Designer of ambient system to select the desired execution machine that adapts to the context of use before the design phase of the component under WComp platform to reduce the time and implementation cost.

In Section IV-C, we briefly introduce how the DEVS specifications can be used by an ambient system Designer to write the code of execution machine. From the two previous cases defined in Section IV-C, we can note that the method of the Bean class under WComp platform of a given ambient component present some similarities with the external transition of the corresponding DEVS atomic model of the same component (in the one part, see Figure 9 and Figure 15, and on the other part Figure 10 and Figure 17).

Furthermore, in Section IV-E, we performed simulations of strategies defined in the local plug-in of the atomic model of the component in DEVSimPy that are loaded in WComp framework through an implementation of python for .NET (IronPython) in the Bean class. This will allow us to validate and implement all the components, which WComp platform reuse them directly.

VI. Conclusion and Future Works

This paper deals with an approach for the design and the implementation of IoT ambient systems based on Discrete Event Modelling and Simulation. The traditional way leans on: (i) the definition of the behavior of IoT components in a Library; (ii) the design of the coupling of components belonging to the Library; (iii) the execution of the resulting coupling. If some errors are detected, the designer has to redefine the behavior of the components (especially by redefining the behavior of the execution machine, which allows to

describe the behavior of the ambient system in case of time conflicts).

This paper introduces a new approach based on DEVS simulations: instead of waiting the implementation phase to detect eventual conflicts, we propose an initial phase consisting in DEVS modeling and simulation of the behavior of components involved in an ambient system, as well as the behavior of execution machines. Once the DEVS simulations have brought successful results, the Designer can implement the behavior of the given ambient system using an IoT framework such as WComp. The presented approach has been applied on a pedagogical example that is described in detail in the paper: implementation of two different behaviors of a given ambient system, definition of the corresponding DEVS specification, implementation of the DEVS behavior using the DEVSimPy framework, analysis of the simulation results. Furthermore, we have also pointed out that the DEVS specifications can be used in order to help the Designer to write the behavior of the IoT components.

Our future work will consist in two main directions. Firstly, we have to work on the Design of complex IoT systems using DEVS formalism and DEVSimPy framework. Secondly, we have to propose an approach allowing to automatically write the behavior of the execution machines after their validation based on DEVS simulation. This automatic generation of the behavior will be performed from the DEVS external state transition function coding and will consist in generating the corresponding execution machine code (for example C# code in the case of the WComp framework).

References

[1] S. Sehili, L. Capocchi, and J.-F. Santucci, "Iot component design and implementation using devs simulations," in The Sixth International Conference on Advances in System Simulation (SIMUL), 2014, pp. 71–76.

[2] M. Weiser, "The computer for the 21st century," SIGMOBILE Mob. Comput. Commun. Rev., vol. 3, no. 3, Jul. 1999, pp. 3–11. [Online]. Available: http://doi.acm.org/10.1145/329124.329126

[3] M. Satyanarayanan, "Pervasive computing: Vision and challenges," IEEE Personal Communications, vol. 8, 2001, pp. 10–17.

[4] M. Zhao, G. Privat, E. Rutten, and H. Alla, "Discrete control for the internet of things and smart environments," in Presented as part of the 8th International Workshop on Feedback Computing. Berkeley, CA: USENIX, 2013. [Online]. Available: https://www.usenix.org/conference/feedbackcomputing13/workshop-program/presentation/Zhao

[5] V. Hourdin, N. Ferry, J.-Y. Tigli, S. Lavirotte, and G. Rey, "Middleware in ubiquitous computing," Computer Science and Ambient Intelligence, 2013, pp. 71–88.

[6] M. Jeronimo and J. Weast, "Upnp design by example: A software developer's guide to universal plug and play," in Intel Press, 2003.

[7] S. Unger, E. Zeeb, F. Golatowski, D. Timmermann, and H. Grandy, "Extending the devices profile for web services for secure mobile device communication," in Presented as part of the 8th International Workshop on Feedback Computing, 2013.

[8] A. Benveniste and G. Berry, "The synchronous approach to reactive and real-time systems," Proceedings of the IEEE, vol. 79, no. 9, 1991, pp. 1270–1282.

[9] S. Schewe and B. Finkbeiner, "Synthesis of asynchronous systems," in 16th International Symposium on Logic Based Program Synthesis and Transformation (LOPSTR 2006). Springer Verlag, 2006, pp. 127–142.

[10] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "Gaia: A middleware platform for active spaces," SIGMOBILE Mob. Comput. Commun. Rev., vol. 6, no. 4, Oct. 2002, pp. 65–67. [Online]. Available: http://doi.acm.org/10.1145/643550.643558

[11] S. W. Han, Y. B. Yoon, H. Y. Youn, and W.-D. Cho, "A new middleware architecture for ubiquitous computing environment," in Software Technologies for Future Embedded and Ubiquitous Systems, 2004. Proceedings. Second IEEE Workshop on. IEEE, 2004, pp. 117–121.

[12] J. Lopes, R. Souza, C. Geyer, C. Costa, J. Barbosa, A. Pernas, and A. Yamin, "A middleware architecture for dynamic adaptation in ubiquitous computing," Journal of Universal Computer Science, vol. 20, no. 9, 2014, pp. 1327–1351.

[13] N. Ferry, V. Hourdin, S. Lavirotte, G. Rey, M. Riveill, and J.-Y. Tigli, "WComp, a Middleware for Ubiquitous Computing", ser. . InTech, Feb. 2011, ch. 8, pp. 151–176. [Online]. Available: http://www.intechopen.com/articles/show/title/wcomp-a-middleware-for-ubiquitous-computing

[14] Y. Liu, "Design of the smart home based on embedded system," in Computer-Aided Industrial Design and Conceptual Design, 2006. CAIDCD'06. 7th International Conference on. IEEE, 2006, pp. 1–3.

[15] D. Cheung-Foo-Wo, "Adaptation dynamique par tissage d'aspects d'assemblage," Ph.D. dissertation, Université de Nice Sophia Antipolis, 2009.

[16] V. Monfort and F. Felhi, "Context aware management plateform to invoke remote or local e learning services: Application to navigation and fishing simulator," in Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence (ISAmI 2010). Springer, 2010, pp. 157–165.

[17] G. Gauffre, S. Charfi, C. Bortolaso, C. Bach, and E. Dubois, "Developing mixed interactive systems: A model-based process for generating and managing design solutions," in The Engineering of Mixed Reality Systems. Springer, 2010, pp. 183–208.

[18] V. Hourdin, J.-Y. Tigli, S. Lavirotte, G. Rey, and M. Riveill, "Slca, composite services for ubiquitous computing," in Proceedings of the International Conference on Mobile Technology, Applications, and Systems. ACM, 2008, p. 11.

[19] S. Eugene Xavier, "Theory of automata formal languages and computation," in The Engineering of Mixed Reality Systems. New Age International (P) Ltd, 2005, ISBN: 978-81-224-2334-1.

[20] V. Monfort and S. Cherif, "Bridging the gap between technical heterogeneity of context-aware platforms: Experimenting a service based connectivity between adaptable android, wcomp and openorb," in IJCSI International Journal of Computer Science Issues, vol. 8, no. 3. IJCSI, May 2011, ISBN: 978-81-224-2334-1.

[21] B. P. Zeigler, "An introduction to set theory,," ACIMS Laboratory, University of Arizona, Tech. Rep., 2003, URL: http://www.acims.arizona.edu/EDUCATION/ [Retrieved: April, 2014].

[22] B. P. Zeigler, H. Praehofer, and T. G. Kim, Theory of Modeling and Simulation, Second Edition,. Academic Press, 2000.

[23] B. Zeigler and H. Sarjoughian, "System entity structure basics," in Guide to Modeling and Simulation of Systems of Systems, ser. Simulation Foundations, Methods and Applications. Springer London, 2013, pp. 27–37. [Online]. Available: http://dx.doi.org/10.1007/978-0-85729-865-2_3

[24] L. Capocchi, J. F. Santucci, B. Poggi, and C. Nicolai, "DEVSimPy: A Collaborative Python Software for Modeling and Simulation of DEVS Systems,," in WETICE. IEEE Computer Society, 2011, pp. 170–175, URL: http://code.google.com/p/devsimpy/ [Retrieved: Dec 2014].

[25] X. Li, H. Vangheluwe, Y. Lei, H. Song, and W. Wang, "A testing framework for devs formalism implementations," in Proceedings on the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, ser. TMS-DEVS '11. San Diego, CA, USA: Society for Computer Simulation International, 2011, pp. 183–188.

[26] F. Perez, B. E. Granger, and J. D. Hunter, "Python: An ecosystem for scientific computing," Computing in Science and Engineering, vol. 13, no. 2, 2011, pp. 13–21, URL: http://dblp.uni-trier.de/db/journals/cse/cse13.html#PerezGH11 [Retrieved: February, 2014].

[27] N. Rappin and R. Dunn, "Wxpython in action." Greenwich, Conn: Manning, 2006.

[28] E. Jones, T. Oliphant, P. Peterson et al., "SciPy: Open source scientific tools for Python," 2001, [accessed 2015-05-26]. [Online]. Available: http://www.scipy.org/

[29] T. E. Oliphant, “Python for scientific computing,” Computing in Science and Engineering, vol. 9, no. 3, May/June 2007.

[30] N. Belloir, J.-M. Bruel, and F. Barbier, “Intégration du test dans les composants logiciels,” in Workshop OCM dans lingnierie des SI during INFORSID, 2002.

[31] M. Foord and C. Muirhead, “Ironpython in action,” in Manning Publications Co., 2009.

Advances in SAN Coverage Architectural Modeling

Trace coverage, modeling, and analysis across IBM systems test labs world-wide

Tara Astigarraga
IBM CHQ
Rochester, NY
United States
asti@us.ibm.com

Yoram Adler and Orna Raz
IBM Research
Haifa, Israel
adler@il.ibm.com
ornar@il.ibm.com

Robin Elaiho and Sheri Jackson
IBM Systems
Tucson, AZ
United States
rlelaiho@us.ibm.com
sheribj@us.ibm.com

Jose Roberto Mosqueda Mejia
IBM Systems
Guadalajara, Mexico
mosqueda@mx1.ibm.com

Abstract - **Storage Area Networks (SAN) architectural solutions are highly complex, often with enterprise class quality requirements. To perform end-to-end customer-like SAN testing, multiple complex interoperability test labs are necessary. One key factor in field quality is test coverage; in distributed test environments this requires a centralized view and coverage model across the different areas of test. We define centralized coverage models and apply our novel trace coverage technology to automatically populate these models. Early results indicate that we are able to create a centralized view of SAN architectural coverage across the multitude of IBM test labs world-wide. Moreover, we are able to compare test lab coverage models with customer environments. Since its inception, this distance matrix project has shown added value in many foreseen and unforeseen ways. The largest benefit of this project is the ability to systematically extract and model coverage across a large number of test and client SAN environments, enabling increased coverage without expanding resource requirements or timelines. One of the key success factors for this model is its scalability. The scalability and reach of the distance matrix project has also uncovered additional unforeseen benefits and efficiencies. As the project matures we continue to see improvements, new capabilities, use case extensions and scaled architectural coverage advances.**

Keywords - Software Test; SAN Coverage; SAN Architecture Coverage; SAN Architectural Modeling; Software Engineering; SAN Test; System Test; Distance Matrix; Trace Coverage Models; SAN Hardware Test Coverage; Test Coverage Analysis; IBM Test; IBM Systems Test.

I. INTRODUCTION AND MOTIVATION

This article is an updated and extended version of a work-in-progress report that was presented and published at the VALID 2014 conference in Nice, France [1].

IBM is a global technology and innovation company with more than 400,000 employees serving clients in 170 countries [2]. The IBM test structure consists of thousands of test engineers world-wide. In addition to function test teams for product streams, there is also an entire world-wide organization of many hundreds of people dedicated to systems and solution test. IBM has interoperability and complex test labs world-wide [3]. Systems test strategies focus on customer-like, end-to-end solution integration testing designed to cover the architectural design points of a broad range of customer environments and operations with the end goal of increased early discovery of high-impact defects, resulting in increased quality solutions. One key area of systems and solution test is innovation. As configurations supported continue to climb, with over 237 million configurations supported on the IBM System Storage Interoperation Center (SSIC) site, test engineers are continually challenged to find ways to test smarter [4]. As part of ongoing test cycles test engineers are continually updating their environments to best represent ever changing technologies, configurations, architectures and integrated technologies and virtualization layers in the server, storage and network environments. In order to keep pace with technology demands test engineers are expected to perform integrated systems planning and recommend new technologies, techniques or automation that will enhance current systems test coverage and support the larger goal of optimized test coverage and minimized field incidents.

One IBM test transformational project we have been working on is the storage area network (SAN) distance matrix project. This project arose from the IBM Test and Research groups as a joint-project aimed at better quantifying and understanding the systems test SAN coverage across IBM test groups world-wide [1]. The project emerged from IBM systems test as a set of requirements and early vision of automated capabilities for SAN coverage modeling. In partnership with the IBM Haifa Research lab we formed a small working team and began to document, model and prototype innovative solutions. At the start of this project we had many questions related to world-wide hardware and SAN coverage, but we did not have a centralized view of the test labs across IBM. Test labs were designed, built, monitored and architected on an individual basis without the ability to easily extract coverage models across the test locations and understand on a global scale the combined IBM test coverage model. Another missing piece was the ability to do broad coverage reviews looking at IBM test labs in comparison to its clients. We have always worked hard to build our test environments to include key characteristics from a diverse range of IBM clients, however, we lacked data environment modeling tools to take customer environment variables and systematically map them against our test environments. The IBM distance matrix project was designed to address these concerns and help to centralize visibility and configuration

details about the systems and solution SAN test labs across IBM and its clients.

The SAN distance matrix project has the abilities to look at key architectural design points across the SAN environments and extract coverage summaries for deep-dive reviews, comparisons and ultimately architecture changes to continually improve our solution test coverage, scalability and customer focus.

In this paper, we will further describe the SAN distance matrix project goals, methods, and advancements achieved within the following sections: Section II. Related Work, Section III. Project Strategy, Section IV. Collecting Data, Section V. Analyzing the Data, Section VI. Early Results, Section VII. Additional Benefits Realized, and Section VIII. Conclusion and Further Development.

II. RELATED WORK

The SAN distance matrix project provides a means for better quantifying and understanding the SAN coverage over the entire test organization, across its different test groups. The same solution also provides the ability to do broad coverage reviews looking at an organization test labs in comparison to its clients. No existing technology that we are aware of provides that.

There are existing tools, including Cisco Data Center Network Manager [5] and Brocade Network Advisor [6] that provide in-depth and detailed modeling capabilities for single environments or environments managed by a single entity; however, there is a gap in the ability to easily look across a heterogeneous group of environments controlled by different companies, divisions or organizations.

There are other tools that can be used to get a consolidated view of the status and performance of your storage and network devices, including SolarWinds Storage Manager [7] and IBM Tivoli Monitoring [8], however, there is a gap in the ability to define new values or parameters to be monitored and generate reports across environments controlled by different companies, divisions or organizations. Additionally, these tools are not developed with the purpose of comparing coverage and architectural models across environments.

In our solution, we deploy the novel idea of trace coverage, relying on the extraction of a functional model from existing switch dump data. In functional modeling and one of its optimization techniques Combinatorial Test Design (CTD), the system under test is modeled as a set of parameters, respective values, and restrictions on value combinations that may not appear together in a test. A test in this setting is a tuple in which every parameter gets a single value. A combinatorial algorithm is applied in order to come up with a test plan (a set of tests) that covers all required interactions between parameters. Kuhn, Wallace and Gallo [9] conducted an empirical study on the interactions that cause faults in software that is the basis for the rationale behind CTD. Nie and Leung [10] provide a recent survey on CTD. The SAN distance matrix that we create can be viewed as a functional model. This functional model could be optimized with tools such as IBM Functional Coverage Unified Solution (IBM FOCUS) [11, 12]. In our case, we automatically extract the model from switch dumps. We term the creation of coverage models from existing traces 'trace coverage'.

III. PROJECT STRATEGY

The SAN distance matrix project strategy is composed of two main phases as shown in Figure 1. Phase 1 consists of collecting switch dump data; a scripted process to extract key data across multiple SAN environments. By identifying key switch data, the script we execute has little impact to the regular activity of the switches. Test team members, with expertise in configuring complex SAN solutions, and in-depth knowledge of best practices and supported configurations, identified the set of switch commands to collect the data required for phase 2 of the project. While the initial set of commands executed were chosen carefully we also built the project structure and scripting capabilities with the assumption that the list of commands executed will likely grow, change and expand with time and project maturity.

Phase 2 consists of analyzing the collected switch dump data. Within this phase, hundreds of switch dump data files from various test and customer labs collected in phase 1 environments were analyzed and parsed into a structured format that would aid in our comparison, analysis and reporting of the collected data.

The SAN distance matrix project is currently extracting data quarterly across teams world-wide. While we chose to implement an ongoing quarterly collection cycle, we also have the capabilities to kick-off a collection stream at any time should the need for new or specified data emerge from any given lab or combination of labs across IBM. In the following sections, we describe each phase and activities in detail

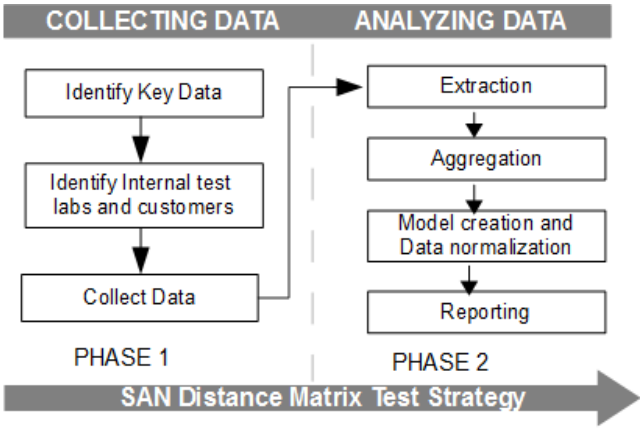


Figure 1. SAN Distance Matrix Project Strategy

IV. COLLECTING DATA

The data collection phase is composed of 3 main activities:

A. *Identify Key Data*

Using switch dump data, we've selected specific switch query commands, which are used to systematically extract the key data for usage and coverage statistics across different IBM test teams and select customers. The switch query commands allow us to extract dump data focused on topologies, coverage points, performance, utilization and other environmental aspects in our SANs. Topology data points include port speeds, port counts and port types. Environmental data points include the switch hardware platforms, protocols used such as Fibre Channel (FC), Fibre Channel over Ethernet (FCoE) and

Fibre Channel over IP (FCIP), code levels, switch up-time and switch special functions/features that are enabled.

Architectural design points include port-channel/trunk usage, virtual storage area network or virtual local area network (VSAN/VLAN) coverage, virtualization data and initiator/target to inter-switch link ratios. Using this raw dump data and subsequent processing logic, we were able to create a summary of all the different port speeds being tested, switch utilization rates, general architecture modeling and software and hardware versions being covered across the initial scope of IBM systems test and customer environments. Additional insights of interest that were identified via analysis of key data include host to storage ratios, host and storage to ISL ratios, architectural design complexity and port and bandwidth utilization rates.

This approach enabled us to easily gather promising data, avoid limitations of manual investigation and create a model that is scalable and easy to use for ongoing analysis. Further, the data structure and quarterly data pulls provided us with results and data that we are then able to use in compiling trending reports and pattern discovery across IBM test labs world-wide

B. Identify internal test labs and customers

The initial IBM test teams added to the project scope were selected based on our team's previous connections and working relationships with the different IBM systems test labs. We had an introductory meeting with several teams that covered the objectives, process and benefits of the SAN distance matrix project. Participation at the early stages of this project was voluntary. As the project progressed and initial results were reviewed with vested management teams the scope was expanded to include a broader list of test labs across systems test and even to include select function test labs.

The process to select customers and include them in the project was different. Since we do not have direct access into customer labs, we looked into different options. One option we chose was to leverage existing client relationships and the IBM customer advocate program to invite customers to submit data for use in this project. Clients who submitted data were incented to do so with the goal of better environment understanding and future IBM test coverage models built utilizing their environment architecture as a piece of the modeling puzzle for future test cycles. Additionally, we reached out to the IBM SAN support organization and requested interlock capabilities to allow selected client dump data be utilized for modeling capabilities for the distance matrix project. These two avenues have been successful in the early stages of the project and we continue to look for ways to systematically expand the number of clients we are able to include. The goal is to ensure that the customer data sets we receive and leverage are balanced across industry, company size, scale, and environment complexity. Although we are not able to replicate and test every environment data set we receive, the distance matrix project allows us to extract key data points and ensure those combined client data points are used as coverage requirements in upcoming test cycles.

C. Collect Data

For data collection within internal IBM test labs, we designed automated scripts to collect the dumps and command query data. The scripts use a source comma separated values (CSV) file, which contains the list of switches, switch types, IPs and credentials. It uses a telnet connection to login the different switches, then executes the appropriate switch query commands and generates a log containing the switch dump data for each switch. The series of commands run in the background and are non-disruptive to the test lab's switch fabric. For the initial scope of this project a subset of IBM test labs was chosen. That subset group included fourteen IBM system test labs, which contained a combined total of four hundred and eighty five SAN top of rack edge and core switches. The output from the fourteen test labs is raw data that consists of a text file for each of the four hundred and eighty five switches that need consolidation and further formatting of the pertinent information for use in the project.

For data collection at customer locations, we do not execute any command in the customer's environment. We instead ask them to send in a switch support dump or specified command query output depending on the brand of switches deployed in the customer environments. The dump information supplied by customers is similar in nature to the data we collected internally, and will also need further formatting during analysis of the data.

V. ANALYZING THE DATA

The problem: SAN switch dump data is heterogeneous based on switch vendor, platform and code levels. Further, the data is collected from various sources and unique collection methods across IBM test labs and customer locations.

The switch dump data is a text file created for each switch. It contains output from multiple switch queries/commands that are executed against the switch. Each switch type has its own set of commands and a unique output format.

The goal: Parse the various switch dump semi structured data and transfer it to structured format.

The solution: the solution relies on the novel notion of trace coverage and the IBM EASER [13] easy log search tool.

Trace coverage extracts report data from traces that already exist in a system or are easy to create according to a defined coverage model. The coverage model can be code coverage – automatically created from the code locations that emit trace data, or functional coverage – manually created to define the system configuration or behavior. In SAN coverage, the traces are created by switch dumps, and the coverage model is a functional coverage of the possible SAN environments. A functional coverage model describes the test space in terms of variation points or attributes and their values. For example, attributes may be port types, port rates, or port utilization percentages. The IBM EASER tool supports extraction of semi-structured data from traces and transforms it into a structured format. It provides both a graphical user interface (GUI) for interactive exploration and a headless mode of operation for automating the extraction and analysis process.

After defining a functional coverage model, the IBM EASER tool is used to extract, aggregate, and compare data:

- Extract functional model values from switch dumps
- Aggregate the coverage of multiple logs from both customers and IBM test labs.
- Compare coverage between a defined set and subsets of labs by generating multiple summary reports.

The SAN Test functional coverage model is extendable; it can be updated to include additional values seen in customer environments. The collected data is aggregated by IBM test groups and customers and definitions are flexible and can be supplied by the end-user.

The automated functional coverage analysis process includes three phases: Extraction, Aggregation and Reporting.

A. Extraction

The functional model attributes' values are extracted from each switch dump file. By using EASER, the log is divided into entries and then the relevant data is extracted, computed and inserted into the relevant model attributes' values. One file with attributes and values is created for each switch log file.

Figure 2 and Figure 4 are excerpts from the original switch dump file, while Figure 3 and Figure 5 are a result of the various stages of our analysis.

Figure 2 shows a sample of a single cisco_fc switch dump data log file, which is created using the automated scripts. In addition to the switch summary, the log file includes the switch query commands and corresponding switch data output. The figure shows a single entry out of the entire switch dump. This is achieved via the EASER parser through its support for smart data partitioning.

```
2014-03-24 14:24:55 INFO Switch Summary
   Name:     slswc10f2cis
   IPAddr:   9.11.195.75
   Brand:    cisco
   Type:     fc
   Area:     cisco san
   Location: tucson
2014-03-24 14:24:55 INFO Log in to  device slswc10f2cis.tuc.stglabs.ibm.com
2014-03-24 14:25:00 INFO Log in to slswc10f2cis.tuc.stglabs.ibm.com successful
2014-03-24 14:25:00 INFO ------------------------------------------------
```

Figure 2. Switch Log File Sample

The EASER parser extracts values from the entry in Figure 2 and updates them into the attributes shown in Figure 3. Figure 3 shows for each category (column header)
it's extracted value. For example, the SwitchType() category has the value cisco_fc, which was extracted from the original switch dump entry, shown in Figure 2.

Area	Locatio	SwitchType	Name	Valu	Name1	Value2
test-team1	tucson	brocade_fc	TotalSwitchWithDataCount	51	SwitchesNames	(XXXXXX
test-team1	tucson	cisco_fc	TotalSwitchWithDataCount	6	SwitchesNames	(XXXXXX
test-team1	tucson	cisco_fcoe	TotalSwitchWithDataCount	8	SwitchesNames	(XXXXXX
test-team1	tucson	brocade_fcoe	TotalSwitchWithDataCount	4	SwitchesNames	(XXXXXX
test-team2	hursley	brocade_fc	TotalSwitchWithDataCount	100	SwitchesNames	(XXXXXX
test-team2	hursley	cisco_fc	TotalSwitchWithDataCount	4	SwitchesNames	(XXXXXX

Figure 3. Parser extracted data Sample

Figure 4 shows a sample of an entry in a Cisco switch dump data extract, as extracted by the EASER parser. The EASER parser then uses this data to compute category summary values. These values become part of the distance model, as shown in Figure 5.

```
--------------------------------------------------------------------------------
Interface  Vsan   Admin  Admin   Status        SFP    Oper  Oper   Port
                  Mode   Trunk                       Mode  Speed  Channel
                         Mode                               (Gbps)
--------------------------------------------------------------------------------
fc1/30     200    F      auto    up            swl    F     16     --
fc1/31     200    auto   off     up            swl    F     4      --
fc1/32     200    auto   off     notConnected  swl    --    --     --
fc1/33     200    auto   off     up            swl    F     16     --
fc1/34     200    auto   off     up            swl    F     16     --
fc1/35     200    auto   off     up            swl    F     16     --
fc1/36     200    auto   auto    notConnected  swl    --    --     --
fc1/37     200    auto   off     up            swl    F     16     --
fc1/38     200    auto   off     up            swl    F     16     --
fc1/39     200    auto   off     up            swl    F     8      --
fc1/40     200    auto   off     up            swl    F     8      --
fc2/20     100    auto   auto    up            swl    E     8      4
fc2/21     100    auto   auto    up            swl    E     8      1
fc2/22     100    auto   auto    up            swl    E     8      1
fc2/23     100    auto   auto    up            swl    E     8      1
fc2/24     100    auto   auto    up            swl    E     8      1
fc3/43     200    E      off     up            swl    E     8      9
fc3/44     200    E      off     up            swl    E     8      9
--------------------------------------------------------------------------------
Interface            Vsan  Admin  Status        Oper  Oper    IP
                           Trunk                Mode  Speed   Address
                           Mode                       (Gbps)
--------------------------------------------------------------------------------
port-channel1        100   auto   up            E     64      --
port-channel2        200   auto   up            E     64      --
port-channel4        100   auto   up            E     32      --
port-channel8        200   off    up            E     8       --
port-channel9        200   off    up            E     16      --
```

Figure 4. Cisco MDS extract data snippet

Figure 5 shows an example of an abbreviated model per switch dump. For example, the line name TotalFcPortsCount in the figure is calculated by counting the number of relevant entries in the original switch dump. The line name FcFPortSpeedsUsed aggregates the speeds used for Cisco FC F-Ports from the original switch dump. For the sake of brevity, only a small portion of the parser extract and model data are shown in these figures.

Area	Location	SwitchType	Name	Value
test_team1	Tucson, AZ	cisco_fc	TotalSwitchWithDataCount	1
test_team1	Tucson, AZ	cisco_fc	TotalFcPortsCount	192
test_team1	Tucson, AZ	cisco_fc	TotalFcPortsLoggedIn	74
test_team1	Tucson, AZ	cisco_fc	PercentageFcPortsUtilized	38
test_team1	Tucson, AZ	cisco_fc	TotalVSANsCount	2
test_team1	Tucson, AZ	cisco_fc	AvgFcEPortRate	8
test_team1	Tucson, AZ	cisco_fc	HighesFctEPortRate	8
test_team1	Tucson, AZ	cisco_fc	AvgFcFPortRate	10
test_team1	Tucson, AZ	cisco_fc	HighestFcFPortRate	16
test_team1	Tucson, AZ	cisco_fc	FcFPortSpeedsUsed	(16,4,8)
test_team1	Tucson, AZ	cisco_fc	PortChannelUsage	yes
test_team1	Tucson, AZ	cisco_fc	LongestKernelUptime	41
test_team1	Tucson, AZ	cisco_fc	ShortestKernelUptime	41
test_team1	Tucson, AZ	cisco_fc	AssociatedSWHWVersions	6.2(7),cisco_MDS_9710

Figure 5. Cisco MDS single switch abbreviated base model.

B. Aggregation

All data from **Extraction** output files is grouped by switch type and switch locations into three files:

1. Summary of all entries,
2. Summary of all samples that contains "full data"
3. Summary of files with "no" or "partial" data.

The contents of the first two files reflect the model: Attributes and their aggregated values from the extraction phase output files. The third file contains an 'illegal' list that should be reviewed by IBM experts for the cause of the failure during collection. Figure 6 contains a subset example.

Area	Location	SwitchType	Name	Value
Test-lab-i	Austin, TX	cisco_fc	#Switches	6
Test-lab-c	Tucson, AZ	cisco_fc	#Switches	26
Test-lab-n	Raleigh, NC	brocade_fc	#Switches	5
Test-lab-d	Tucson, AZ	brocade_fc	#Switches	13
Test-lab-o	China	cisco_fc	#Switches	4
Test-lab-b	Tucson, AZ	brocade_fc	#Switches	20
Client1	NY	cisco_fc	#Switches	14
Test-lab-i	Austin, TX	cisco_fc	PortCount	516
Test-lab-i	Austin, TX	brocade_fc	PortCount	112
Test-lab-c	Tucson, AZ	cisco_fc	PortCount	1394
Test-lab-n	Raleigh, NC	brocade_fc	PortCount	568
Test-lab-d	Tucson, AZ	brocade_fc	PortCount	496
Test-lab-o	China	cisco_fc	PortCount	340
Test-lab-b	Tucson, AZ	brocade_fc	PortCount	2380
Client1	NY	cisco_fc	PortCount	2213
Test-lab-i	Austin, TX	cisco_fc	VSANCount	6
Test-lab-c	Tucson, AZ	cisco_fc	VSANCount	22
Test-lab-n	Raleigh, NC	cisco_fc	VSANCount	6
Test-lab-d	Tucson, AZ	cisco_fc	VSANCount	5
Test-lab-o	China	cisco_fc	VSANCount	34
Client1	NY	cisco_fc	VSANCount	23
Test-lab-n	Raleigh, NC	cisco_fc	PortSpeedsUsed	(4,8)
Test-lab-i	Austin, TX	cisco_fc	PortSpeedsUsed	(2,4,8)
Test-lab-b	Tucson, AZ	brocade_fc	PortSpeedsUsed	(4,8,16)
Test-lab-c	Tucson, AZ	cisco_fc	PortSpeedsUsed	(4,8,10,16)
Client1	NY	cisco_fc	PortSpeedsUsed	(1,2,4,8,10)

Figure 6. Summary of select full data samples

C. Model creation and data normalization

A functional model encapsulates the combination of data and analysis based on human expert knowledge to allow analysis and comparison of configurations. After carefully identifying the key data points we worked to create functional models and analysis capabilities based on domain expertise in SAN coverage and SAN test architecture. We created several different functional models. Functional models that

1. Identify interesting information in switch dumps, per switch type
2. Summarize the switch information per test lab
3. Summarize the switch information across test labs.

For all the models we worked with the SAN architecture and test experts to both identify the interesting data and define the attribute resulting from various computations over these data. Extracting the right data is prerequisite for a functional model, however, understanding, normalizing and properly qualifying the data values is essential to creating reliable analysis.

Figure 5 provides an example of a model that identifies interesting attributes. These attributes are computed from the raw switch data. Figure 6 provides an example of a model that summarizes the information per test lab. Figure 8 provides an example of a model that summarizes the information across test labs.

We found it essential to define model attributes that summarize data into single measures. This allows immediate comparison of configurations among different test labs and customers. For example, looking at Figure 6 we see a significant difference between Test-lab-n and Test-lab-b in the Brocade FC port count (2380 compared with 568).

Another example can be seen in Figure 8. In terms of Cisco NX-OS code levels. Test-lab-c is more similar to Client1 than Test-lab-i. These examples demonstrate a simple and straight forward comparison between configurations. This allows us to immediately spot differences at an eye glance. If needed, complex comparisons can be defined as well. Of course, the comparison can be automated.

D. Reporting

Data from the **Aggregation** phase is broken into several reports. There are two summary reports types: code levels and machine types, which are based on aggregation summary of all entry files and results report, which contains data including: switch functions, SAN design principles, switch utilization, port speeds, errors, peak traffic rates and average traffic rates. We also took into consideration the switches which may have been offline during the data collection phase. If our scripted process was unable to gather the switch dumps, the parser would attempt to analyze the data and if unsuccessful the parser will create an illegal switch summary report. Figure 7 shows a sample of illegal switches, with their given problem.

Directory	file	problem
/home/switchtoo	cisco_san_fcoe_20150216_152059.txt	switch_with_no_data
/home/switchtoo	brocade_san_fc_20150216_151801.txt	switch_with_no_data
/home/switchtoo	brocade_san_fcoe_20150216_155913.txt	NumberPorts_eq_0

Figure 7. Illegal switch summary

Figure 8 contains an example of number of switches running select Cisco NX-OS code levels from two IBM test labs and one client location. As you can see in Figure 8 Test-lab-c has a large variety of code levels running in its test environment, which include coverage of the levels in use by Client1. However, Test-lab-i has a smaller number of switches in test and the code level coverage is limited to the NX-OS 5.2.x, NX-OS 6.2.x, and NX-OS 7.0.x code streams. In order to best summarize the code coverage the switch code levels have been abstracted to show only two numeric values in the code stream. For example, both NX-OS 6.2.5a and NX-OS 6.2.9 would be referenced as NX-OS 6.2.x. This method of reporting was built into the aggregation model to better categorize and compare broad samples of data across a multitude of client and test labs. Although the data has been abstracted in this model, the full code stream data is also stored in a more detailed model for use by test teams focused more closely on specific switch code qualification test efforts.

Cisco NX-OS	Test-lab-c	Test-lab-i	Client1
3.2.x	4	0	2
3.3.x	3	0	0
4.1.x	3	0	2
5.0.x	5	0	0
5.2.x	8	1	10
6.2.x	6	5	0
7.0.x	4	2	0

Figure 8. Code level sample report

VI. EARLY RESULTS

We established a functional model, which gives a unified view of hundreds of SAN switches. See Figure 9 for details. IBM Systems Test switch count ratio is proportionate to the global SAN market share where Brocade is the #1 player in SAN [14] owning more that 54% of the Fibre Channel market in 2013 [15]. Although Brocade, Cisco and Lenovo are not the only SAN switches in IBM systems test environments, for the distance matrix project we made the conscious decision to focus on these brands to best align with market penetration and the majority of IBM SAN support statements. The goal of the SAN distance matrix project is to extract quarterly data in order to create trend reports, continually update test coverage and to understand what variables are changing or remaining static across test environments.

The first round of analysis completed in December 2013. As stated earlier, we utilized EASER Log Analysis to extract the information from the dumps. Coverage comparisons were established as we reviewed how the different test teams utilized their switches. Upon formulating the data, we created a functional model that has enabled us to provide results to IBM test teams. Those results have proven useful in driving interlock and complementary coverage models between IBM and switch vendors, and ensuring our test environments are representative of our clients.

	Cisco	Brocade	SND	Total
FC	65	335		400
FCoE	25	33	27	85
Total	90	368	27	**485**

Figure 9. Total number of FC and FcoE switches across Systems Test Groups

This information identifies key SAN coverage and test variants. For example, switch type, code versions, switch functions (enabled/disabled) and switch utilization (port speeds, errors, peak traffic rates, and average traffic rates). After analysis and review of the data within our team, we provide deep dive environment cross-test-cell reviews with test technical leads from IBM systems test labs world-wide.

Figure 10 shows a sample summary of two test groups located in Tucson, AZ and Hursley, UK. From this summary, we can easily examine the high-level switch usage across the two test groups. When the data is looked at over time it provides better insight into the environment variability and utilization rates for a given environment. The insight that can be derived from this high-level data summary is valuable, but limited. However, when the high-level utilization numbers are combined with other data factors and SAN coverage analytics, they can present powerful data points for skilled test architects and engineers to utilize in order to better adapt, design and drive the ideal levels of stress across test labs.

The detailed SAN coverage review allows test teams to easily identify their switch utilization rates and compare their environment numbers to a range of customer environments. The utilization data across time provides a better understanding of our global SAN test environments and drill-down capabilities for individual test labs. Additionally, when used in combination with trace coverage analysis teams are able to better perform gap analysis, code coverage reviews, and improve our larger system test coverage strategies.

Area	Location	SwitchType	Name	Value1	Value2
test-team1	tucson	brocade_fc	TotalSwitchWithDataCount	51	
test-team1	tucson	cisco_fc	TotalSwitchWithDataCount	6	
test-team1	tucson	cisco_fcoe	TotalSwitchWithDataCount	8	
test-team1	tucson	brocade_fcoe	TotalSwitchWithDataCount	4	
test-team2	hursley	brocade_fc	TotalSwitchWithDataCount	100	
test-team2	hursley	cisco_fc	TotalSwitchWithDataCount	4	
test-team2	hursley	brocade_fcoe	TotalSwitchWithDataCount	4	
test-team1	tucson	brocade_fc	TotalPortCount	2892	
test-team1	tucson	cisco_fc	TotalPortCount	500	
test-team1	tucson	cisco_fcoe	TotalPortCount	360	
test-team1	tucson	brocade_fcoe	TotalPortCount	124	
test-team2	hursley	brocade_fc	TotalPortCount	3443	
test-team2	hursley	cisco_fc	TotalPortCount	96	
test-team2	hursley	brocade_fcoe	TotalPortCount	128	
test-team1	tucson	brocade_fc	TotalPortsLoggedIn	1667	
test-team1	tucson	cisco_fc	TotalPortsLoggedIn	197	
test-team1	tucson	cisco_fcoe	TotalPortsLoggedIn	127	
test-team1	tucson	brocade_fcoe	TotalPortsLoggedIn	68	
test-team2	hursley	brocade_fc	TotalPortsLoggedIn	1536	
test-team2	hursley	cisco_fc	TotalPortsLoggedIn	23	
test-team2	hursley	brocade_fcoe	TotalPortsLoggedIn	108	
test-team1	tucson	brocade_fc	PercentageFcPortsUtilized	57	
test-team1	tucson	cisco_fc	PercentageFcPortsUtilized	48	
test-team2	hursley	brocade_fc	PercentageFcPortsUtilized	44	
test-team2	hursley	cisco_fc	PercentageFcPortsUtilized	23	
test-team1	tucson	brocade_fc	FcEPortSpeedsUsed	(16,2,4,8)	
test-team2	hursley	brocade_fc	FcEPortSpeedsUsed	(2,4,8)	
test-team1	tucson	brocade_fc	FcFPortSpeedsUsed	(1,16,2,4,8)	
test-team1	tucson	cisco_fc	FcFPortSpeedsUsed	(1,2,4,8)	
test-team2	hursley	brocade_fc	FcFPortSpeedsUsed	(16,2,4,8)	
test-team2	hursley	cisco_fc	FcFPortSpeedsUsed	(2,4)	
test-team1	tucson	brocade_fc	LongestKernelUptime	441	
test-team1	tucson	cisco_fc	LongestKernelUptime	329	
test-team2	hursley	brocade_fc	LongestKernelUptime	152	
test-team2	hursley	cisco_fc	LongestKernelUptime	152	
test-team1	tucson	brocade_fc	ShortestKernelUptime	107	
test-team1	tucson	cisco_fc	ShortestKernelUptime	108	
test-team2	hursley	brocade_fc	ShortestKernelUptime	3	
test-team2	hursley	cisco_fc	ShortestKernelUptime	16	
test-team1	tucson	brocade_fc	AssociatedSoftwareHardwareVersions	v6.4.3f	58.2
test-team1	tucson	brocade_fc	AssociatedSoftwareHardwareVersions	v7.2.1a	58.1
test-team1	tucson	cisco_fc	AssociatedSoftwareHardwareVersions	6.2(9)	cisco_MDS_91
test-team1	tucson	cisco_fc	AssociatedSoftwareHardwareVersions	5.2(8c)	cisco_MDS_91
test-team1	tucson	brocade_fcoe	AssociatedSoftwareHardwareVersions	v6.4.2b	76.7
test-team1	tucson	brocade_fcoe	AssociatedSoftwareHardwareVersions	v7.1.1c	76.7
test-team2	hursley	brocade_fc	AssociatedSoftwareHardwareVersions	v6.3.1b	26.2
test-team2	hursley	brocade_fc	AssociatedSoftwareHardwareVersions	v6.1.0b	42.2
test-team2	hursley	cisco_fc	AssociatedSoftwareHardwareVersions	3.2(2c)	cisco_MDS_91

Figure 10. Switch compare sample summary

A. Interlock Test Coverage

With the various test teams located world-wide, the need for a central list of SAN switch hardware across IBM test has become apparent. The information gathered from the switches is an initial step in allowing IBM systems test groups to more closely interlock and drive test coverage across test labs. The SAN distance matrix project has helped us to identify test labs that are closely aligned and those that provide unique coverage points. While continuity is important and we need to ensure we are covering the most typical SAN field deployments we also realize the need to balance that model with one of broad coverage.

B. Additional benefits from early results

Along with balancing our coverage, the early results provided insight on switch utilization that provided additional benefits to our test teams.

It also allowed us to identify groups utilizing dated switch hardware and place them into a hardware refresh pool to help us get new switches to the teams that may need it the most. It also allowed us to collect information on which test groups were on IBM supported Cisco and Brocade switch code levels. Testing a variety of code levels helps in our testing coverage since customers have a variety of environments and update at different rates.

Another benefit from the results was that we were able to look at switch utilization and stress rates to ensure we are accurately stressing our equipment and in the identified cases where we were not, to put plans in place to help increase load coverage. With this type of review of environment architecture designs we can recommend changes or complexity additions where appropriate and create more customer-like environments.

Figure 11 gives an example of a cross-test-cell review, which was done with one systems test group that consisted of a main test coverage mission spread across five environments at unique site locations.

Each of these test groups were responsible for unique IBM Server and Storage focused system test. This project provided the framework and data to bring the groups together to collectively review, compare and analyze how each group architected, deployed and utilized their SAN switches. The groups benefited from having a better understanding of the broader SAN coverage model. From these reviews, we are able to recommend changes and/or complexity additions to each SAN environment. Additionally, the broader coverage review exercise proved to be useful and was later implemented on a more frequent basis across the labs in this illustrative example.

Name	test-team1	test-team2	test-team3	test-team4	test-team5
AvgFcEPortRate	8	N/A	3	8	4
AvgFcFPortRate	6	4	4	9	8
HighesFctEPortRate	8	N/A	10	8	4
HighestFcFPortRate	8	4	8	16	8
LongestKernelUptime	329	152	195	56	54
ShortestKernelUptime	108	16	108	1	54
PercentageEthPortsUtilized	3	N/A	N/A	N/A	N/A
PercentageFcPortsUtilized	48	23	25	27	12
PercentagePortsUtilized	39	23	25	26	12
PercentageVfcPortsUtilized	50	N/A	N/A	N/A	N/A
PortChannelUsage	no,yes	no	no,yes	yes	no
TotalEthPortsCount	96	N/A	N/A	16	N/A
TotalEthPortsLoggedIn	3	N/A	N/A	N/A	N/A
TotalFcPortsCount	402	96	452	610	48
TotalFcPortsLoggedIn	193	23	113	169	6
TotalPortCount	500	96	452	626	48
TotalPortsLoggedIn	197	23	113	169	6
TotalVfcPortsCount	2	N/A	N/A	N/A	N/A
TotalVfcPortsLoggedIn	1	N/A	N/A	N/A	N/A
TotalSwitchWithDataCount	6	4	5	4	1
TotalVLANsCount	3	N/A	N/A	2	N/A
TotalVSANsCount	44	N/A	4	9	1

Figure 11. Cisco FC Cross-test-cell Results Table

Overall, we were able to systematically collect data from global IBM systems test labs and create a centralized view of SAN switch equipment and coverage across IBM systems test. The scripted process extracted data from the switch dumps was used to build compare logic to define and understand meaningful distances (comparisons) among the groups as well as summarize the charted data to compare trends and coverage analysis over time. We were also able to gather dump data from select customers representing a broad range in company size and industry focus. The comparison of our test lab coverage models with customer environments allows our test teams to continually alter test configurations and architectures to be more customer-like and helping to ensure our testing is continually evolving and relevant. The goal of this project was not to be used as a SAN report card, grading tool, or micromanaging utility, but rather an overall method to look across IBM test groups and understand large scale SAN coverage models and gaps and continual areas for improvement.

VII. ADDITIONAL BENEFITS REALIZED

Since its inception, the distance matrix project has shown added value in many foreseen and unforeseen ways. The largest benefit of this project is the ability to systematically extract and model coverage across a large number of test and client SAN environments enabling increased coverage without expanding resource requirements or timelines. One of the key success factors for this model is its scalability. The scalability and reach of the distance matrix project has also uncovered additional unforeseen benefits. In this section we will introduce and expand briefly on a few of these benefits:

1. Centralized visibility of switch inventory and distribution across IBM test teams
2. Decreased root cause analysis time,
3. Client critical situation recreate advancement opportunities
4. Increased technical interlock across systems test labs

The value of asset knowledge and a centralized view of deployed SAN switches across IBM test labs is a critical success factor to make enlightened decisions considering the infrastructure as a whole. The distance matrix project provided a centralized list of switches and switch characteristics across IBM test labs world-wide. This centralized view allowed vested parties to review deployed assets and increase asset pooling, sharing and roll-off sharing. For example, when a large SAN lab in Tucson, AZ was undergoing a reconfiguration project and upgrading its SAN infrastructure the team was able to make better informed decisions of which teams could benefit from the surplus switches removed from the previous environment.

Decreased root cause analysis time is another side benefit that can be extracted from the distance matrix project. Having the data to understand which switch configurations encountered certain defects provides valuable insight that can lead to decreased root cause analysis time frames. Additionally, the data can be used to extract trending information on SAN topologies and characteristics that most often lead to increased defect discovery.

Another side benefit realized during the course of this project is the ability to utilize the centralized switch and topology data across test labs to select the most appropriate lab and location for customer debug or recreate activities. For example, if a Customer is experiencing an issue in a Brocade SAN environment with XIV storage we can take the Brocade environment specifics including port speeds, code levels and environment complexity and search across IBM test labs for the environment best resembles the customer environment to setup the recreate. Utilizing this method helps cut recreate time by mitigating the time needed for test or support teams to reconfigure an environment to closely resemble the customer environment.

This project also led to increased technical interlock and technical sharing across worldwide systems test labs. Since its inception, the project has been well received across systems test labs and has helped to create an open dialogue and tool for sharing coverage and best practices across the systems test labs world-wide. By forming a review and sharing process across technical test leads and architects the distance matrix project has sparked strong ongoing relationships and dialogue across key technical leaders world-wide. Teams that originally created their designs in a more isolated environment now have extended resources and lab models available to them for review and leverage. In a company as large as IBM, bringing together test leaders across systems test labs and providing an open sharing SAN coverage model for continued technical leverage across world-wide test environments is a critical step in the right direction.

VIII. CONCLUSION AND FURTHER DEVELOPMENT

As solution complexity and the number of supported configurations increase in the IT industry, we must continue to re-invent the ways we do solution testing. In our global test environment, the need to have procedures in place to extract data and create advanced comparison and coverage models is essential.

This project has shown tremendous promise for being able to systematically extract and model coverage across a large number of test and client SAN environments. One of the key factors of this models continuing success is its scalability. The IBM test group started with business requirements and an early operational model vision and worked directly with the IBM Haifa Research lab to expand and translate early visions into a working model that is currently being deployed and leveraged across test labs world-wide.

We are currently working on plans to extend the distance function beyond reducing the data to a single dimension. For example, today one distance function is the difference in the average rates among different groups. We could instead compute a distance metric over the rate vectors. We are also looking into opportunities to expand the areas of coverage, the scope of the environments we are able to capture and working on data optimization and smart analytics to help ensure we continue to provide leading edge test coverage and innovation.

As we continue to implement the distance matrix project across test labs within IBM we are gathering key data and making methodical changes is SAN test architecture to provide better test coverage points for IBM products and solutions.

REFERENCES

[1] Y. Adler, T. Astigarraga, S. Jackson, Jose R. Mosqueda, and O. Raz, "IBM SAN Distance Matrix Project," Proc. VALID, 2014, The Sixth International Conference on Advances in System Testing and Validation Lifecycle (VALID 2014), IARIA October 2014, pp. 84-87, ISSN: 2308-4316, ISBN: 978-1-61208-370-4.

[2] "IBM Basics," ibm.com [Online]. Available from: http://www.ibm.com/ibm/responsibility/basics.shtml. [Accessed: 2015-05-19].

[3] T. Astigarraga, "IBM Test Overview and Best Practices" SoftNet 2012, Available from: http://www.iaria.org/conferences2012/filesVALID12/IBM_Test_Tutorial_VALID2012.pdf, [Accessed: 2015-05-19].

[4] "IBM System Storage Interoperation Center (SSIC)," ibm.com [Online]. Available from: http://www-03.ibm.com/systems/support/storage/ssic/interoperability.wss, [Accessed: 2015-05-19].

[5] "Cisco DCNM Overview," cisco.com [Online]. Available from: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/mds9000/sw/5_2/configuration/guides/fund/DCNM-SAN-LAN_5_2/DCNM_Fundamentals/fmfundov.html, [Accessed: 2015-05-19].

[6] "Brocade Network Advisor," brocade.com [Online]. Available from: http://www.brocade.com/products/all/management-software/product-details/network-advisor/index.page, [Accessed: 2015-05-19].

[7] "EMC Storage Performance Monitoring," solarwinds.com [Online], Available from: http://www.solarwinds.com/es/solutions/emc-storage-performance.aspx, [Accessed: 2015-05-19].

[8] "IBM Tivoli Monitoring," ibm.com [Online]. Available from: http://www-03.ibm.com/software/products/en/tivomoni/, [Accessed: 2015-05-19].

[9] Kuhn, D. Richard, Dolores R. Wallace, and Jr AM Gallo. "Software fault interactions and implications for software testing," Software Engineering, IEEE Transactions on 30.6 (2004), pp. 418-421.

[10] Changhai Nie and Hareton Leung, 2011, "A survey of combinatorial testing," ACM Comput. Surv. 43, 2, Article 11 (February 2011).

[11] I. Segall, R. Tzoref-Brill, E. Farchi, "Using Binary Decision Diagrams for Combinatorial Test Design," ACM, 2011, Proc. 20th Intl. Symp. on Software Testing and Analysis (ISSTA'11).

[12] "IBM Functional Coverage Unified Solution (IBM FOCUS)," ibm.com [Online]. Available from: http://researcher.watson.ibm.com/researcher/view_group.php?id=1871, [Accessed: 2015-05-19].

[13] Y. Adler, A. Aradi, Y. Magid, and O. Raz, "IBM Log Analysis Tool (EASER)," unpublished.

[14] "Beating the Tech Titan" [Online], Available from: http://www.investingdaily.com/22245/beating-the-tech-titan/, [Accessed: 2015-05-19].

[15] "Brocade gains SAN market share in 2013, Cisco dips, says Infonetics" [Online], Available from: http://www.infotechlead.com/networking/brocade-gains-san-market-share-2013-cisco-dips-says-infonetics-20880, [Accessed: 2015-05-19].

Novel High Speed and Robust Ultra Low Voltage CMOS NP Domino NOR Logic and its Utilization in Carry Gate Application

Abdul Wahab Majeed*, Halfdan Solberg Bechmann †, and Yngvar berg ‡
Departments of Informatics
University of Oslo, Oslo, Norway
*Email: abdulwm@ifi.uio.no
†Email: halfdasb@ifi.uio.no
‡Email: yngvarb@ifi.uio.no

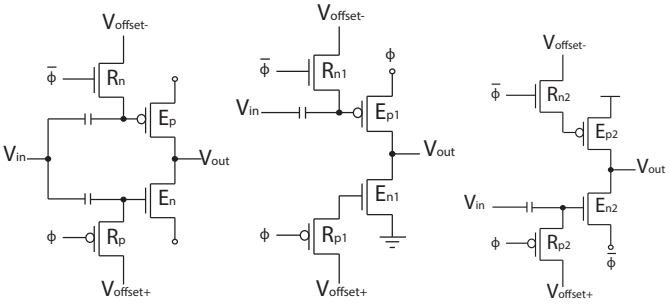
***Abstract*—This paper is based on two parts. In Part 1, we shall present a new Ultra Low Voltage Static differential NOR topology. This will show how Ultra Low Voltage circuits are designed and what are the pros and cons of these circuits. In Part 2, utilizing the design presented in Part 1, we shall present a novel design of an Ultra Low voltage Carry Gate. This shall emphasize the use of such design in an application such as carry gate. The Ultra Low Voltage topologies presented in Part 1 are well known for their high speed relative to conventional CMOS topologies regarding subthreshold operation. The main objective is to target the robustness of the presented ciruits. We shall also imply as to what extent these circuits can be improved and what their benefits, compared to conventional topologies, are. The design presented in Part 2, compared to a conventional CMOS carry gate, is area efficient and high speed. The relative delay of a Ultra Low Voltage carry gate lies at less than 3% compared to conventional CMOS carry gate. The circuits are simulated using the TSMC 90nm process technology and all transistors are of the Low Threshold Voltage type.**

***Index Terms*—ULV; Carry Gate; NP domino.**

I. Introduction Part 1

Technology, being an important factor of the modern civilization, has been facing challenges enormously in every aspect. Demand for low power and faster logic pursue an overwhelming position in modern electronic industry. As the industrial demand grows for the CMOS transistor, from time to time it needs to go through rehabilitation process accordingly. However, as the Moores law suggests, advancement in CMOS technology, in the means of dimension scaling has almost hit a barrier for a number of reasons. The most important one is power dissipation at the smaller dimensions of the transistor. To overcome this problem, a number of approaches have been proposed [1][2]. Scaling supply voltage (V_{DD}), being prominently the most effective, has been proposed and adopted by many [3][4][5]. However, scaling supply voltage has an adverse affect on the performance of the CMOS circuits as it decreases the ON current I_{ON} and hence the speed.[6] presents a solution to this problem by employing floating gate Ultra Low Voltage (ULV) design, which raises the DC level of the input floating node even more than the supply voltage itself and thereby increasing the I_{ON}.

Floating-gate is achieved by connecting a capacitor at the input of the transistor gate. This isolates the gate terminal electrically, i.e., no DC path to a fixed potential. Such a gate is called non-Volatile Floating-gate. Given that the transistor dimensions are smaller than 0.13 μm and gate oxide is less than $70AA\frac{1}{2}$, there shall be a significant gate leakage current. To avoid this leakage, frequent initialization of the gate is required. This can be achieved by connecting floating gates of the NMOS input transistor and PMOS input transistor to, a fixed potential, i.e. through a PMOS to the Voffset+ and through NMOS to the Voffset- respectively. This approach, first presented in [4], is called semi-floating gate (SFG) and has been used in this paper. In Section I-A, a brief introduction to ULV design is presented. Section II, presents, first, a Non-differential circuit proposed in [7] and thereby presents a new solution to the problem encountered in non-differential ULV circuit by designing a new NP domino static differential ULV NOR. In Section III, we shall present the simulation results of all the ULV circuits and Dual Rail domino NOR relative to conventional NOR.

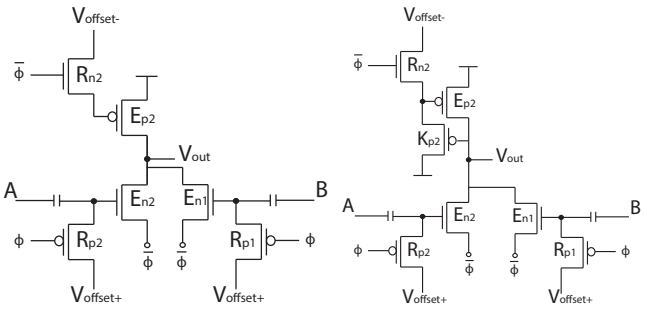


(a) A simple SFG-ULV inverter (b) P-type inverter (c) N-type inverter

Fig. 1. SFG NP ULV domino inverter.

A. ULV Inverter

1) Evaluation and Precharge Phase: A simple ULV inverter model is presented in the Figure 1a. A ULV SFG circuit design consists of two phases. An evaluation phase, determined by the evaluation transistors E_n and E_p, and a precharge phase determined by the precharge transistors R_n and R_p. As seen in the Figure 1a inverted clock ($\overline{\phi}$) is applied to R_n and clock (ϕ) is applied to R_p. In such a circuit, the precharge phase occurs when ϕ=0 and the circuit enters evaluation phase when ϕ=1. During the precharge phase, the input floating nodes are charged to a desired level, i.e, logical 1 or V_{DD} for the E_n floating gate and logical 0 or Ground



(a) N-type ULV Dynamic NOR (b) N-type ULV Static NOR

Fig. 2. NP domino dynamic and static ULV NOR gate.

(GND) for the E_p floating gate. No input transition occurs during the precharge phase. However, once the clock shifts from logical 0 to 1 and has reached a stable value of 1, an input transition may occur, which determines the logical state of the circuit's output. We can engage the circuit in an NP domino chain by connecting the source terminal of E_n to $\overline{\phi}$ (where $\overline{\phi}$=1 during the precharge phase) and the source terminal of the E_p to V_{DD}. Such a configuration gives us a precharge level of logical 1 and is called an N-type circuit. On the other hand, if we connect the source terminal of E_n to GND and the source terminal of the E_p to ϕ we can obtain a precharge level of 0. Such a configuration is called a P-type circuit.

Considering the example of N-type inverter, we know that the output of the N-type is precharged to 1. Once ϕ shifts from 0 to 1, circuit enters the evaluation phase. During the evaluation phase, there are two possible scenarios. If no input transition occurs, the output shall remain unchanged and hold its value to 1. Indicating that no work is to be done. However, if an input transition occurs and input is brought to 1 the E_{n2} shall be turned on and the output shall be brought to logical 0 or close to 0. This indicates that the only work to be done during the evaluation phase is to bring the output from 0 to 1 when an input transition occurs.

We have seen that the only work that is to be done, during the evaluation phase, is to bring the output to the logical 0 when an input transition occurs. This suggests that E_{p2} does not require an input transition at any stage. Therefore, we can remove the input capacitor of E_{p2}. Such a configuration can be called pseudo SFG ULV inverter and is shown in Figure 1c. An equivalent P-type Pseudo SFG ULV inverter is shown in Figure 1b. This will lead to load reduction and hence higher speed. However, we may encounter some robustness issues with respect to noise margin due to leakage current.

II. Methods

A. Non-differential ULV NOR circuit

A Non-differential Dynamic ULV NOR (DULVN) and Static ULV NOR (SULVN) gate is shown in Figure 2. Recall configuration of ULV inverter. The only difference in configuration of ULV NOR circuit is that, in order to obtain a P-type DULVN, we have to apply an extra input at the evaluation transistor E_{p1} in a P-type inverter. In order to obtain N-type DULVN employ an extra evaluation transistor E_{n1} in parallell with E_{n2}.

The SULVN is configured in the same manner as the DULVN. However, we add keeper transistor in the described configuration. An NMOS keeper is connected to the floating gate of E_{n1} and a PMOS keeper is connected to the floating gate of the E_{p2} in P-type and N-type SULVN, respectively. However, these circuits are prone to some noise margin (NM) issues due to precharged input floating nodes that holds it's value under evaluation phase. Thereby resulting in short circuit leakage current. In order to solve this problem, let us consider an example of N-type DULVN. Discharging of the floating gate of the $E_{n1/n2}$, when no input transition occurs, and charging of the floating gate of the E_{p2} with V_{DD}, when an input transition occur, will ensure a better noise margin. This can be achieved by engaging keeper transistors at these nodes and connecting the source and drain terminal of the keeper transistors K_p and K_n, respectively, which may not interupt in precharging of the floating gate and and still manages to discharge these nodes under evaluation phase when required. A problem with such a circuit is the potential false output transient if the input transient is significantly delayed compared to the clock edge [7]. Synchronization of the signals employed through the keeper transistors with the input may solve the problem.

B. Static differential ULV NOR

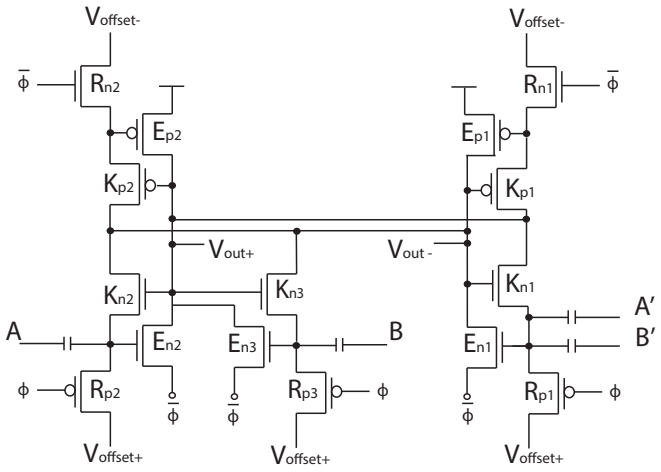
A static differential ULV NOR (SDULVN) gate will always have the same precharge level at the both outputs in the preacharging phase and differential outputs in the evaluation phase. A SDULVN gate is shown in Figure 3. We have connected outputs of the opposite ends, V_{out+} and V_{out-}, at the drain terminals of the both keeper transistors. In order to achieve maximum robustness, MTCMOS method is used, i.e, transistors in the path with critical timing has lower threshold voltage, to achieve the maximum speed, and transistor in the path with critical leakage issues has higher threshold voltage, to achieve the minimum leakage.

III. Simulation Result

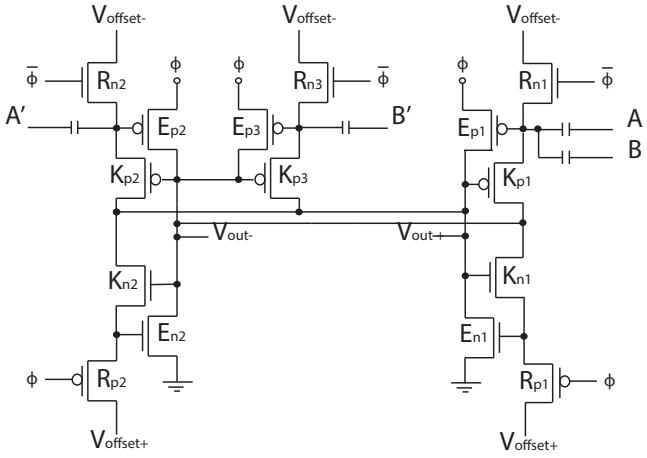
We have simulated four different topologies, conventional Dual Rail domino NOR, DULVN, SULVN and SDULVN each with a load of FO1. Worst case scenario for three ULV topologies, considering delay, is when both inputs has opposite logical values and considering power and NM is when the output holds the precharged value under evaluation phase.

A. EDP and PDP of Dynamic, Static, Differential ULV topology and dual rail domino NOR

It is suggested in [3] that in subthreshold regime the transistor may operate as a current source, hence switching the output. Author suggests that the transistor may work as a current source for as little as 100 mV at room temprature.



(a) Static differential N-type ULV NOR^2/NAND^2.



(b) Static differential P-type ULV NOR^2/NAND^2.

Fig. 3. Static Differential NP ULV NOR.

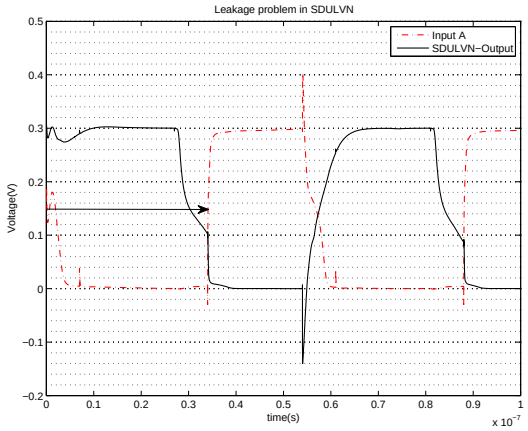


Fig. 4. Output of SDULVN where input is delayed and the output tends to shift before input due to leakage current. The graph is taken from Monte Carlo simulation in order to show why the limits for functional circuits are set as they have been discussed. Supply voltage at 300 mV.

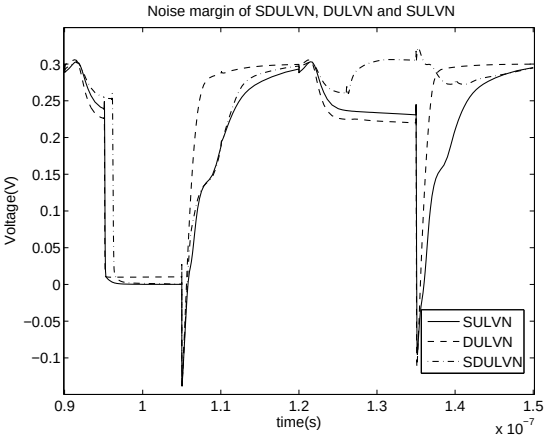


Fig. 5. Noise margin of SDULVN compared to DULVN and SULVN. Supply voltage of 300 mV.

So, before we start analyzing EDP and PDP for varied supply voltages, we have to set some limits that constitutes a functional circuit. Mentioned earlier, dynamic and static topologies suffers from current leakage problem. So as we increase the V_{DD} current leakage increase resulting in a non-functional circuit. However, this can be overcommed by strengthening down the transistor. SDULVN manages to integrate itself according to the input provided. So, even if the output is delayed and a leakage occur, at the arrival of input it will manage to change the output accordingly. However, if the leakage in device is greater than $V_{DD}/2$, we may not be able to measure a propagation delay. Figure 4 highlights the leakage problem, where a measurement of propagation delay is avoided due to early switching of the output. Thus, in order to achieve maximum robustness, we shall consider a circuit non-functional if the output of the circuit exceeds $V_{DD}/2$.

Figure 5 shows improved noise margin of differential topology with respect to dynamic and static topology at supply voltage of 300 mV. Keeper transistors manages to turn off the evaluation transistors when required. Consequently, SDULVN has 30% and 36% better noise margin in worst case scenario compared to SULVN and DULVN, respectively. Delay of ULV NOR topologies relative to conventional NOR can be seen in Figure 6. Average relative delay of SDULVN lies at 6%, i.e, to switch an output SDULVN consumes 6% of time consumed by a conventional NOR to switch an output. Figure 7 shows the PDP of three ULV topologies. It shows that ratio between the relative delay and relative power normalizes itself to unity at some supply voltages yet SDULV wins at most of them. EDP graph shown in Figure 8 again displays enhanced speed of the ULV topologies overcomes exaggerated power dissipation.

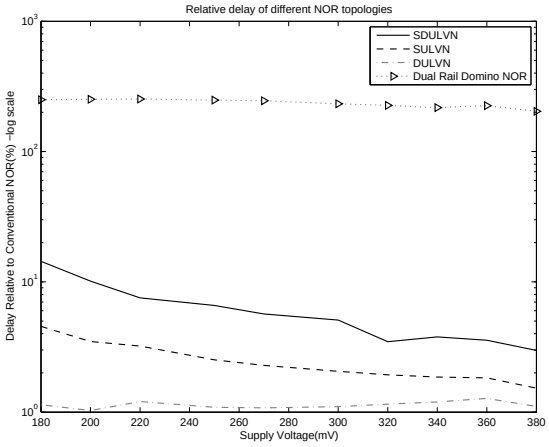


Fig. 6. Relative delay of three ULV topologies Dual Rail domino NOR to conventional NOR.

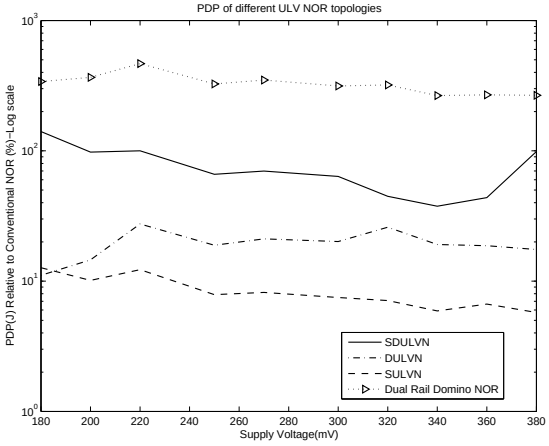


Fig. 7. Relative PDP of three ULV topologies Dual Rail domino NOR to conventional NOR.

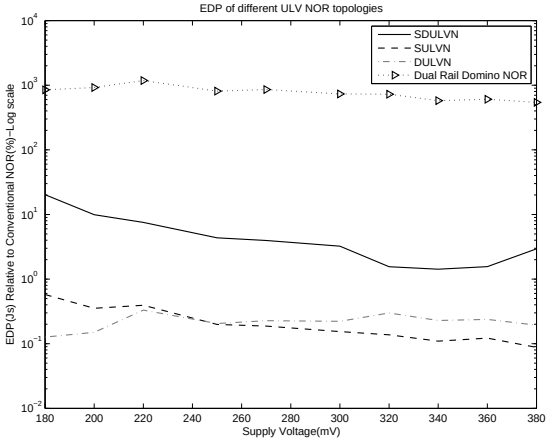


Fig. 8. Relative EDP of three ULV topologies and Dual Rail domino NOR to conventional NOR.

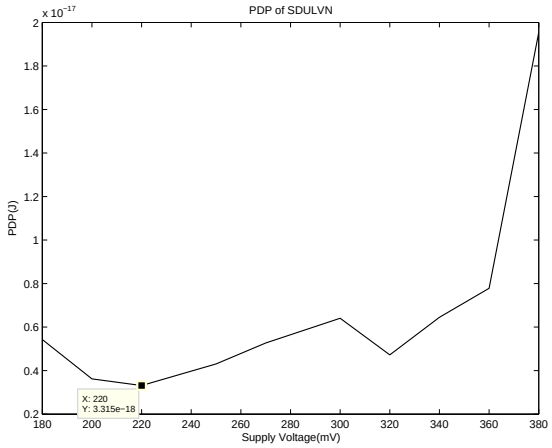


Fig. 9. PDP of SDULVN. Graph shows the Mimimum Energy Point

B. Maximum and minimum Supply voltage and Minimum energy point

Threshold voltage for these low voltage transistor lies around 260 mV with normal strength. As we have strengthen up the evaluation transistor, threshold point decreases. So, we can see from Figure 9 that minimum energy point of differential topolgy lies around 220 mV. Taking into consideration our limits that constitutes a functional circuit we got minimum and maximum V_{DD} at 180mV and 380mV, respectively.

C. Process and mismatch variation

Attributes of a transistor at 90nm process suffers from variation under fabrication. Such variations can be of two types, inter-die, where all the transistors are printed on one die and may be shorter than normal because they were etched excessively, and intra-die, where number of dopant atoms implanted varies from neighboring transistor [8]. A change in behavior of the circuit can occur due to variation in V_t and channel length. Therefore, it is important to highlight this variation in any circuit. In order to obtain an idea of how robust a circuit tends to be toward process and mismatch variation, Monte Carlo simulation environment is the best solution to apply. A number of precautions can be taken to avoid further variation. Such as sizing up transistors, carefull layout design and so on. Law of Large number indicates that the larger the number of trials the closer it would be to expected value. Therefore, the number of simulations in Monte Carlo should be high as possible. We have used 100 simulations to mark the mean value.

D. PDP, EDP and minimum energy point results in Monte Carlo environment

Figure 10 shows that the minimum energy point shifts from 220 mV to 250 mV due to process and mismatch variation. As stated earlier V_t varies due to random number of dopant atom. This results in slight randomness in behavior of the

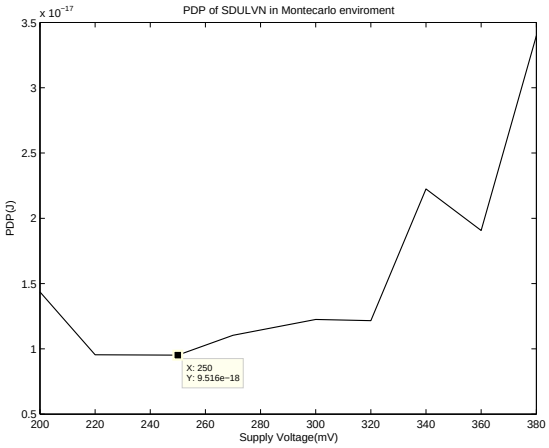


Fig. 10. PDP of SDULVN in Monte Carlo environment. Graph exhibits shift in Minimum Energy Point.

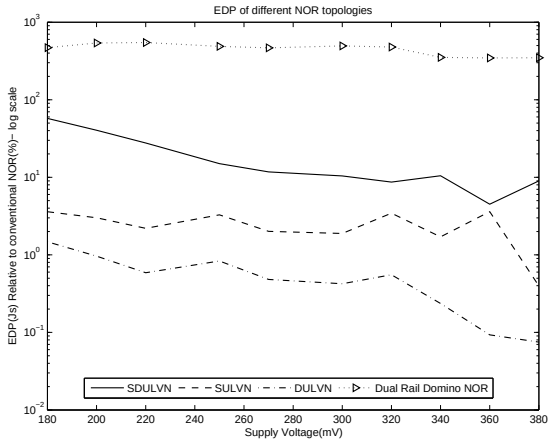


Fig. 12. Motecarlo simulation for EDP of three ULV topologies and Dual Rail domino NOR relative to conventional NOR.

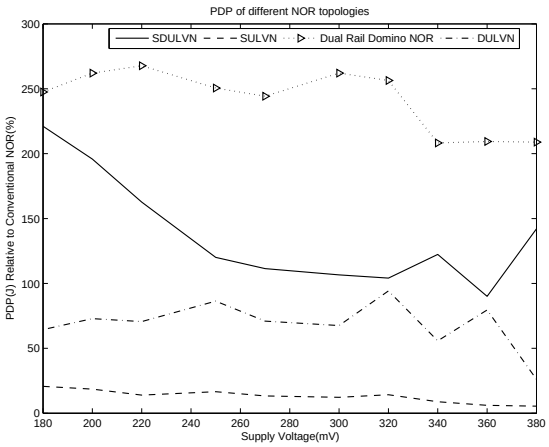


Fig. 11. Motecarlo simulation for PDP of three ULV topologies Dual Rail domino NOR relative to conventional NOR.

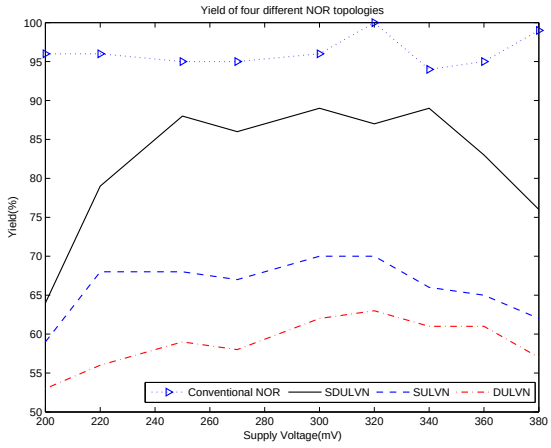


Fig. 13. Yield of SDULVN, DULVN, SULVN Dual Rail domino NOR and conventional NOR.

circuit and, therefore, results in shift in minimum energy point. Figure 11 shows PDP of three different ULV NOR topologies relative to conventional NOR and compared to Dual rail domino NOR. Figure 12 shows that EDP fluctuates from having relative mean value of 76% to 120% due to process variation.

E. Yield and 3σ EDP variation

As described earlier, we have to set a limit to differ between a functional and non-functional circuits. Considering those limits we have taken out graph for yield of all these circuits. Figure 13 shows that of ULV gates SDULVN has the best yield at an average of 82% yield compared to SULVN and DULVN, which have an average of 66% and 58% yield respectively.

$$PDP = V_{DD}^2 \cdot C \tag{1}$$

$$EDP = \frac{C^2 \cdot V_{DD}^3}{I} \tag{2}$$

As we know that objective of employing semi floating gate is to increase the current to increase the speed of the circuit. PDP is independent of current as shown in (1). Therefore, in order to obtain a variation that occurs due to higher current we have to focus on variation of EDP. Figure 13 shows 3σ variation of the EDP for four different NOR topologies. We can see that below minimum energy point of SDULVN the variation are alot higher than conventional topology. However, above this point, the EDP variation in SDULVN is better than Dual Rail domino NOR and almost at same level as in DULVN and conventional NOR.

IV. Conclusion Part 1

In Part 1 we have presented a new design for NP domino ULV NOR topology and demonstrated improvement in NM and yield. Although SDULVN topology has 2× the logic

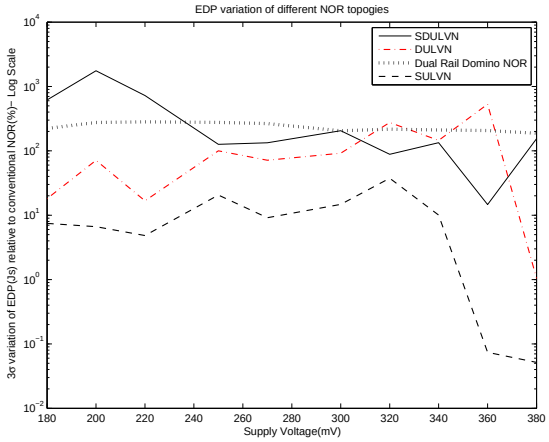


Fig. 14. 3σ variation of EDP for SDULVN, DULVN, SULVN relative to conventional NOR.

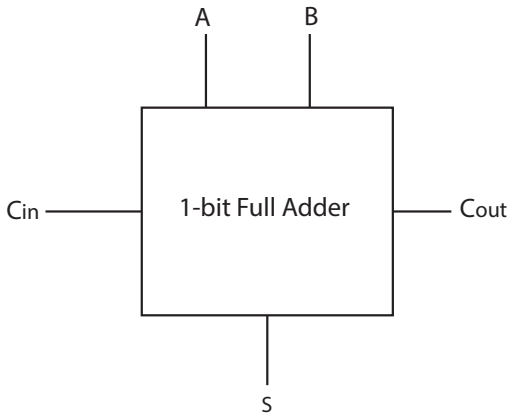


Fig. 15. 1 bit full adder

and almost $3\times$ complexity (number of transistor operate under evaluation phase) compared to conventional NOR, it is still $17\times$ faster. The output leakage problem encountered in SULVN and DULVN has been minimized by employing SDULVN design.

V. Introduction Part 2

As stated earlier, the demand for Ultra Low Voltage (ULV) circuits is increasing with the growth of the semiconductor industry. These circuits are being implemented in VLSI, where different kind of functions are combined on one chip. The Arithmetic Logic Units (ALU)s are one of the many circuits that are implemented in the VLSI chips. Since an adder is an important part of the ALU, the speed of the adder used, is important for the ALU's performance. The speed of the adder is determined by the propagation delay of the carry chain. Although high speed conventional carry circuits like Carry Look Ahead, Dual rail domino carry, CPL, etc., are well established design topologies, their performance suffers from degradation at ULV [9]. Several approaches are proposed for the improvement in performance [10][11] but the design presented in this paper is influenced by [12]. This paper shall present a new high speed NP domino ULV carry design. To highlight the improvement, the results shall be compared to conventional domino design such as Dual Rail Domino carry. In order to show as to what extent one is better than the other, regarding their speed and power, both the carry circuits are implemented in a 32-bit carry chain.

Section I-A presented a general introduction to the ULV circuits presented in [4]. Section VI presents different configurations of ULV carry designs and gives an explanation on how it works. Section VII presents the performance of the proposed ULV carry gate compared to the conventional carry gate.

VI. Methods

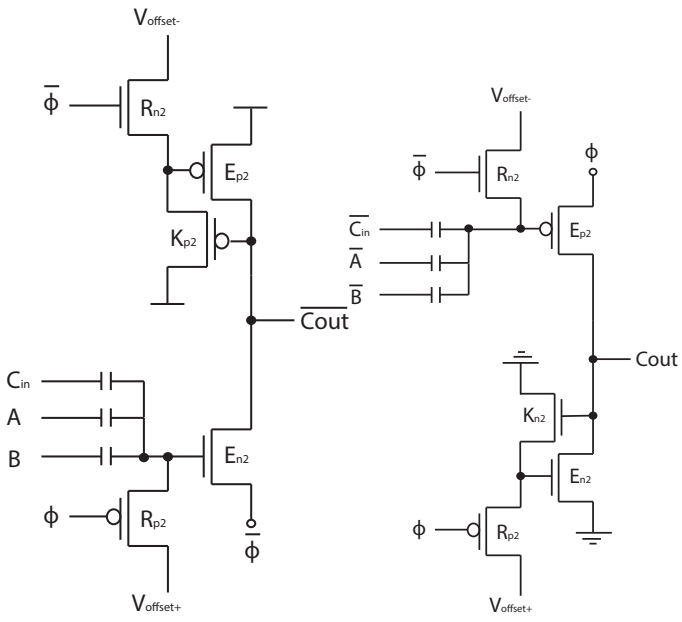
$$C_{out} = A \cdot B + (C_{in} \cdot (A \oplus B)) \quad (3)$$

The output of a carry circuit is generated using two inputs and a carry bit from the previous stage, if available (carry bit at the least signficant bit is always zero so it has no previous carry), as shown in Figure 15. Equation (3) shows an arithmetic approach to carry generation, where A and B is the input signal and C_{in} is the carry bit from the previous stage. There are two parts of this equation, one is generated internally, $A \cdot B$, and can be called carry generation (CG), the other one is dependent on the carry bit from the previous stage, $(C_{in} \cdot (A \oplus B))$, and is known as carry propagation (CP). The speed of any carry chain depends on the second part of this equation, because it has to wait for the carry bit from the previous stage to arrive. Inputs A and B both arives simultaneously at any stage of an N bit carry chain. Most conventional designs use two seperate parts for CG and CP but the design presented in this paper differs from the most designs as it is able to generate both CG and CP by applying all the inputs to a single transistor. This technique is called Multiple valued Logic (MVL), where classical truth value, logical 1 and 0, are replaced by finit or infinite logical values. It has a potential to decrease the chip area and total power dissipation [13].

A. Non-Differential Carry Gate

The Static Ultra Low Voltage Carry (SULVC) is a modified version of the ULV N-P domino inverter shown in Section I-A. The carry circuit uses a keeper, as proposed in [4], and 3 capacitors in parallel at the input gate providing the input logic for the circuit. The circuit is designed to make the input signals, A and B signal, cancel each other out when A and B have contrasting values to allow the carry input signal to determine the carry output in this case. Because of the cancellation requirement between the A and B signals they need to arrive as equally sized rising or falling transitions, this can be acheived by utilizing level-to-edge converters or a logic style with a VDD/2 precharge level.

If both A and B are rising, the floating node will rise causing a falling transition on the carry output of the N-



(a) N type SULVC. (b) P type SULVC.

Fig. 16. SULVC circuit.

TABLE I. TRUTH TABLE FOR A CARRY CIRCUIT

Inputs			Output
A	B	C_{in}	C_{out}
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

type circuit regardless of the carry input signal. If they are both falling, the carry input signal can not elevate the floating node voltage enough to cause a transition, leaving the carry output at precharge level. If A and B are not equal, their two transitions cancels each other out and the floating node remains at precharge level until a possible rising edge occurs on Cin. A P-type equivalent of the circuit is shown in Figure 16 (b), where all signals and logic are the inverse of those in the N-type circuit. For both circuits, a transition on the output indicates carry propagation and they can both be characterized as a carry generate circuit corresponding with the truth table shown in Table I, the transition logic for the N-type circuit can be seen in Table II.

During the precharge phase, the voltage level of the floating node is set to ground for the P-type circuit and V_{DD} for the N-type and can only be changed by the inputs through the capacitors in the evaluation phase. In these circuits, when used in CPAs (Carry Propagate Adder), C_{in} can arrive later than A and B when the carry bit has to propagate through the chain of carry circuits. This introduces the challenge of keeping the

TABLE II. TRANSITION TRUTH TABLE FOR N-TYPE SULVC

Inputs			Output
A	B	C_{in}	$\overline{C_{out}}$
↓	↓	0	1
↓	↑	0	1
↑	↓	0	1
↑	↑	0	↓
↓	↓	↑	1
↓	↑	↑	↓
↑	↓	↑	↓
↑	↑	↑	↓

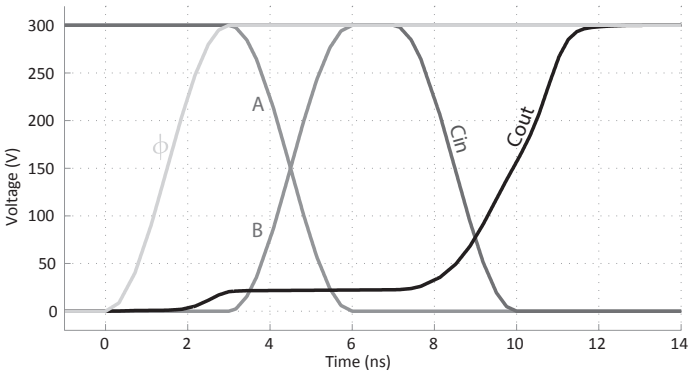


Fig. 17. Carry input and output for SULVC gate. Supply voltage at 300mV.

output precharge value during the evaluation phase in case no carry signal arrives. As Figure 17 shows, the floating node of the P-type circuit is precharged to 0V. This causes the transistor E_{p2} in Figure 17 (b) to conduct and the output will drift and may eventually cause an incorrect output value as shown in Figure 18 at 70ns. The drifting effect is countered with the K_{n2} and K_{p2} keeper transistors but the effect limits the length of the evaluation pase and therby the number of carry circuits that can be put in a chain and the maximum number of bits an adder based on the circuit can process in one clock cycle. The maximum achieved number of bits acheived varies with the supply voltage as shown in Figure 19 and at 300 mV a 32-bit carry chain can be implemented.

The transistor sizing is adjusted to accommodate the change

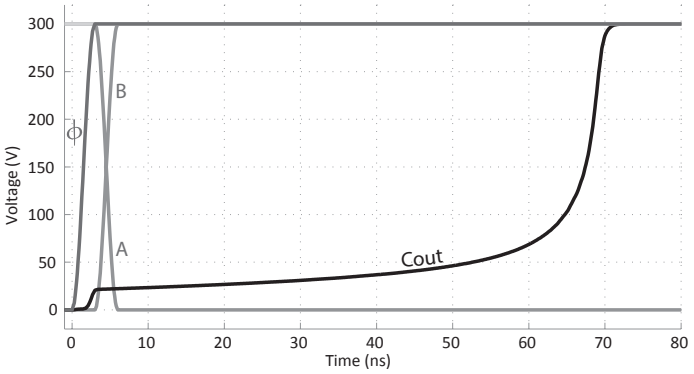


Fig. 18. Drifting problem of the SULVC output.

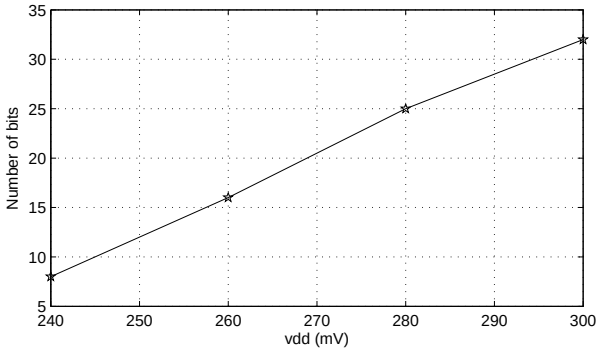


Fig. 19. Number of carry circuits or bit obtained from carry chain when supply voltage is varied.

in NMOS/PMOS mobility difference with changed supply voltage. In these simulations, the NMOS evaluation transistor size is kept minimum sized and the PMOS evaluation transistor length is changed to match the NMOS drive strength.

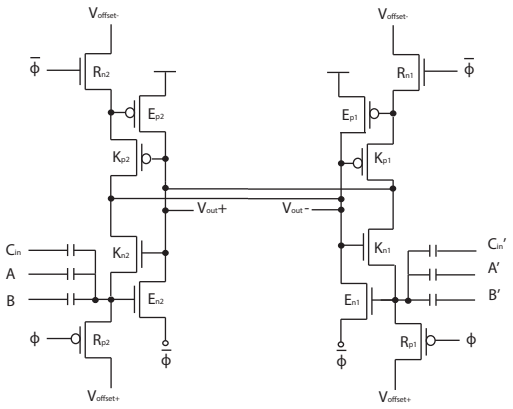
B. Differential Carry Gate

In order to overcome the challenges with robustness and drifting of the SULVC circuit, a differential approach is a possible solution. A Static Differential Ultra Low Voltage Carry (SDULVC) as shown in Figure 20 is designed in exactly the same manner as the SULVC, however, with differential inputs and outputs. The differential nature of the circuit makes it less prone to drifting and eliminates the need for level-to-edge converters it can be sized to allow a single edge without causing an output transition. The outputs of the proposed circuit are precharged to the same level during the precharge phase, however, it yields a differential output during the evaluation phase. So, instead of employing an inverter to obtain the carry bit we can read it from the opposite end of the circuit, i.e., in an N-type SDULVC if inputs A, B, and C are applied to E_{n2} output can be read from V_{out-}. Figure 20 demonstrates the design of an SDULVC circuit. The backgate of the keeper transistors of these circuits are connected to the floating gate to achieve maximum robustness.

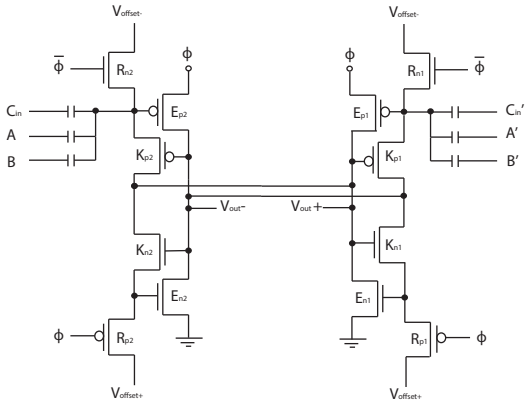
$$V_{fg} = V_{initial} + k_{in} \cdot V_{in} \text{where } k_{in} = \frac{\sum_{i=1}^{n} C_{innHigh_i}}{C_{total}} \quad (4)$$

The variable *'i'* in (4) denotes the index of the input and the *'n'* denotes fan-in. $V_{initial}$ is the precharge voltage level of the floating gate. C_{inHigh} is a combination of input capacitors with a high (rising) input.

Considering an example of an N-type SDULVNC, we can calculate the voltage level of the floating gate using (4). We assume that the diffusion capacitance is equal to the input capacitance and that the supply voltage is equal to the input voltage. The load capacitance introduced by the keeper's back-gate connection to the floating node should also be considered



(a) N type SDULVC.

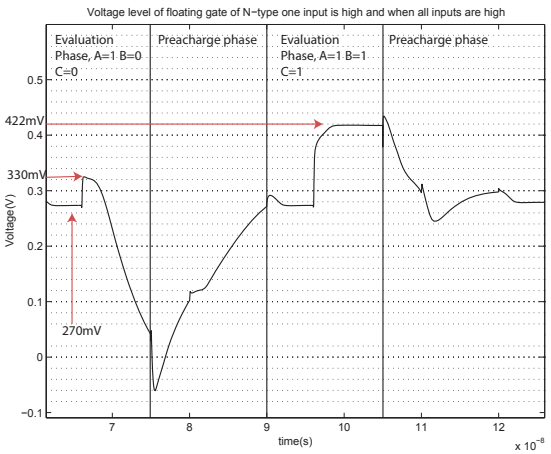


(b) P type SDULVC.

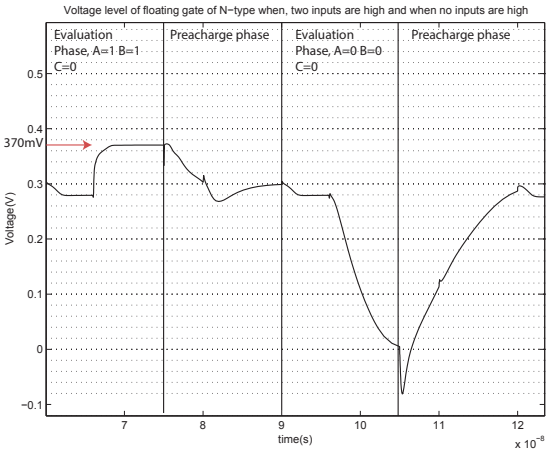
Fig. 20. SDULVC circuit.

and in this paper is assumed to be equal to the input capacitor as well.

Our calculation in (4) gives us a theoretical idea of the voltage level at V_{fg} (floating gate voltage). In the real world, the capacitance size might not be exactly the same as our assumption and depends on transistor size and many other factors like process variation and mismatch. The simulation results of the voltage levels for the floating gate in Figure 21 shows that the floating node is precharged to 270mV. Equation (4) yields an analytical result for the floating gate input of 330 mV, 390mV and 450 mV for one, two and three high inputs, respectivly. The simulation results in Figure 21a shows that the voltage level of the floating node gets to 330mV for a single rising input transition and to 420mV when all inputs are high. These results are marginally different from the calculated values. This is possibly due to the assumptions on capacitance sizes. Figure 21a shows that if only one input gets high, the keeper transistor turns on and discharges the floating node. The reason for this is that the transition at the input, i.e., 60mV, is not sufficient to produce enough current at the output. Figure 21b shows the results for two high inputs and all low inputs.



(a) Voltage level of input floating gate of an N-type SDULVC/SDULVC when A=1 B=0 and C=0, and when A=1 B=1 C=1.



(b) Voltage level of input floating gate of an N-type SDULVC/SDULVC when A=1 B=1 and C=0, and when A=0 B=0 C=0.

Fig. 21. Voltage level of floating gate of N-type at supply voltage of 300mV.

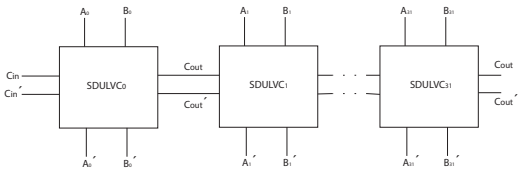


Fig. 22. 32 bit ULV carry chain.

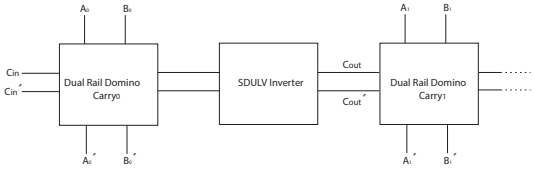
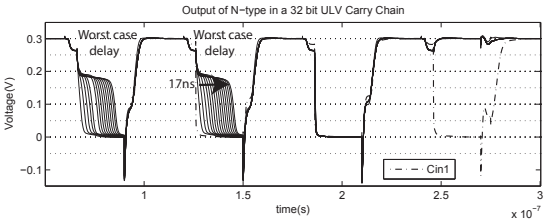
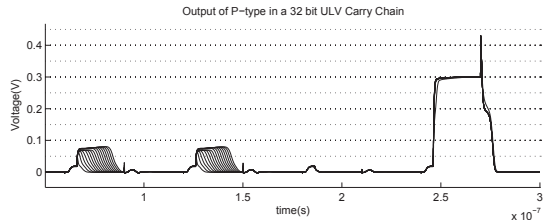


Fig. 23. Implementation of hybrid Dual rail domino carry.



(a) output ULV carry chain N-type.



(b) output ULV carry chain P-type.

Fig. 24. Simulation result of 32 bit ULV carry chain at a supply voltage of 300mV.

VII. Simulation

A. 32 bit SDULVC chain

A 32 bit ULV carry chain is implemented using 32 SDULVC circuits connected in a chain or NP domino fashion shown in Figure 22. Figure 24 shows the simulation response of a 32 bit ULV carry chain. The propagtion delay of this carry chain is 17ns. In order to compare the SDULVC to other carry gate topologies, a dual rail domino carry gate designed in a hybrid fashion, i.e., instead of utilizing conventional inverters at the output, the Static Differential ULV inverter presented in [14] and a conventional NP Domino Dual Rail carry is used. Compared to the hybrid dual rail domino carry (HDRDC) chain shown in the Figure 23 the SDULVC chain is almost 10× faster and compared to a Conventional Dual Rail Domino Carry (CDRDC) this is closer to 35×. These numbers are based on the propagation delay for the carry bit through the chain, which is 166ns for the hybrid dual rail domino carry and 636ns for the conventional dual rail domino carry, all at 300 mV.

The robustness of the SDULVC can be analyzed by looking at the simulation response shown in Figure 24b. The plot for the worst case delay scenario, i.e., A=1, B=0, C=0, exhibits that due to a delayed carry bit and the early arrival of inputs, A and B, a marginal transition at the output occurs. However, once the carry bit has arrived, the output shifts to its final value. Average transition at the output for a P-type and N-type SDULVC when waiting for the carry bit is between 70mV and 100mV. This can be seen as a problem for the noise margin and power consumption. The output manages to return to the right final value due to synchronisation of keeper signals with the input. Therefore, the issue of noise margin can be ignored by concluding that the final value can be read at the end of the evaluation phase.

Figure 25 shows the delay of an SDULVC chain compared

TABLE III. DIMENSIONS OF HYBRID DUAL RAIL DOMINO CARRY GATE

	Supply voltage variation	Width of dual rail domino evaluation transitor/Width of SDULVN evaluation transitor	Length of dual rail domino evaluation transitor/Length of SDULVN evaluation transitor	Width of dual rail domino precharge transitor/Width of SDULVN precharge transitor	Length of dual rail domino precharge transitor/Length of SDULVN precharge transitor
Size 1	270mv-400mv	4×	3.3×	1×	1×
Size 2	220mv-400mv	6.67×	8.3×	3.33×	35×

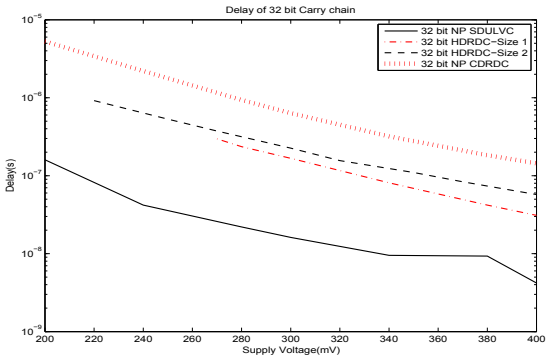


Fig. 25. Delay of 32 bit SDULVC and hybrid dual rail domino at varried supply voltage.

to an HDRDC and a CDRDC chain. Table III shows that the transistor size has to be increased in order to increase the ON current of the device [3] and be able to decrease the supply voltage for HDRDC. Table IV shows the minimum operating frequency required for the clock to simulate SDULVC, HDRDC and CDRDC at different supply voltages.

B. PDP and EDP of SDULVC chain

PDP charachteristics of a circuit highlights its efficiency with respect to power consumtion. A low PDP means a more energy efficient circuit. Although the ULV circuits presented in this paper are power hungry, it still manages to maintain its PDP at approximately the same level as conventional circuits where the power consumption is lower. The average power of the HDRDC and the SDULVC is $0.347\mu W$ and $1.28nW$ respectively at a supply voltage of 300 mV. This indicates that the power consumption of HDRDC is up to $3\times$ better than ULV circuits. However, at the same supply voltage the ULV circuit is $10\times$ faster than the HDRDC. Therefore, the ULV

TABLE IV. MINIMUM CLOCK OPERATING FREQUENCY FMIN REQUIRED BY THREE TOPOLOGIES

Supply Voltage (mV)	f_{min} for SDULVC (MHz)	f_{min} for HDRDC-Size 1 (MHz)	f_{min} for HDRDC-Size 2 (MHz)	f_{min} for CDRDC
200	1.6	-	-	0.08
220	-	-	-	-
240	3.125	-	-	0.217
250	-	-	0.83	-
270	-	1.66	1.225	-
280	6.25	-	-	0.5
300	8.33	2.3	2	0.769
320	-	-	2.27	-
340	16.66	5.5	3.33	1.562
380	21	10	5.55	2.5
400	23.8	60	7.692	3.33

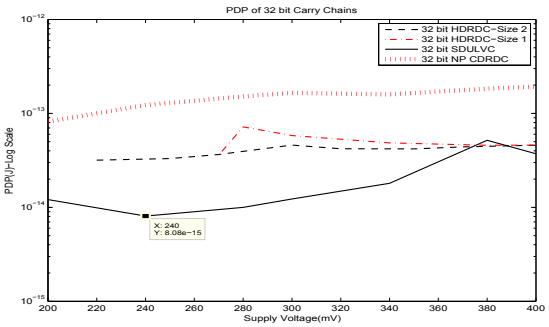


Fig. 26. PDP of 32 bit carry chains.

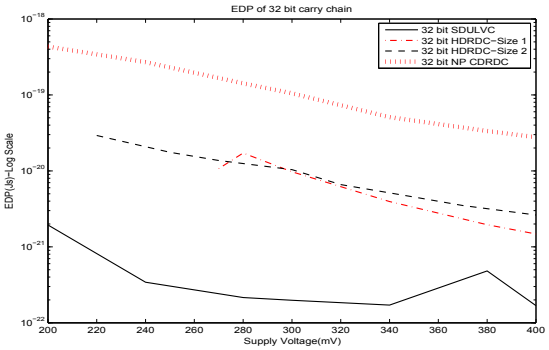


Fig. 27. EDP of 32 bit carry chains.

circiuts are still more energy efficient. Figure 26 shows PDP of three different 32 bit carry chain topologies at varied supply voltage. The minimum energy point of the 32 bit SDULVC carry chain is found at 240 mV.

Another important charachteristic of any circuit is EDP. It demonstrates enhanced speed of any circuit with respect to its energy effieciency. It is obvious that circuits with better propagation delay shall stand out in this characteristic. Figure 27 shows the EDP of three carry chains and the evident performance advantages of SDULVC circuits.

VIII. CONCLUSION PART 2

In this paper, a new ULV carry circuit has been presented and performance enhancements have been demonstrated. The ULV carry circuits are better than conventional topologies in both speed and energy efficiency, shown by comparing the SDULVC to the HDRDC and CDRDC circuit topologies. A credible conclusion is that a static differential dynamic ULV carry circuit is a favorable choice when speed and robustness at low voltages are important.

REFERENCES

[1] A. Majeed, H. Bechmann, and Y. Berg, "Novel High Speed and Robust Ultra Low Voltage CMOS NP Domino Carry gate," in *CENICS 2014, The Seventh International Conference on Advances in Circuits, Electronics and Micro-electronics*, 2014. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=cenics_2014_1_10_60004

[2] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power cmos digital design," *Solid-State Circuits, IEEE Journal of*, vol. 27, no. 4, pp. 473 –484, Apr 1992.

[3] M. Alioto, "Ultra-low power vlsi circuit design demystified and explained: A tutorial," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, no. 1, pp. 3 –29, Jan. 2012.

[4] Y. Berg and O. Mirmotahari, "Ultra low-voltage and high speed dynamic and static cmos precharge logic," in *Faible Tension Faible Consommation (FTFC), 2012 IEEE*, June 2012, pp. 1 –4.

[5] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, L. Nazhandali, T. Austin, D. Sylvester, and D. Blaauw, "Exploring variability and performance in a sub-200-mv processor," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 4, pp. 881 –891, April 2008.

[6] Y. Berg, D. Wisland, and T.-S. Lande, "Ultra low-voltage/low-power digital floating-gate circuits," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 46, no. 7, pp. 930–936, 1999.

[7] Y. Berg and O. Mirmotahari, "Static ultra low-voltage and high performance cmos nand and nor gates," *Rn*, vol. 1005, p. 2.

[8] N. Weste and D. Harris, *Integrated Circuit Design*. Pearson Education, Limited, 2010. [Online]. Available: http://books.google.no/books?id=gIAIQgAACAAJ

[9] M. Alioto and G. Palumbo, "Impact of supply voltage variations on full adder delay: Analysis and comparison," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 14, no. 12, pp. 1322–1335, 2006.

[10] ——, "Very high-speed carry computation based on mixed dynamic/transmission-gate full adders," in *Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conference on*, 2007, pp. 799–802.

[11] Y. Berg and O. Mirmotahari, "Ultra low voltage and high speed cmos carry generate circuits," in *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on*, 2009, pp. 69–72.

[12] Y. Berg, "Ultra low voltage static carry generate circuit," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 2010, pp. 1476–1479.

[13] Y. Berg, S. Aunet, O. Naess, O. Hagen, and M. Hovin, "A novel floating-gate multiple-valued cmos full-adder," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 1, 2002, pp. I–877–I–880 vol.1.

[14] Y. Berg and O. Mirmotahari, "Static differential ultra low-voltage domino cmos logic for high speed applications," *North atlantic university union: International Journal of Circuits, Systems and Signal Processing.*, pp. 269–274.

Stochastic Models for Quantum Device Configuration and Self-Adaptation

Sandra König* and Stefan Rass†
*Digital Safety & Security Department, Austrian Institute of Technology, Klagenfurt, Austria
Sandra.Koenig@ait.ac.at
†Department of Applied Informatics, System Security Group,
Universität Klagenfurt, Universitätsstrasse 65-67, 9020 Klagenfurt, Austria
stefan.rass@aau.at

***Abstract*—Quantum carriers of information are naturally fragile and as such subject to influence by various environmental factors. Cryptographic techniques that exploit the physical properties of light particles to securely transmit information strongly hinge on a proper calibration and parameterizations to correctly distinguish natural distortions from artificial ones, the latter of which would indicate the presence of an attacker. Consequently, it is necessary and useful to know how environmental working conditions influence a quantum device so as to optimize its operational performance (say, the qubit transmission or error rates, etc.). This work extends a previous copula-based modeling approach to build a stochastic model of how different device parameters depend on one another and how they influence the device performance. We give a full detailed practical description of how a model can be fit to the data, how the goodness of fit can be tested, and how the quantities of interest for a self-calibration can be obtained from the resulting stochastic models.**

Keywords–stochastic modeling; copula; estimation; goodness of fit; quantum network; quantum devices; statistics

I. Introduction

Quantum key distribution (QKD) is a technique that exploits light (particles) as carrier of information. The natural fragility of such a carrier naturally ties even passive eavesdropping attempts to an unavoidable increase of errors that is detectable for the user(s) of the quantum channel. To reliably indicate the presence of an adversary by classifying some errors as being artificial and distinguishing these from natural error rates, several environmental factors have to be taken into account to compute the expected channel characteristics (error rate, noise, etc.) when the transmission is unimpeded. To this end, [1] proposed the use of copula models to capture the influence of environmental factors on the performance characteristic of a QKD device, most importantly, the quantum bit error rate, which indicates the presence or absence of an intruder.

Physically, the fact that any access to the channel induces errors is implied by the impossibility of creating a perfect copy of a single photon. This fundamental result of quantum physics was obtained by [2].

Recent experimental findings on the quantum key distribution network demonstrated as the result of the EU project SECOQC (summarized in [3]) raised the question of how much environmental influences affect the "natural" quantum bit (qubit) error rate (QBER) observed on a quantum line that is not under eavesdropping attacks. A measurement sample reported in [4] was used to gain first insights in the problem, but the deeper mechanisms of dependency between QBER and the device's working conditions have not been modeled comprehensively up to now.

The desire of having a model that explains how the QBER depends on environmental parameters like temperature, humidity, radiation, etc. is motivated by the problem of finding a good calibration of QKD devices, so that the channel performance is maximized. Unfortunately, with the QBER being known to depend on non-cryptographic parameters, it is difficult to give reasonable threshold figures that distinguish the natural error level from that induced by a passive eavesdropping. We spare the technical details on how a QBER threshold is determined for a given QKD protocol here (that procedure is specific for each known QKD protocol and implementation), and focus our attention on a statistical approach to obtain a model of interplay between the qubit error rate and various environmental parameters. More precisely, our work addresses the following problem: *given the current working conditions of a QKD device, what would the natural qubit error rate be, whose transgression would indicate the presence of an eavesdropper?* The basic intention behind this research is aiding practical implementations of QKD-enhanced networks, where our models provide a statistically grounded help to react on changing environmental conditions.

For that purpose, we utilize a general tool of probability theory, a copula function, which is an interdependency model as contrasted to the parameter model (probability distribution of a single environment parameter). In that regard, we outline in Section II the basics of copula theory to the extent required here. This is to quickly get to the point where we can give effective methods to infer an expected qubit error rate upon known external influence parameters.

The remainder of this work is structured as follows: after theoretical groundwork in Section II, we move on by showing how to use empirical data (measurements) drawn from a given device to construct an interdependency model that explains how the QBER and other variables mutually depend on each other. Section IV then describes how to single out the QBER from this overall dependency structure towards computing the expected error rate from the remaining variables. The concluding Section V summarizes the procedure and provides final remarks.

Related Work

Surprisingly, there seem to be only a few publications paying attention to statistical dependencies of cryptographic parameters and the working conditions of a real device, such as [4], [5]. While most experimental implementations of QKD, such as [3], [6]–[9] give quite a number of details on device parameters, optimizations of these are mostly out of focus. An interesting direction of research is towards becoming "device-independent" [10], [11], which to some extent may relieve issues of hacking detection facilities, yet leaves the problem of optimal device configuration nevertheless open. The idea of self-adaptation is not new and has already seen applications in the quantum world [12]–[14] including the concept of copulas, applications of the latter to the end of self-adaption remain a seemingly new field of research. Copulas have been successfully applied to various problems of explaining and exploiting dependencies among various risk factors (related to general system security [15], [16]), and the goal of this work is taking first steps in a study of their applicability in the yet unexplored area of self-configuring quantum devices.

II. Preliminaries and Notation

We denote random variables by uppercase Latin letters $(X, Z, \ldots)$, and let matrices be uppercase Greek or bold-printed Latin letters $(\Sigma, \mathbf{D}, \ldots)$. The symbol $X \sim F(x)$ denotes the fact that the random variable X has the distribution function F. For each such distribution, we let the corresponding lower-case letter denote its density function, i.e., f in the example case.

For self-containment of our presentation, we give a short overview of the most essential facts about copulas that we are going to use, as for a more detailed introduction we refer to [17].

Definition II.1. *A* copula *is a (n-dimensional) distribution function $C : [0,1]^n \to [0,1]$ with uniform marginal distributions.*

Especially, a copula satisfies the following properties:

Lemma II.1. 1) *For every $u_1, \ldots, u_n \in [0,1]$, $C(u_1, \ldots, u_n) = 0$ if at least one of the arguments is zero and*
2) *$C(u_1, \ldots, u_n) = u_i$ if $u_j = 1$ for all $j \neq i$.*

A family of copulas that leads to handy models in higher dimensions is known as the family of Archimedean copulas, of which many extensions exist.

Definition II.2. *An* Archimedean copula *is determined by the so called* generator function $\phi(x)$ *via*

$$C(u_1, \ldots, u_n) = \phi^{-1}(\phi(u_1) + \ldots + \phi(u_n)). \qquad (1)$$

The generator function $\phi : [0,1] \to [0,\infty]$ has to satisfy $\phi(1) = 0$ and $\phi(\infty) = 0$, furthermore, ϕ has to be n-monotone*, i.e., to be differentiable up to order $n-2$ with $(-1)^{n-2}\phi^{(n-2)}(t)$ being nondecreasing and convex and*

$$(-1)^i \phi^{(i)}(t) \geq 0 \text{ for } 0 \leq i \leq n-2$$

for all $t \in [0, \infty)$.

As one of the cornerstones in copula theory, *Sklårs theorem* connects these functions to the relationship between n univariate distribution functions and their joint (multivariate) distribution:

Proposition II.2. *Let the random variables $X_1, \ldots, X_n$ have distribution functions $F_1, \ldots, F_n$ respectively and let H be their joint distribution function. Then there exists a copula C such that*

$$H(x_1, \ldots, x_n) = C(F_1(x_1), \ldots, F_n(x_n)) \qquad (2)$$

for all $x_i, \ldots, x_n \in \mathbb{R}$. If all the F_is are continuous, then the copula C is unique.

The usefulness of this result lies in the fact that the joint distribution function of $X_1, \ldots, X_n$ can be decomposed into n univariate functions $F_1, \ldots, F_n$ that describe the behaviour of the individual variables and another component (namely the function C) that describes the dependence structure, which allows to model them independently.

Conversely, it is also possible to extract the dependence structure from the marginal distributions F_i and the joint distribution H via

$$C(u_1, \ldots, u_n) = H(F_1^{-1}(u_1), \ldots, F_n^{-1}(u_n)) \qquad (3)$$

where $F_i^{-1}(u)$ denotes the pseudo-inverse of $F_i(x)$, which is given by $F_i^{-1}(u) = \sup\{x | F_i(x) \leq u\}$. A special case of this connection between Copula and random variables leads to an alternative characterization of independence, which is usually written as $H(x_1, \ldots, x_n) = F_1(x_1) \cdot \ldots \cdot F_n(x_n)$.

Example II.3. *If the (unique) copula from (3) turns out to be the product copula $C(u_1, \ldots, u_n) = u_1 \cdot \ldots \cdot u_n$, then the random variables $X_1, \ldots, X_n$ are independent.*

III. A Copula Model of the QKD network

A. Summary of the Data

A summary description of the measurement data obtained from an implemented QKD network in Vienna [3] can be found in [5]. The following quantities were measured and are used here (abbreviation in brackets): qubit error rate in percentage terms (QBER), air temperature (TEMP), relative humidity (HUM), sunshine duration in seconds (DUR), global radiation in watt/m^2(RAD).

Since we are here focusing on the relationship between QBER and environmental quantities, we only use data that were measured on the same device to avoid getting biased results. The quantiles of our sample of size $n = 276$ are displayed in Table I.

Throughout the rest of the paper, let $\mathbf{D}$ denote the data matrix that comprises the entirety of samples as a table with headings corresponding to the row labels in Table I. Thus, the matrix $\mathbf{D}$ is of shape $(n \times 5)$ for our $n = 276$ samples, and has entries $(X_1, \ldots, X_5)$ modeling the measurements of (QBER, TEMP, HUM, DUR, RAD) as random variables.

TABLE I. Quantiles of measured quantities

	min	$q_{0.25}$	median	$q_{0.75}$	max
QBER	0.98	1.33	1.47	1.63	2.12
TEMP	117.00	134.75	148.00	163.00	184.00
HUM	71.00	80.00	84.00	91.00	93.00
DUR	0.00	0.00	0.00	0.00	600.00
RAD	0.00	0.00	0.00	146.00	539.00

B. Building up a Model

Mainly interested in the dependence structure, we do not make explicit assumptions about the distributions of the quantities each, but rather use $U(0,1)$-distributed pseudo-observations $U_1, \ldots, U_n$ transformed from the empirical distributions of the quantities. A basic first choice is to consider a multidimensional copula C that models the joint distribution H of all the quantities via $H(x_1, \ldots, x_n) = C(U_1, \ldots, U_n)$. Fitting a copula is usually done by maximizing the log-likelihood function

$$\ell(x_1, \ldots, x_n) = \log\left[c\left(u_1, \ldots, u_n\right)\right],$$

with c denoting the density of the copula C. In a general setting, this can easily become infeasible in our five-dimensional case, so we first choose a parametric family C_θ of copulas and then seek the parameter θ that maximizes the one-dimensional function

$$\ell(\theta) = \log\left[c_\theta\left(u_1, \ldots, u_n\right)\right].$$

As for the parametric family, we first choose the *Gumbel copula*, which is generated by $\phi(t) = (-\ln(t))^\theta$, yielding

$$C(u_1, \ldots, u_n) = \exp\left\{-[(-\ln(u_1))^\theta + \ldots + (-\ln(u_n))^\theta]^{1/\theta}\right\}.$$

A p-value of zero clearly shows that this model is not describing the data properly.

The above model is simple to construct and to use but it also has its weaknesses: firstly it describes the behaviour of five random variables with just one number and secondly its components are all exchangeable. Taking a closer look at the pairwise correlations of the considered quantities (Figure 1), we see that this exchangeability is not fulfilled in our case.

To take care of possibly different correlations among the occurring variables, we consider a more flexible model called *nested* copulas (sometimes also called hierarchical copulas), which is often used in finance, see for example [15]. The basic idea of a nested copula model is to use several copulas at different levels to describe the relation between the variables.

For clarity of such a hierarchically constructed probability distribution we use a graphical tree-notation like shown in Figure 2 to "depict" the (otherwise complicated) distribution function. To formally specify the latter, we introduce some notational conventions: at each level $\ell \in 1, \ldots, L$ (counting bottom-up in the hierarchy tree) we have n_ℓ copulas, where $C_{\ell,j}$, $j \in 1, \ldots, n_\ell$, is the $j-$th copula at level ℓ. Further, every copula $C_{\ell,j}$ has dimension $d_{\ell,j}$ that gives the number of arguments u_i that directly or indirectly enter this copula.

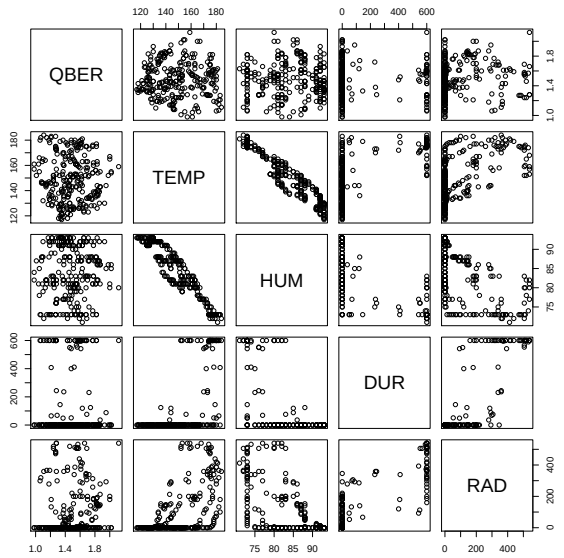


Figure 1. Pairwise correlations among variables

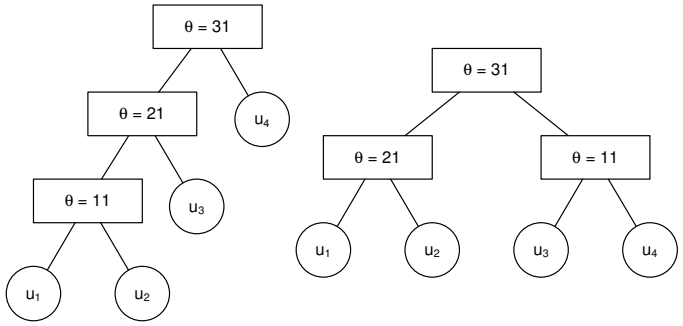


Figure 2. Fully nested vs. partially nested copula

For the sake of illustration only, two example cases of nesting are shown in Figure 2 for the four-dimensional case: the fully nested copula, which adds one dimension at each step (left side) and a partially nested copula where the number of copula decreases at each level (right side). Our task in the following is finding out the particular structure of nesting of the random variables, based on the empirical data available (on which, e.g., Figure 1 is based on).

Formally, a fully nested copula is defined by

$$\begin{aligned} &C(u_1, \ldots, u_n) = \\ &\phi_{n-1}^{-1}[\phi_{n-1}(\ldots[\phi_2(\phi_1^{-1}[\phi_1(u_1) + \phi_1(u_2)] + \phi_2(u_3)] \\ &+ \ldots + \phi_{n-2}(u_{n-1})) + \phi_{n-1}(u_n))], \end{aligned} \quad (4)$$

where the occurring generator functions $\phi_1, \ldots, \phi_{n-1}$ may come from different families of Archimedean copulas.

All in all, the dependence structure is determined by $n-1$ parameters (instead of just one as in the model above) and there are $\frac{n(n-1)}{2}$ different bivariate margins.

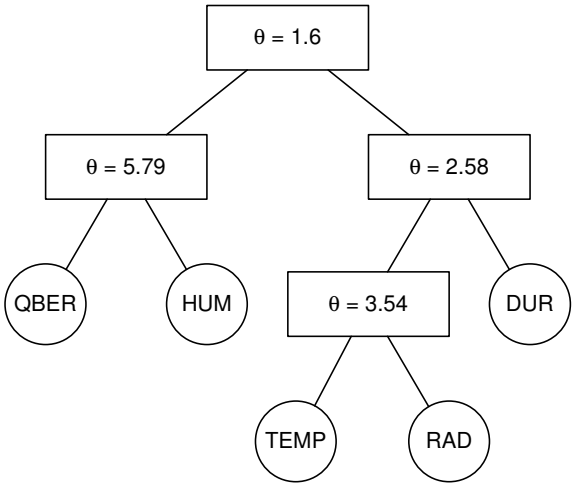


Figure 3. Dependence structure for HAC model

The partially nested copula may be defined similarly, for reasons of clarity and comprehensibility we here give the expression for $n = 4$, corresponding to the case shown in the right side of Figure 2:

$$C(u_1,u_2,u_3,u_4) = \phi_{21}^{-1}[\phi_{21}(\phi_{11}^{-1}[\phi_{11}(u_1)+\phi_{11}(u_2)] + \phi_{21}(\phi_{12}^{-1}[\phi_{12}(u_3)+\phi_{12}(u_4)])], \quad (5)$$

where the generator ϕ_{ij} is from the jth copula on the ith level, usually denoted by C_{ij}.

Finding a suitable nested copula model may quickly become laborious since one might have to check all possible subsets of variables and compare the goodness of fit of the corresponding estimated copula. Handling this problem in `R`, one may use the package `HAC`, introduced in [18]. In our case, we find that a suitable model consists of four two-dimensional Gumbel copulas, which are defined as follows:

Definition III.1. *A* Gumbel copula *is an Archimedean copula that is generated by*

$$\phi(t) = (-\ln(t))^\theta$$

for $\theta \geq 1$. *In the two-dimensional case, the copula is explicitly given by*

$$C(u,v) = \exp\left[-\left((-\ln(u))^\theta + (-\ln(v))^\theta\right)^{\frac{1}{\theta}}\right] \quad (6)$$

for $u, v \in [0,1]$.

The dependence structure between the considered quantities is shown in Figure 3.

It is known that in a nested copula model with a Gumbel generator the parameters have to decrease with the level (see [15] for fully nested copulas and [19] for the general case). Since in our case the parameters on the upper levels are rather close, we consider a modification of this model by allowing to aggregate Copulas whose parameters do not differ too much. A justification for this approach is the close relation between the parameter θ of the generator and Kendall's tau τ, which is connected to copulas via

$$\tau = 4\int_{[0,1]^2} C(u,v)dC(u,v) - 1. \quad (7)$$

For Archimedean copulas with generator function $\phi(t)$, it was shown in [17] that (7) simplifies to

$$\tau = 1 + 4\int_0^1 \frac{\phi(t)}{\phi'(t)}dt, \quad (8)$$

which for the Gumbel copula leads to

$$\tau = 1 - 4\int_0^1 \frac{(-\log(t))^\theta \cdot t}{\theta(-\log(t))^{\theta-1}}dt = 1 - \frac{1}{\theta}.$$

Hence, if the parameters of two subsequent copulas are close, so is their dependence when characterized through Kendall's τ and it might be beneficial to model the affected variables with only one copula.

These calculations can conveniently be done with the help of `R`s function `estimate.copula` from the `HAC` package. This function estimates both the structure of the hierarchical copula as well as all corresponding parameters for several different Archimedean copula families. The fitting is most commonly done by Maximum Likelihood or quasi Maximum Likelihood. A simple improvement of this estimation is given in appendix A. Once a suitable model has been found the `HAC` package also allows to compute the density or the cumulative distribution function for a sample from the corresponding hierarchical copula, which will be used to test the goodness of fit as described below.

C. Goodness of fit test for Hierarchical Archimedian Copulas

In order to get an impression on how suitable each of the above models is, we adapted the bootstrapping goodness of fit test [20] that was used in the case of a one-parametric copula to the estimation of nested copulas.

We leave the details of the testing algorithm to the literature [20], and confine ourselves to a brief description here and an implementation outline in appendix A, to make things at least plausible: in general, we would consider a model F_{fit} as a "good fit", if its Cramer-van Mises statistic being the integrated squared difference between F_{fit} and the true distribution is "small". The exact numeric magnitude (limit) for a value to be "small" in that sense is unclear, however, and must be fixed first. This is done by bootstrapping: to get an idea of when a deviation is "small" (good fit) or "large" (bad fit), we draw artificial data samples from the estimated model F_{fit}, and re-fit another model $F_{\text{re-fit}}$ to the so-obtained data. Since the new model is based on data coming from F_{fit}, its deviation from F_{fit}, i.e., its Cramer-van Mises statistics, must be "small" in the sense we need (no matter of its particular numerical magnitude). Given this bootstrapping threshold for

small deviations, we can then move on testing the real data against the fitted model F_{fit}, computing another Cramer-van Mises statistic. This final value is then compared to be larger or smaller than the previously obtained bootstrapping threshold (limit for small deviations) to obtain an empirical p-value of the test.

In our first 200 trial tests, each of which with a sample size of $N = 1000$ and a confidence level of 0.95, we never got a positive p-value if the tolerance was set to zero. When copulas are allowed to be aggregated, a p-value of 0.014 was found once, which still leads to rejection of the null hypothesis that the data at hand are drawn from a distribution given through this copula. This indicates that some preconditioning of the data matrix might be necessary to get a good fit. One solution for such a preprocessing is described in the next section.

D. Preconditioning Towards Better Fits

As indicated by our quantum network data, it may occasionally be the case that none of the tried copula-models models the data satisfactorily. More precisely, existing software packages for copula fitting (such as `HAC` in `R`) assume *positive correlations* between all variables of interest. Unfortunately, our experimental QKD prototype supplied data exhibiting *negative* correlations amongst some of the observed variables.

In order to fix this, we can apply a linear transformation $\mathbf{M}$ to the data matrix $\mathbf{D}$ in order to make all pairwise correlations in the transformed data matrix $\mathbf{M} \cdot \mathbf{D}$ strictly positive. To this end, consider the Cholesky-decomposition of the covariance matrix Σ of the data $\mathbf{D}$, given as $\Sigma = \mathbf{U}^T \cdot \mathbf{U} = \mathbf{U}^T \cdot \mathbf{I} \cdot \mathbf{U}$. By the linearity properties of covariance, it is easy to check that the covariance matrix of $\mathbf{D} \cdot \mathbf{U}^{-1}$ is the identity matrix, having zero-correlations among all pairwise distinct variables. It is then a simple matter of multiplication with another invertible matrix (with low condition number to avoid numerical round-off-errors in the inverse transform) with all strictly positive entries to artificially introduce positive correlations, as required in the copula fitting process. Given such a matrix $\mathbf{A}$, the final linear transformation takes the form

$$\mathbf{D}' := \mathbf{D} \cdot (\mathbf{U}^{-1} \cdot \mathbf{A}), \tag{9}$$

thus our pre-conditioning transformation matrix is $\mathbf{M} := \mathbf{U}^{-1} \cdot \mathbf{A}$, where $\mathbf{U}$ comes out of the Cholesky decomposition of the original covariance matrix Σ, and $\mathbf{A}$ can be chosen freely, subject to only positive entries and a good condition number (for numerically stable invertibility).

In our experiments, we used a bootstrap fitting with tolerance $\varepsilon = 0.4$. We constructed $\mathbf{A}$ as a 5×5-matrix having Gamma-distributed entries (with shape-parameter 5 and scale-parameter $1/2$). In 5 out of 200 trials, the p-value after preconditioning with $\mathbf{M} = \mathbf{U}^{-1}\mathbf{A}$ was larger than 0.05. The best fit giving $p = 0.613$ was obtained under the transformation coefficients (rounded to three decimals after the comma)

$$\mathbf{A} = \begin{pmatrix} 0.122 & 4.444 & 0.378 & 1.634 & 4.384 \\ 0.650 & 0.870 & 1.321 & 0.941 & 2.293 \\ 0.606 & 3.326 & 0.763 & 2.172 & 2.102 \\ 2.534 & 0.415 & 2.055 & 1.969 & 1.659 \\ 2.668 & 2.031 & 3.590 & 2.241 & 1.015 \end{pmatrix},$$

whose condition number is $\|\mathbf{A}\|_2 \cdot \|\mathbf{A}^{-1}\|_2 \approx 24.4945$, and determinant given as $\det(\mathbf{A}) \approx 29$, thus indicating good numerical stability for the inverse transformation.

In a second run of 200 experiments, we lowered the tolerance $\varepsilon = 0$, and did the preconditioning as before. This time, we got 20 out of 200 trials with a positive p-value, although only in three cases, our fit was accepted at $p > 0.05$. The best fit was obtained at $p = 0.536$, showing that the preconditioning works equally well with more complex hierarchical structures due to lower tolerance levels.

This transformation is applied *before* the copula fit, and must be carried through the derivation of predictive densities when obtaining a fit. More specifically, with the preconditioned random vector being $Y = \mathbf{M} \cdot X$ to which we could fit a density function (copula model) f_Y, then the original data X is distributed with density function

$$f_X(\mathbf{x}) = f_Y(\mathbf{M} \cdot \mathbf{x}) \cdot |\det(\mathbf{M})|\,, \tag{10}$$

where the determinant is a constant, and not even the inversion of the transformation matrix $\mathbf{M}$ is actually required.

The preconditioning does come at the drawback of loosing the copula-representation of the joint density, which simplifies the subsequent construction of conditional (predictive) densities. Without this representation, i.e., when one is forced to work with a model of the form (10), computing conditional and predictive densities works via the definition, i.e.,

$$\begin{aligned} f(x_1|x_2,\ldots,x_n) &= \tfrac{f(x_1,\ldots,x_n)}{f(x_2,\ldots,x_n)} \\ &= \tfrac{f(x_1,\ldots,x_n)}{\int_{\mathbb{R}} f(x_1,\ldots,x_n)dx_1}, \end{aligned} \tag{11}$$

where $f(x_1,\ldots,x_n)$ is the joint density obtained through (10) and the marginal density can be computed by (numerical) integration (e.g., Monte-Carlo algorithms; cf. [21]), which can be complex. To ease matters, we thus assume the model of the joint variables to take the form (2) as in proposition (II.2).

As an open issue, moreover, it remains interesting to find better ways than simple try-and-error to find a preconditioning matrix $\mathbf{A}$ that gives better fits than the plain data would do. Moreover, we believe that this trick may be of independent interest and use in other applications of copula theory, not limited to statistical descriptions of quantum key distribution devices.

IV. Prediction of QBER Rates

Based on a model that describes the relationship between QBER and the environmental quantities, we look for a prediction of the QBER when all the other quantities are known. Having an idea of what values are to be expected, one might

suspect an adversary to be present if these values are clearly exceeded. An essential ingredient to find a prediction is the conditional density, as it shows which values are likely in a given situation, that is, we seek the density of QBER conditional on all the other environmental parameters, i.e., the function

$$f(\text{QBER}|\text{TEMP}, \text{HUM}, \text{DUR}, \text{RAD}).$$

Section IV-A describes the general technique to compute the sought density, taking QBER as the n-th variable x_n in the upcoming descriptions. We stress that, however, the method is equivalently applicable to predict any other variable than QBER, too.

A. Computing Conditional Densities via Copulas

In the case where all the marginals and the copula are continuous, it holds for the transformed variables $u_i = F_i^{-1}(x_i)$ by the independence of copula and margins that

$$f(x_1, \ldots, x_n) = f_1(x_1) \cdot \ldots \cdot f_n(x_n) \cdot c_n(u_1, \ldots, u_n),$$

where $c_n(u_1, \ldots, u_n)$ denotes the density of the n-dimensional copula $C_n(u_1, \ldots, u_n)$ and f_i denotes the density of the marginal distribution F_i.

Example IV.1. *In the case of independent random variables, the above formula yields $c_n(u_1, \ldots, u_n) = 1$, which is the derivative of the independence copula $C_n(u_1, \ldots, u_n) = u_1 \cdots u_n$ from Example II.3.*

With this decomposition, the conditional density is obtained as

$$f(x_n|x_1, \ldots, x_{n-1}) = f_n(x_n) \frac{c_n(u_1, \ldots, u_n)}{c_{n-1}(u_1, \ldots, u_{n-1})} \qquad (12)$$

for $u_i = F_i(x_i)$. Using (12) to compute the conditional density requires the lower-dimensional copula density $c_{n-1}(u_1, \ldots, u_{n-1})$, excluding the variable u_n (corresponding to the variable x_n of interest). So, computing the conditional density (12) from our full n-dimensional copula model proceeds as follows: let the variable x_i range within $[\underline{x}_i, \overline{x}_i]$, then the $(n-1)$-dimensional marginal density is

$$\begin{aligned} f(x_1, \ldots, x_{n-1}) &= \int_{\underline{x}_n}^{\overline{x}_n} f(x_1, \ldots, x_n) dx_n \\ &= \int_{\underline{x}_n}^{\overline{x}_n} \prod_{j=1}^{n} f_j(x_j) c_n(F_1(x_1), \ldots, F_n(x_n)) dx_n \\ &= [\Delta(\overline{x}_n) - \Delta(\underline{x}_n)] \cdot \prod_{j=1}^{n-1} f_j(x_j) \end{aligned}$$

with

$$\Delta(x) := \frac{\partial^{n-1}}{\partial x_1 \cdots \partial x_{n-1}} C_n(F_1(x_1), \ldots, F_{n-1}(x_{n-1}), F_n(x))$$

From this, the sought conditional distribution is immediately found as

$$f(x_n|x_1, \ldots, x_{n-1}) = f_n(x_n) \frac{c_n(F_1(x_1), \ldots, F_n(x_n))}{\Delta(\overline{x}_n) - \Delta(\underline{x}_n)} \qquad (13)$$

Note that the density f_n of the variable of interest can be estimated both parametrically or non-parametrically (e.g., via kernel estimators), while in practice the distribution functions are estimated empirically to avoid additional assumptions.

In a general setting, we first compute the copula density (if the copula at hand is differentiable), the tedious technicalities of which may conveniently be handled by a computer algebra system like MATHEMATICA or MAPLE. Again, this procedure simplifies within a smaller family of copulas.

For a n-dimensional Archimedean copula, the density turns out to be

$$c(u_1, \ldots, u_n) = (\phi^{-1})^{(n)}(\phi(u_1) + \ldots + \phi(u_n)) \prod_{i=1}^{n} \phi'(u_i)$$

where $(\phi^{-1})^{(n)}(t)$ denotes the n-th derivative of the inverse function $\phi^{-1}(t)$. This can be computed for Gumbel, Frank and Ali-Mikhael-Haq copulas, as for example done in [22], but becomes infeasible for the Gaussian copula considered at the beginning.

In the case of a nested copula, there is no simple closed expression available. One has to compute the derivative of the top level copula that describes the behaviour of all variables together, which invokes the chain rule. While this may get complex in the general case, it is still practicable in our case.

In models that involve more levels of sub-copulas than the one considered here, one might use the derivative of $C_{L,1}(C_{L-1,1}, \ldots, C_{L-1,n_{L-1}})$ that evaluates to

$$\begin{aligned} \frac{\partial^d C_{L,1}}{\partial u_1 \cdots \partial u_d} = \sum_{i=0}^{d-n_{L-1}} \sum_{k_1, \ldots, k_{n_{L-1}}} \Bigg\{ & \frac{\partial^{d-i} C_{L,1}}{\partial C_{L-1,1}^{k_1} \cdots \partial C_{L-1,n_{L-1}}^{k_{n_{L-1}}}} \\ & \times \prod_{r=1}^{n_{L-1}} \sum_{v_1, \ldots, v_{k_r}} \frac{\partial^{|v_1|} C_{L-1,r}}{\partial v_1} \cdots \frac{\partial^{|v_{k_r}|} C_{L-1,r}}{\partial v_{k_r}} \Bigg\} \end{aligned}$$

where the outer sum is taken over all integers $k_1, \ldots, k_{n_{L-1}}$ that sum up to $d - i$ and satisfy $k_j \leq d_{L-1,j}$ while the inner sum is over partitions $v_1, \ldots, v_{k_r}$ of those u_i showing up in the r-th copula at level $L-1$. For more details about this specific case, see [19].

B. Self-Adaptation to Environmental Conditions

For a general description, we relabel the variables and let X_n be the device or performance parameter that we wish to predict based on the known environmental conditions $x_1, \ldots, x_{n-1}$. Section IV-C illustrates this for $X_n = \text{QBER}$ and $(X_1, X_2, X_3, X_4) = (\text{DUR}, \text{RAD}, \text{TEMP}, \text{HUM})$.

A prediction of X_n, e.g., the QBER rate given the current environmental conditions, is then given by the conditional expectation or, alternatively, by any value x_n that maximizes

expression (13) for $f(x_n|x_1,\ldots,x_{n-1})$ for the given values $x_1,\ldots,x_{n-1}$. This maximization can be done using standard numerical techniques, whose details are outside our scope here.

Since the indication of an adversary's presence hinges on known performance characteristics, most importantly the QBER rate, it is easy to adapt the respective thresholds to the expected values under the current environmental conditions. Adapting to different conditions then amounts to doing the optimization again under the new configuration.

C. A Worked Example

The density $c(u_1,\ldots,u_5)$ of the top level copula $C_{L,1}$ can be calculated by applying the chin rule. To avoid errors in potentially messy calculations like the following, a computer algebra system may come in handy.

The copula C describing our network was found to be

$$\exp\left\{-\left[\left((-\ln u_1)^{\theta_2}+(-\ln u_2)^{\theta_2}\right)^{\frac{\theta_1}{\theta_2}}+\left[\left((-\ln u_3)^{\theta_4}+(-\ln u_4)^{\theta_4}\right)^{\frac{\theta_3}{\theta_4}}+(-\ln u_5)^{\theta_3}\right]^{\frac{\theta_1}{\theta_3}}\right]^{1/\theta_1}\right\} \tag{14}$$

Generally, it holds

$$\frac{\partial^5 C_{3,1}}{\partial u_1\cdots\partial u_5}=\frac{\partial^5 C_{3,1}}{\partial^2 C_{2,1}\partial^3 C_{2,2}}\cdot\frac{\partial^2 C_{2,1}}{\partial u_1\partial u_2}\cdot\frac{\partial^3 C_{2,2}}{\partial^2 C_{1,1}\partial u_5}\cdot\frac{\partial^2 C_{1,1}}{\partial u_3\partial u_4},$$

where the two most inner derivatives compute as

$$\begin{aligned}\frac{\partial^2 C}{\partial u_1\partial u_2}&=\frac{1}{u_1\cdot u_2}(\log(u_1)\cdot\log(u_2))^{\theta-1}\\&\cdot\exp\left[-\left((-\log(u_1))^{\theta}+(-\log(u_2))^{\theta}\right)^{\frac{1}{\theta}}\right]\\&\cdot\left((-\log(u_1))^{\theta}+(-\log(u_2))^{\theta}\right)^{\frac{1}{\theta}-2}\\&\cdot\left(\left((-\log(u_1))^{\theta}+(-\log(u_2))^{\theta}\right)^{\frac{1}{\theta}}+\theta-1\right)\end{aligned} \tag{15}$$

for any two-dimensional Gumbel copula C. Alternatively to this straightforward calculation, the two-dimensional density (15) can be computed directly from the generator function using the chain rule

$$\begin{aligned}c(u_1,u_2)&=\frac{\partial^2}{\partial u_1\partial u_2}\phi^{-1}(\phi(u_1)+\phi(u_2))\\&=-\frac{\phi''(C(u_1,u_2))\phi'(u_1)\phi'(u_2)}{[\phi'(C(u_1,u_2))]^3}\end{aligned} \tag{16}$$

if both derivatives exist (see also [17]).

To find the expression for $\Delta(x)$ we analogously compute

$$\frac{\partial^4 C_{3,1}}{\partial^1 C_{2,1}\partial^3 C_{2,2}}\cdot\frac{\partial^1 C_{2,1}}{\partial u_2}\cdot\frac{\partial^3 C_{2,2}}{\partial^2 C_{1,1}\partial u_5}\cdot\frac{\partial^2 C_{1,1}}{\partial u_3\partial u_4} \tag{17}$$

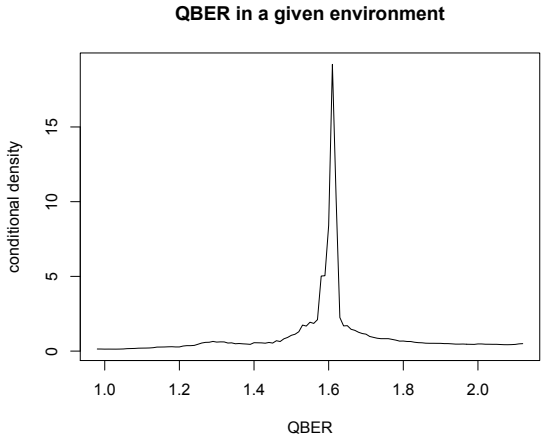


Figure 4. Density of QBER in a known environment

with the third order derivative of a Gumbel copula

$$\begin{aligned}\frac{\partial^3 C}{\partial u_1\partial u_2\partial u_3}&=\frac{(-\log(u_1)\cdot\log(u_2)\cdot\log(u_3))^{\theta-1}}{u_1\cdot u_2\cdot u_3}\cdot\exp\left[-z^{\frac{1}{\theta}}\right]\\&\cdot\left(z^{3/\theta-3}+3(\theta-1)\cdot z^{2/\theta-3}+(\theta-1)(2\theta-1)z^{1/\theta-3}\right)\end{aligned} \tag{18}$$

where $z=(-\log(u_1))^{\theta}+(-\log(u_2))^{\theta}+(-\log(u_3))^{\theta}$. Again, this density can be computed from the generator function directly if all necessary derivatives exist, yielding

$$\begin{aligned}&\frac{\partial^3}{\partial u_1\partial u_2\partial u_3}\phi^{-1}\left(\phi(u_1)+\phi(u_2)+\phi(u_3)\right)\\&=\phi'(u_1)\phi'(u_2)\phi'(u_3)\frac{3[\phi''(C)]^2-\phi'''(C)\cdot\phi'(C)}{[\phi'(C)]^5}\end{aligned} \tag{19}$$

with the abbreviation $\phi(C)=\phi(C(u_1,u_2,u_3))$.

For the quantum network considered here, the conditional density of the QBER displayed in Figure 4 displays a unique maximum of the conditional density around QBER $=1.61\%$, given typical environmental conditions that represent the current situation: sunshine duration DUR $=0$s, global radiation RAD $=0\mathrm{W/m^2}$, relative humidity HUM $=88\%$, and air temperature TEMP $=14.4°\mathrm{C}$. This means that QBER-values lower than 1.14% or higher then 2.07% are unlikely (i.e., these regions have a probability mass of 5% together) and probably arising from the presence of an eavesdropper. Our analysis has been performed for typical values of the environmental variables, i.e., we set the variable DUR to zero as the sun did typically not shine during the measurement process.

Variation of these values does not fundamentally affect our findings but the actual shape of the conditional density turned out to be quite sensitive to small changes.

For example, if we chose TEMP $=14.5°\mathrm{C}$ and HUM $=90\%$, higher values of QBER are more likely and the conditional density becomes even more narrow than before. Figure 5 displays the effect of this change. Despite these differences

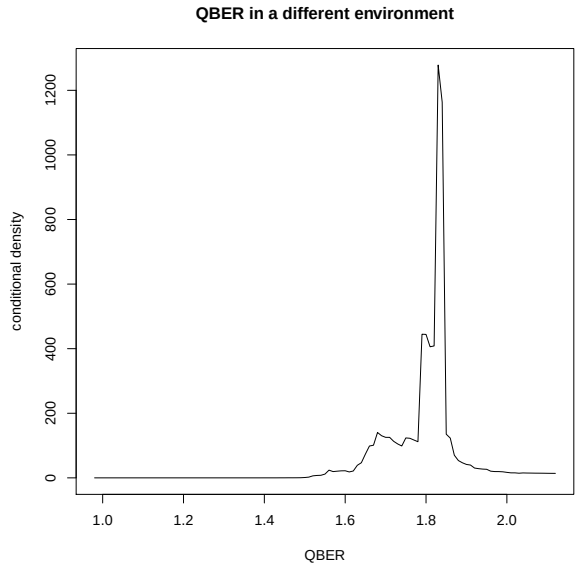


Figure 5. Density of QBER in a slightly different environment

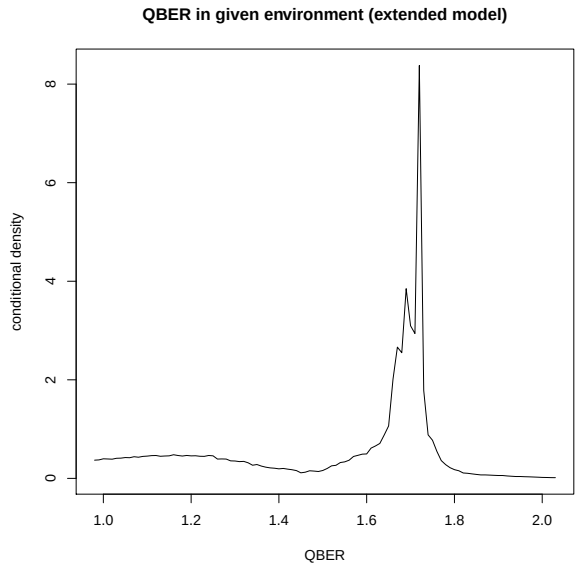


Figure 6. Density of QBER in a given environment based on the extended model

the conditional density still exhibits a single maximum and thus allows again to determine unlikely values.

In appendix A we explain how this estimation procedure can be improved. Figure 6 shows the conditional density based on this modified model. The density exhibits a similar behavior, i.e. there is a narrow peak corresponding to the most plausible values of the QBER in the given environment.

A more detailed documentation of our experiments is found in appendix A, where we give a step-by-step description of the calculations, augmented by R-code to help the reader in applying our method in other scenarios.

V. Conclusion

Now, we come back to the initial problem that motivated this entire study. Recall that in a QKD setting, an unnaturally high qubit error rate indicates the presence of an adversary. Conversely, we need an idea about the "natural" rate of qubit errors. Given the conditional density (12) and according to the previous remarks, we can thus obtain a threshold for the qubit error rate that is tailored to the implementation, environment and device, and which can be adapted to changing environmental conditions. The steps are the following, and graphically summarized in Figure 7:

1) We run the device in a setting where there is no eavesdropper on the line to draw a series of measurements under clean conditions. In particular, we elicit all environmental variables of interest, especially the qubit error rate.
2) We fit a copula model to the so-obtained data $\mathbf{D}$, possibly doing a pre-conditioning (as described in Section III-D) for a statistically and numerically good fit. The fitting can be done using standard statistical software like R, using copula-specific libraries like HAC [18]. The derivation of the conditional distribution is easy by virtue of computer algebra systems like MATHEMATICA.
3) Having the copula-model, we obtain the conditional distribution (13) of the QBER under all environmental influences. Its maximization gives the currently valid threshold under the present environmental conditions. Speaking differently, this process tells us which values of the QBER are *not* likely enough to occur for a given value of the keyrate.

The respective details of each step have been described in previous sections, giving examples along the way to illustrate the particular tasks. Nevertheless, the above process remains of generic nature and calls for appropriate instantiation (e.g., different environmental influences such as noisy source and detectors or turbulence structure of the air could be considered).

Once the probability density of the QBER conditional on current working conditions is obtained, it is a simple matter to equip a QKD device with sensory to keep the expected natural QBER rate continuously updated. We stress that this updating is unaffected by the presence of an attacker, unless the intruder manages to steer the environmental conditions in a way s/he likes. Assuming the absence of such an ability, the copula dependency model and its implied predictive distributions are an effective mean to let the devices re-calibrate themselves under the changing working conditions. Next steps in this research direction comprise practical experiments under variable lab conditions to test the quality of QBER adaption in terms of a performance gain over statically configured devices. As an important side-effect, this would also reveal possibilities

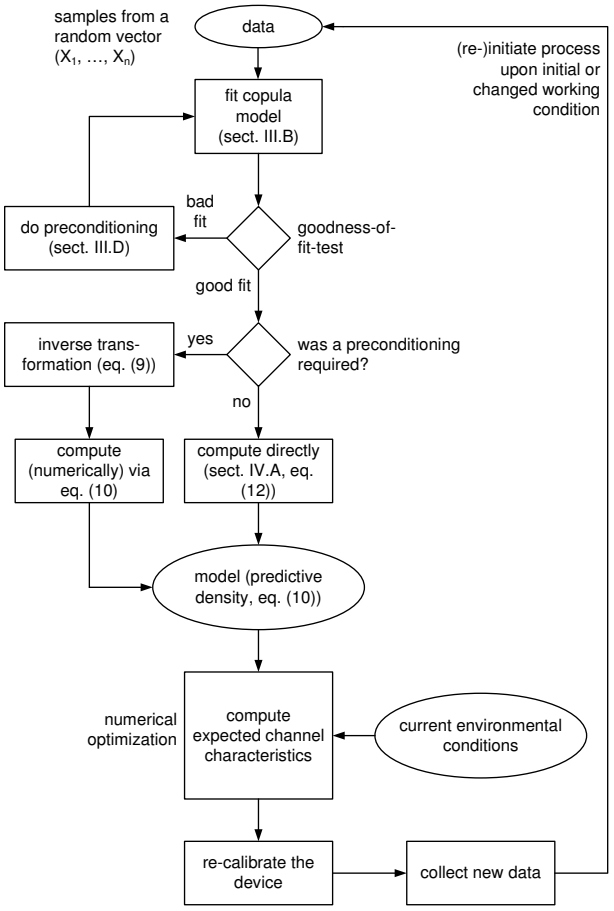


Figure 7. Building up and using the stochastic models for device calibration

to attack a QKD line by changing environmental factors. Such an attack has seemingly not been considered in the literature so far.

REFERENCES

[1] S. König and S. Rass, "Self-adaption of quantum key distribution devices to changing working conditions," in Proc. of the International Conference on Quanum-, Nano- and Microtechnology (ICQNM). IARIA XPS Press, 2014, pp. 1–7.

[2] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," Nature, vol. 299, no. 802, 1982, pp. 802–803.

[3] Peev et al., "The SECOQC quantum key distribution network in Vienna," New Journal of Physics, vol. 11, no. 7, 2009, p. 075001.

[4] K. Lessiak, C. Kollmitzer, S. Schauer, J. Pilz, and S. Rass, "Statistical analysis of QKD networks in real-life environments," in Proceedings of the Third International Conference on Quantum, Nano and Micro Technologies. IEEE Computer Society, February 2009, pp. 109–114.

[5] K. Lessiak, "Application of generalized linear (mixed) models and nonparametric regression models for the analysis of QKD networks," Master's thesis, Universität Klagenfurt, 2010.

[6] T. Schmitt-Manderbach, "Long distance free-space quantum key distribution," Ph.D. dissertation, Ludwig–Maximilians–University Munich, Faculty of Physics, 2007.

[7] H. Xu, L. Ma, A. Mink, B. Hershman, and X. Tang, "1310-nm quantum key distribution system with up-conversion pump wavelength at 1550 nm," Optics Express, vol. 15, Jun. 2007, pp. 7247–7260.

[8] M. Li et al., "Measurement-device-independent quantum key distribution with modified coherent state," Opt. Lett., vol. 39, no. 4, Feb 2014, pp. 880–883.

[9] P. Jouguet, S. Kunz-Jacques, A. Leverrier, P. Grangier, and E. Diamanti, "Experimental demonstration of long-distance continuous-variable quantum key distribution," Nature Photonics, no. 5, 2013, pp. 378–381. [Online]. Available: http://www.nature.com/nphoton/journal/v7/n5/full/nphoton.2013.63.html [retrieved: September, 2014]

[10] A. Acín, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani, "Device-independent security of quantum cryptography against collective attacks," Physical Review Letters, vol. PRL 98, 230501, no. 1–4, 2007.

[11] Y. Liu et al., "Experimental measurement-device-independent quantum key distribution," Phys. Rev. Lett., vol. 111, no. 13, 2013, p. 130502. [Online]. Available: http://www.biomedsearch.com/nih/Experimental-Measurement-Device-Independent-Quantum/24116758.html [retrieved: September 2014]

[12] C. Ruican, M. Udrescu, L. Prodan, and M. Vladutiu, "Adaptive vs. self-adaptive parameters for evolving quantum circuits," in Evolvable Systems: From Biology to Hardware, ser. Lecture Notes in Computer Science, G. Tempesti, A. Tyrrell, and J. Miller, Eds. Springer Berlin Heidelberg, 2010, vol. 6274, pp. 348–359.

[13] C.-J. Lin, C.-H. Chen, and C.-Y. Lee, "A self-adaptive quantum radial basis function network for classification applications," in Proc. of International Joint Conference on Neural Networks, Vol. 4. IEEE, July 2004, pp. 3263–3268.

[14] A. M. Al-Adilee and O. Nánásiová, "Copula and s-map on a quantum logic." Inf. Sci., vol. 179, no. 24, 2009, pp. 4199–4207.

[15] P. Embrechts, F. Lindskog, and A. McNeil, Modelling Dependence with Copulas and Applications to Risk Management, Handbook of Heavy Tailed Distributions in Finance, Elsevier, 2001.

[16] D. Kelly, "Using copulas to model dependence in simulation risk assessment," in Proc. of 2007 ASME International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers, 2007, pp. 81–89.

[17] R. Nelsen, An Introuction to Copulas. Springer, 2006.

[18] O. Okhrin and A. Ristig, "Hierarchical archimedean copulae: The HAC package," Journal of Statistical Software, vol. 58, no. 4, 2014, pp. 1–20. [Online]. Available: http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2012-036.pdf [retrieved: September, 2014]

[19] C. Savu and M. Trede, "Hierarchies of Archimedean copulas," Quantitative Finance, vol. 10, no. 3, February 2010, pp. 295–304.

[20] C. Genest and B. Rémillard, "Validity of the parametric bootstrap for goodness-of-fit testing in semiparametric models," Annales de l'institut Henri Poincaré (B) Probabilités et Statistiques, vol. 44, no. 6, 2008, pp. 1096–1127. [Online]. Available: http://eudml.org/doc/78005 [retrieved: September, 2014]

[21] C. P. Robert, The Bayesian choice. New York: Springer, 2001.

[22] C. Savu and M. Trede, "Goodness-of-fit tests parametric families of Archimedean copulas," Quantitative Finance, vol. 8, no. 2, March 2008, pp. 109–116.

[23] J.-D. Fermanian, D. Radulovic, and M. Wegkamp, "Weak convergence of empirical copula processes," Bernoulli, vol. 10, no. 5, 10 2004, pp. 847–860. [Online]. Available: http://dx.doi.org/10.3150/bj/1099579158

APPENDIX

To ease reproducing our computations in practical applications, we attach our R-implementation of the procedures sketched in the previous paragraphs here. Inline, we comment

on the code where necessary to extend the description in the body of the paper.

The libraries that we used were `copula`, `HAC` and `MASS`. The original data has been loaded into a data frame `X`.

The following code decorrelates the data and leaves a data frame `Y` whose covariance structure is the identity matrix:

```
U <- chol(cov(X)) # Cholesky decomposition
Uinv <- solve(U)  # inversion of U
X <- as.matrix(X) # coerce X into a matrix
Y <- X%*%Uinv     # do the decorrelation
```

This data frame is then (positively) recorrelated by the matrix `A` as described in Section III-D.

```
A <- matrix(c(...)) # matrix values
Z <- Y%*%A          # re-correlation
```

In the paper this whole process is described by equation (9).

Given the positively recorrelated data, the fitting method from the `HAC` package applies, giving us a copula model and the θ-values (cf. Figure 3). We used full maximum likelihood estimation (ML) here.

```
UZ <- pobs(Z)    # pseudo observations
estim.full <- estimate.copula(
                  UZ,
                  method = 2, # = full ML
                  hac = estim,
                  margins = NULL,
                  epsilon = 0.4)
theta.full <- get.params(estim.full)
```

At this stage, we ought to check the goodness of fit for the copula model. Here, we enter the bootstrapping stage as sketched in Section III-C. An empirical d-dimensional copula based on n data records in a matrix $\mathbf{V} \in \mathbb{R}^{n \times d}$ is defined by $C_{\mathbf{V}}(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(V_1^i \leq u_1, \ldots, V_d^i \leq u_d)$, where $\mathbb{I}$ is an indicator function. (The estimate $\hat{C}(\mathbf{u}) = \mathbf{C}_{\mathbf{V}}(\mathbf{u})$ is known to converge uniformly to the underlying true copula, at least in the case of independent marginal distributions [23].)

```
empCop <-function(V, u){
     1/n * length(which(V[,1] <= u[1] &
                        V[,2] <= u[2] &
                        V[,3] <= u[3] &
                        V[,4] <= u[4] &
                        V[,5] <= u[5]))
}
```

Next comes the bootstrapping procedure, which takes `N` iterations ($N = 1000$ in our experiments). A single test for the goodness of fit can be implemented as follows:

```
estimatedCopula <- estimate.copula(UZ,
        method = 1, margins = NULL,
        epsilon = 0.4)
```

This estimate can be improved in the following way:

```
# quasi ML estimation as before (method = 1)
qMLCopulaEst <- estimate.copula(UZ,
        method = 1, margins = NULL,
        epsilon = 0.4)
# update (method = 1 -> full ML)
estimatedCopula <- estimate.copula(UZ,
        method = 2,
        hac = qMLCopulaEst,
        margins = NULL, epsilon=0.4)
```

Notice however that this increases the runtime significantly.

For the bootstrap (as prescribed by [20]), we need to cast the observations into uniformly distributed values by applying the empirical copula function based on the pseudo-observations `UZ` from above. This gives the data matrix `C1`. The estimated copula should, by construction, resemble this data quite well, and thus perform equally good as the empirical copula function in casting the observations into uniformly distributed values. Hence, we should almost obtain the same results by applying the fitted copula (distribution function `pHAC`) to `UZ`, giving the observation data `C2`. The difference between the two tells the numeric magnitude of a "small deviation" between the data and the model (cf. Section III-C).

```
C1 <- apply(UZ, 1,
        function(x)(empCop(UZ,x)))
C2 <- pHAC(UZ,
        estimatedCopula,
        margins=NULL)
Sn <- sum((C1 - C2)^2)  # bootstrap value
```

The actual bootstrap is done by drawing random values from the copula model (function `rHAC`), turning it into pseudo-observations and estimating the copula in the same way as before, but based on the random observations now. Over N repetitions (we took $N = 1000$), the k-th such fit is "accepted", if its deviation `Snk` is less than `Sn`, as computed above, i.e., the p-value of the test is defined as [20] $p = \frac{1}{N} \sum_{k=1}^{n} \mathbb{I}(S_{nk} > S_n)$, with S_n being `Sn` from above. To save space in the listing below, the ellipsis (...) in the parameter list is to be replaced by the same parameters in the identical calls as in previous listings.

```
pValueEst <- 0
for(k in 1:N) {
      Xk <- rHAC(n, estimatedCopula)
      Uk<-pobs(Xk)
      bootstrapQML <- estimate.copula(Uk,
            method = 1, ...)
      bootstrapEst <- estimate.copula(Uk,
            method = 2,
            hac = bootstrapQML, ...)
      C1 <- apply(Uk, 1,
            function(x)(empCop(Uk,x)))
      C2 <- pHAC(Uk, bootstrapEst, ...)
      Snk <- sum((C1 - C2)^2)

        if (Snk > Sn) {
```

```
            pValueEst <- pValueEst + 1
        }
}
pValueEst <- pValueEst / N
```

Our experiments revealed that a single trial usually yields not a good fit, so the above iteration can be repeated until a sufficiently large p-value is obtained (in our setting, we took 200 rounds to come up with a few good fits).

Given that the fit has a p-value > 0.05, we accept it and step towards estimating the predictive density; equation (13): First, we need the unconditional density of QBER, which in our case is the first variable in the (still re-correlated) data frame `Z`. We fitted a gamma-distribution by maximum likelihood:

```
f <- fitdistr(Z[,1], "gamma")
fn <- function(x) {
dgamma(x, shape = f$estimate[1],
          rate = f$estimate[2])
}
```

The conditional density is then directly computed from formula (13), by first transforming the input data into uniformly distributed values (by applying the empirical marginal distribution functions obtained from a call to `ecdf`) and implementing the expression for Δ as a function `delta` (omitted here for space reasons):

```
# get the empirical distribution functions
F1<-ecdf(Z[,1]) # QBER
F2<-ecdf(Z[,3]); # HUM
F3<-ecdf(Z[,2]); # TEMP
F4<-ecdf(Z[,5]); # RAD
F5<-ecdf(Z[,4]); # DUR
# range of QBER
qbermin<-min(X[,1])
qbermax<-max(X[,1])

# conditional density function
conddens<-function(DUR,RAD,TEMP,HUM,QBER){
# transform data into uniformly distr.
u1<-F1(QBER); u2<-F2(HUM);
    u3<-F3(TEMP); u4<-F4(RAD);
    u5<-F5(DUR)
# conditional density formula (13)
fn(QBER) * cn(u1,u2,u3,u4,u5) /
         (delta(F1(qbermaxz),u2,u3,u4,u5) -
          delta(F1(qberminz),u2,u3,u4,u5))
}
```

The `conddens` function is now ready to be used for configuring the device, for example, by determining its maximum w.r.t. QBER (maximum likelihood estimation), given the current environmental conditions DUR, RAD, TEMP and HUM. We stress, however, that care has to be taken since all this construction works on the transformed data `Z` rather than the actual (physical) measurements `X`. In order to properly apply the function, we therefore must transform the current environmental data in much the same way as the data has been transformed to find a suitable model. That is, we apply the transformation matrix $\mathbf{M}$ to the physical input data and use the results as the arguments in the `conddens` function: calling `xdat` the real environmental conditions (values as given in Section IV-C), then the transformed `zdat` is the input to `conddens` as described above.

```
Zdat<-matrix(rep(0,l*5),nrow=l)
# relabel the variables to fit notation
# of the derivatives cn and delta
colnames(Zdat) <- c("QBER", "HUM",
                    "TEMP", "RAD", "DUR")
# transform QBER-values in given environment
for (i in 1:l){
xdat<-c(x[i],148,90,0,0)
zdat<-t(xdat)%*%Uinv%*%A
zdat<-t(zdat)
DUR<-zdat[4]; RAD<-zdat[5];
HUM<-zdat[3]; TEMP<-zdat[2]; QBER<-zdat[1]
Zdat[i,]<-c(QBER,HUM,TEMP,RAD,DUR)
}
# determine range of transformed data
# (input to function delta)
minz<-min(Zdat[,1])
maxz<-max(Zdat[,1])
```

The density is then visualized by plotting QBER-values x against the corresponding output of the `conddens` function y for each of those values.

```
# range of QBER
qbermin<-min(X[,1]) # 0.98
qbermax<-max(X[,1]) # 2.12
x<-seq(qbermin,qbermax,0.01)
l<-length(x)
# corresponding values of density
y<-rep(0,l)
for (i in 1:l){
  y[i]<-conddens(Zdat[i,1],Zdat[i,2],
            Zdat[i,3],Zdat[i,4],Zdat[i,5])
}
plot(x, y,
   type='l',
   main="QBER in a given environment",
   xlab='QBER',ylab='conditional density')
```

Robustness of Optimal Basis Transformations to Secure Entanglement Swapping Based QKD Protocols

Stefan Schauer and Martin Suda

Digital Safety and Security Department
AIT Austrian Institute of Technology GmbH
Vienna, Austria
Email: stefan.schauer@ait.ac.at, martin.suda.fl@ait.ac.at

***Abstract*—In this article, we discuss the optimality of basis transformations as a security measure for quantum key distribution protocols based on entanglement swapping as well as the robustness of these basis transformations considering an imperfect physical apparatus. To estimate the security, we focus on the information an adversary obtains on the raw key bits from a generic version of a collective attack strategy. In the scenario described in this article, the application of general basis transformations serving as a counter measure by one or both legitimate parties is analyzed. In this context, we show that the angles, which describe these basis transformations, can be optimized compared to the application of a Hadamard operation, which is the standard basis transformation recurrently found in literature. Nevertheless, these optimal angles for the basis transformations have to be precisely configured in the laboratory to achieve the minimum amount for the adversary's information. Since we can not be sure that the physical apparatus is perfect, we will look at the robustness of the optimal choice for the angles. As a main result, we show that the adversary's information can be reduced to an amount of $I_{AE} \simeq 0.20752$ when using a single basis transformation and to an amount of $I_{AE} \simeq 0.0548$ when combining two different basis transformations. This is less than half the information compared to other protocols using a Hadamard operation and thus represents an advantage regarding the security of entanglement swapping based protocols. Further, we will show that the optimal angles to achieve these results are very robust such that an imperfect configuration does only have an insignificant effect on the security of the protocol.**

Keywords–quantum key distribution; optimal basis transformations; imperfect apparatus; Gaussian distribution of angles; security analysis; entanglement swapping

I. Introduction

In a recent article [1], the authors have shown that in a quantum key distribution (QKD) protocol based on entanglement swapping the Hadamard operation is not the optimal choice to secure the protocol against an adversary. Moreover, a combination of basis transformations will reduce the amount of the adversary's information drastically when using general basis transformations. Additionally, we want to show in this article that these general basis transformations are also robust against an imperfect configuration of the physical apparatus.

QKD is one of the major applications of quantum mechanics and, in the last three decades, QKD protocols have been studied at length in theory and in practical implementations [2]–[9]. Most of these protocols focus on prepare and measure schemes, where single qubits are in transit between the communication parties Alice and Bob. The security of these protocols has been discussed in depth and security proofs have been given, for example, in [10]–[12]. In addition to these prepare and measure protocols, several protocols based on the phenomenon of entanglement swapping have been introduced [13]–[18], where entanglement swapping is used to obtain correlated measurement results between the legitimate communication parties, Alice and Bob.

Entanglement swapping has been introduced by Bennett et al. [19], Zukowski et al. [20] as well as Yurke and Stolen [21], respectively. It provides the unique possibility to generate entanglement from particles that never interacted in the past. In detail, Alice and Bob share two Bell states of the form $|\Phi^+\rangle_{12}$ and $|\Phi^+\rangle_{34}$ (cf. picture (1) in Figure 1) in such a way that Alice sends qubit 2 to Bob and Bob sends qubit 3 to Alice. Hence, afterwards Alice is in possession of qubits 1 and 3 and Bob of qubits 2 and 4 (cf. picture (2) in Figure 1). The state of the overall system can thus be described as

$$|\Phi^+\rangle_{12} \otimes |\Phi^+\rangle_{34} = \frac{1}{2}\Big(|\Phi^+\rangle|\Phi^+\rangle + |\Phi^-\rangle|\Phi^-\rangle + |\Psi^+\rangle|\Psi^+\rangle + |\Psi^-\rangle|\Psi^-\rangle\Big)_{1324} \quad (1)$$

Next, Alice performs a complete Bell state measurement on the two qubits in her possession. After this measurement, the qubits 2 and 4 at Bob's side collapse into a Bell state although both qubits originated at completely different sources (cf. picture (4) in Figure 1). Moreover, the state of Bob's qubits fully depends on Alice's measurement result. As presented in (1), Bob always obtains the same result as Alice when performing a Bell state measurement on his qubits. In the aforementioned QKD protocols based on entanglement swapping, Alice and Bob use these correlated measurement results to establish a secret key among them.

A basic technique to secure a QKD protocol is to use a basis transformation, usually a Hadamard operation, to make it easier to detect an adversary. This is implemented, for example, in the prepare and measure schemes described in [2] and [4] but also in QKD schemes based on entanglement swapping (e.g., [14] [17] [22]). Nevertheless, this security measure has just been discussed on the surface so far when it comes to QKD protocols based on entanglement swapping. It has only been shown that these protocols are secure against intercept-resend attacks and basic collective attacks (cf. for example, [13] [14] [17]).

In this article, we will analyze the security of QKD protocols based on entanglement swapping against the *simulation attack*, a general version of a collective attack [23]. As a security measure we will analyze the application of a general

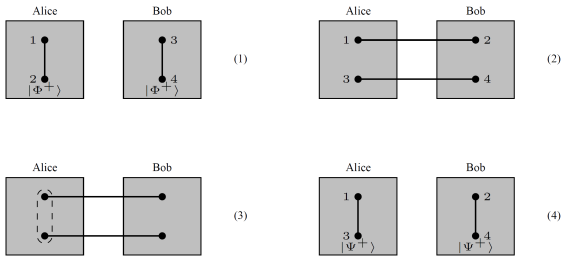


Figure 1. Illustration of entanglement swapping where Alice and Bob share two Bell states each of the form $|\Phi^+\rangle$. The dashed line indicates a measurement in the Bell basis.

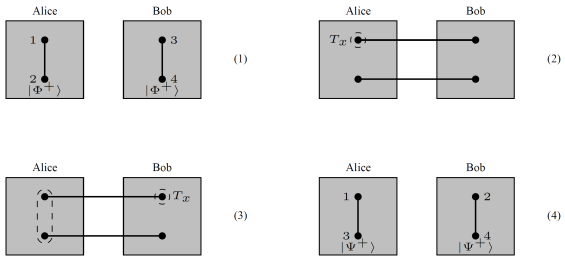


Figure 2. Sketch of a standard setup for an entanglement swapping based QKD protocol. Qubits 2 and 3 are exchanged (cf. picture (2)) and a basis transformation T_x is applied on qubit 1 and inverted by using T_x on qubit 2.

basis transformation T_x, defined by the angles θ and ϕ (cf. (4) and picture (2) in Figure 2). In the course of that, we are going to identify, which values for θ and ϕ are optimal such that an adversary has only a minimum amount of information on the secret raw key. Furthermore, we will look at the robustness of these optimal values for θ and ϕ, i.e., how much the expected error probability and the adversary's information change if Alice and Bob are not able to precisely adjust their apparatus to the optimal values for θ and ϕ.

In the following section, the simulation attack is described in detail and it is explained how an adversary is able to perfectly eavesdrop on a protocol where no basis transformations are applied. In Section III, we look in detail at the general definition of basis transformations and their effect onto Bell states and entanglement swapping. Using these definitions, we discuss in the following sections the effects on the security of entanglement swapping based QKD protocols. Therefore, we look at the application of a general basis transformation by one communication party in Section IV and at the application of two different basis transformations by each of the communication parties in Section V. In Section VI, we will analyze how these results change if the physical apparatus is not configured precisely and the choice of angles can be described by a Gaussian distribution. In the end, we sum up the implications of the results on the security of entanglement based QKD protocols.

II. The Simulation Attack Strategy

In entanglement swapping based QKD protocols like [13]–[15], [17], [18] Alice and Bob rest their security check onto the correlations between their respective measurement results coming from the entanglement swapping (cf. (1)). If these correlations are violated, Alice and Bob have to assume that an adversary is present. In other words, an adversary stays undetected if these correlations are not violated. Hence, a general version of a collective attack has the following basic idea: the adversary Eve tries to find a multi-qubit state, which preserves the correlation between the two legitimate parties. Further, she introduces additional qubits to distinguish between Alice's and Bob's respective measurement results. If she is able to find such a state, Eve stays undetected during her intervention and is able to obtain a certain amount of information about the key (cf. also Figure 3).

In a previous article [23], we already described such a collective attack called *simulation attack* for a specific protocol [18]. The attack implements the strategy described in the previous paragraph, i.e., the correlations are preserved (or "simulated") such that the Eve stays undetected. The generalization from the version presented in [23] is straight forward as described in the following paragraphs.

It has been pointed out in detail in [23] that Eve uses four qubits in a state similar to (1) to simulate the correlations between Alice and Bob. Further, she introduces additional systems $|\varphi_i\rangle$ to distinguish between Alice's different measurement results. This leads to the state

$$|\delta\rangle = \frac{1}{2}\Big(|\Phi^+\rangle|\Phi^+\rangle|\varphi_1\rangle + |\Phi^-\rangle|\Phi^-\rangle|\varphi_2\rangle + |\Psi^+\rangle|\Psi^+\rangle|\varphi_3\rangle + |\Psi^-\rangle|\Psi^-\rangle|\varphi_4\rangle\Big)_{PRQSTU} \tag{2}$$

which is a more general version than described in [23]. From (2) it is easy to see that after a Bell measurement on qubits P and R the state of qubits Q and S collapses into a correlated state. Hence, the state $|\delta\rangle$ preserves the correlation of Alice's and Bob's measurement results coming from the entanglement swapping (cf. (1)). To be able to eavesdrop Alice's and Bob's measurement results, Eve has to choose the auxiliary systems $|\varphi_i\rangle$ such that they are pairwise orthogonal, i.e.,

$$\langle\varphi_i|\varphi_j\rangle = 0 \qquad i,j \in \{1,...,4\} \;\; i \neq j \tag{3}$$

This allows her to perfectly distinguish between Alice's and Bob's respective measurement results and thus gives her full information about the classical raw key generated out of them.

In detail, Eve distributes qubits P, Q, R and S between Alice and Bob such that Alice is in possession of qubits P and R and Bob is in possession of qubits Q and S (cf. picture (1) and (2) in Figure 2). When Alice performs a Bell state measurement on qubits P and R the state of qubits Q and S collapses into the same Bell state, which Alice obtained from her measurement (compare equations (1) and (2) as well as pictures (3) and (4) in Figure 2). Hence, Eve stays undetected when Alice and Bob compare some of their results in public to check for eavesdroppers. The auxiliary system $|\varphi_i\rangle$ remains at Eve's side and its state is completely determined by Alice's measurement result. Therefore, Eve has full information on Alice's and Bob's measurement results and is able to perfectly eavesdrop the classical raw key.

There are different ways for Eve to distribute the state $|\delta\rangle_{P-U}$ between Alice and Bob. One possibility is that Eve is in possession of Alice's and Bob's source and generates $|\delta\rangle_{P-U}$ instead of the respective Bell states. This is a rather strong assumption because the sources are usually located at Alice's or Bob's laboratory, which should be a secure environment. Nevertheless, Eve's second possibility is to intercept the qubits 2 and 3 flying from Alice to Bob and vice versa and

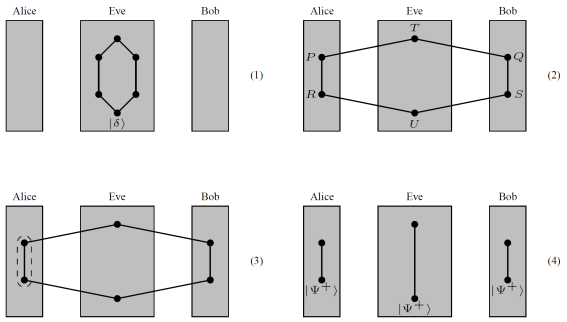


Figure 3. Illustration of the simulation attack for an entanglement swapping based QKD protocol where no basis transformation is applied. It is assumed that Eve directly distributes the state $|\delta\rangle$ between Alice and Bob.

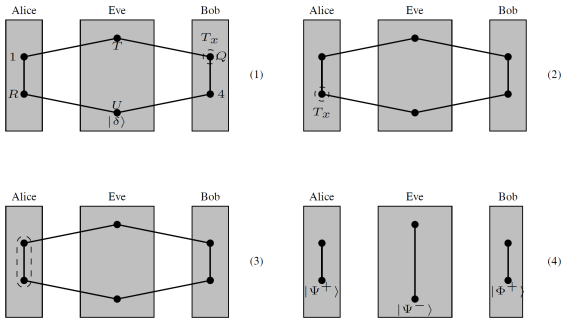


Figure 4. Illustration of the simulation attack for an entanglement swapping based QKD protocol where the basis transformation T_x is applied by Bob. Eve's intervention destroys the correlation between Alice and Bob.

to perform entanglement swapping to distribute the state $|\delta\rangle$. This is a straight forward method as already described in [23].

We want to stress that the state $|\delta\rangle$ is generic for all protocols where 2 qubits are exchanged between Alice and Bob during one round of key generation as, for example, the QKD protocols presented by Song [17], Li et al. [18] or Cabello [13]. As already pointed out in [23], the state $|\delta\rangle$ can also be used for different initial Bell states. For protocols with a higher number of qubits, the state $|\delta\rangle$ has to be extended accordingly.

III. Basis Transformations

In QKD, the most common way to detect the presence of an adversary is to use a random application of a basis transformation by one of the legitimate communication parties. This method can be recurrently found in prepare and measure protocols (e.g., in [2] or [4]) as well as entanglement swapping based protocols (e.g., in [14] [17] or the improved version of the protocol in [18]). The idea for Alice or Bob (or both parties) is to choose at random whether to apply a basis transformation on one of their qubits. This randomly alters the initial state and makes it impossible for an adversary to eavesdrop the transmitted information without introducing a certain error rate, i.e., without being detected. The basis transformation most commonly used in these protocols is the Hadamard operation, which is a transformation from the Z-into the X-basis. In general, a transformation T_x from the Z basis into the X-basis can be described as a rotation about the X-axis by some angle θ, combined with two rotations about the Z-axis by some angle ϕ, i.e.,

$$T_x\big(\theta,\phi\big) = e^{i\phi} R_z\big(\phi\big) R_x\big(\theta\big) R_z\big(\phi\big). \tag{4}$$

The rotations about the X- or Z-axis are described in the most general way by the operators (cf. for example, [24] for further details on rotation operators)

$$\begin{aligned} \mathrm{R}_x\big(\theta\big) &= \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \\ \mathrm{R}_z\big(\theta\big) &= \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \end{aligned} \tag{5}$$

Based on these operators, we directly obtain the matrix representation for $T_x(\theta,\phi)$ as

$$T_x\big(\theta,\phi\big) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\,e^{i\phi}\sin\frac{\theta}{2} \\ -i\,e^{i\phi}\sin\frac{\theta}{2} & e^{2i\phi}\cos\frac{\theta}{2} \end{pmatrix} \tag{6}$$

and the effect of $T_x(\theta,\phi)$ on the computational basis

$$\begin{aligned} T_x\big(\theta,\phi\big)|0\rangle &= \cos\frac{\theta}{2}|0\rangle - i\,e^{i\phi}\sin\frac{\theta}{2}|1\rangle \\ T_x\big(\theta,\phi\big)|1\rangle &= -i\,e^{i\phi}\sin\frac{\theta}{2}|0\rangle + e^{2i\phi}\cos\frac{\theta}{2}|1\rangle. \end{aligned} \tag{7}$$

From these two equations above we immediately see that the Hadamard operation is just the special case where $\theta = \phi = \pi/2$.

In QKD protocols based on entanglement swapping, the basis transformation is usually applied onto one qubit of a Bell state. Taking the general transformation $T_x(\theta,\phi)$ from (4) into account, the Bell state $|\Phi^+\rangle$ changes into

$$\begin{aligned} T_x^{(1)}\big(\theta,\phi\big)|\Phi^+\rangle_{12} = \cos\frac{\theta}{2}\,\frac{1}{\sqrt{2}}\Big(|00\rangle + e^{2i\phi}|11\rangle\Big) \\ -i\,e^{i\phi}\sin\frac{\theta}{2}\,\frac{1}{\sqrt{2}}\Big(|01\rangle + |10\rangle\Big) \end{aligned} \tag{8}$$

and accordingly for the other Bell states. The superscript "(1)" in (8) indicates that the transformation $T_x\big(\theta,\phi\big)$ is applied on qubit 1. As a consequence, the application of $T_x(\theta,\phi)$ before the entanglement swapping is performed changes the results based on the angles θ and ϕ. In detail, after the application of the basis transformation on qubit 1, the overall state of Alice's and Bob's qubits is (cf. picture (2) in Figure 2)

$$\begin{aligned} T_x^{(1)}\big(\theta,\phi\big)|\Phi^+\rangle_{12}|\Phi^+\rangle_{34} = \\ \frac{1}{2}\Big(|\Phi^+\rangle_{13}T_x^{(2)}\big(\theta,\phi\big)|\Phi^+\rangle_{24} \\ +|\Phi^-\rangle_{13}T_x^{(2)}\big(\theta,\phi\big)|\Phi^-\rangle_{24} \\ +|\Psi^+\rangle_{13}T_x^{(2)}\big(\theta,\phi\big)|\Psi^+\rangle_{24} \\ +|\Psi^-\rangle_{13}T_x^{(2)}\big(\theta,\phi\big)|\Psi^-\rangle_{24}\Big) \end{aligned} \tag{9}$$

Next, Alice performs her Bell state measurement on qubits 1 and 3 of this state and obtains one of the four Bell states (cf. picture (3) in Figure 2). The superscripts "(1)" and "(2)" in (9) indicate that after Alice's Bell state measurement on qubits 1 and 3 the transformation $T_x\big(\theta,\phi\big)$ swaps from qubit 1 onto qubit 2. Thus, when Bob performs his Bell state measurement on qubits 2 and 4, he will not obtain a result correlated to Alice's measurement outcome any more. In detail, assuming that Alice obtained $|\Phi^+\rangle_{13}$ from her measurement we can

directly see from (8) that Bob will obtain $|\Phi^+\rangle_{24}$ only with probability (cf. also (9) above)

$$\begin{aligned} P_{corr} &= T_x^{(2)}\big(\theta,\phi\big)\ \langle\Phi^+||\Phi^+\rangle\langle\Phi^+|\ T_x^{(2)}\big(\theta,\phi\big)|\Phi^+\rangle \\ &= \frac{1}{4}\cos^2\frac{\theta}{2}\Big(2+e^{2i\phi}+e^{-2i\phi}\Big) \\ &= \cos^2\frac{\theta}{2}\ \cos^2(\phi). \end{aligned} \tag{10}$$

(and similarly for Alice's other possible results). Otherwise, he obtains an uncorrelated result, which results in a problem because Bob is no longer able to compute Alice's state based on his result and vice versa.

Fortunately, Bob can resolve this problem by transforming the state of qubits 2 and 4 back into its original form before he performs his Bell state measurement. Following (9), where Alice performs $T_x\big(\theta,\phi\big)$ on qubit 1, he achieves that by applying the inverse of the basis transformation, i.e.,

$$T_x^{-1}\big(\theta,\phi\big) = \begin{pmatrix} \cos\frac{\theta}{2} & i\,e^{-i\phi}\sin\frac{\theta}{2} \\ i\,e^{-i\phi}\sin\frac{\theta}{2} & e^{-2i\phi}\cos\frac{\theta}{2} \end{pmatrix} \tag{11}$$

on qubit 2 in his possession. Afterwards, he will obtain a correlated result from his measurement on qubits 2 and 4.

As we will see in the following section, if an adversary interferes with the communication, the effects of Alice's basis transformation can not be represented as in (9) any longer. Thus, even if Bob applies the inverse transformation, Alice's and Bob's results are uncorrelated to a certain amount. This amount is reflected in an error rate detected by Alice and Bob during post processing.

IV. Single Application of General Basis Transformations

Previous works [25] [26] already deal with the scenarios where Alice or Bob or both parties randomly apply a simplified version of basis transformations. Therein, the simplification addresses the angle ϕ, i.e., the rotation about the Z-axis. In the security discussions in [25], the angle ϕ is fixed at $\pi/2$ for reasons of simplicity. That means, the rotation about the Z-axis is constant at an angle of $\pi/2$ such that only the angle θ can be chosen freely.

In this section and the next one, we want to extend the results from [25] [26] by applying general basis transformations, which means Alice and Bob are able to choose both angles θ and ϕ in (4) freely. At first, we are looking only on one party performing a basis transformation on the respective qubits and in the next section on two different basis transformations performed by each of the parties. For each scenario we will show, which values for θ and ϕ are optimal to give an adversary the least information about the raw key bits. In the course of the two scenarios, we will denote Alice's operation as $T_x\big(\theta_A,\phi_A\big)$ and, accordingly, Bob's operation as $T_x\big(\theta_B,\phi_B\big)$.

As already pointed out above, the application of the basis transformation occurs at random and, due to the structure of the state $|\delta\rangle$, Eve is able to obtain full information about Alice's and Bob's secret, if the two parties do not apply any basis transformation at all (cf. [25] [26]). Therefore, we look at first at the effects of a basis transformation at Alice's side. Her initial application of the general basis transformation $T_x\big(\theta_A,\phi_A\big)$ does alter the state $|\delta\rangle_{1QR4TU}$ introduced by Eve such that it is changed to

$$|\delta'\rangle_{1QR4TU} = T_x^{(1)}\big(\theta_A,\phi_A\big)|\delta\rangle_{1QR4TU} \tag{12}$$

After a little algebra, we see that Alice obtains all four Bell states with equal probability and after her measurement the state of the remaining qubits is

$$\begin{aligned} &e^{i\phi_A}\cos\frac{\theta_A}{2}\cos\phi_A\ |\Phi^+\rangle_{Q4}|\varphi_1\rangle_{TU} \\ -&ie^{i\phi_A}\cos\frac{\theta_A}{2}\sin\phi_A\ |\Phi^-\rangle_{Q4}|\varphi_2\rangle_{TU} \\ &-ie^{i\phi_A}\sin\frac{\theta_A}{2}\ |\Psi^+\rangle_{Q4}|\varphi_3\rangle_{TU} \end{aligned} \tag{13}$$

assuming Alice obtained $|\Phi^+\rangle_{1R}$. We are presenting just the state for this particular result in detail because it would be simply too complex to describe the representation of the whole state for all possible outcomes here. Nevertheless, for the other three possible results the remaining qubits end up in a similar state, where only Bob's Bell states of the qubits Q and 4 as well as Eve's auxiliary states of the qubits T and U change accordingly to Alice's measurement result.

Before Bob performs his Bell state measurement, he has to reverse Alice's basis transformation. As already pointed out in the previous section, this can be achieved by applying $T_x^{-1}\big(\theta_A,\phi_A\big)$ on qubit Q in his possession. Whereas this would reverse the effect of Alice's basis transformation if no adversary is present, the structure of Eve's state $|\delta\rangle$ makes this reversion impossible. Hence, the application of $T_x^{-1}\big(\theta_A,\phi_A\big)$ on qubit Q changes the state in (13) into

$$\begin{aligned} &e^{i\phi_A}\cos\frac{\theta_A}{2}\cos\phi_A\Bigg[\cos\frac{\theta_A}{2}\frac{1}{\sqrt{2}}\Big(|00\rangle_{Q4}+e^{-2i\phi_A}|11\rangle_{Q4}\Big) \\ &\qquad +ie^{-i\phi_A}\sin\frac{\theta_A}{2}\frac{1}{\sqrt{2}}\Big(|01\rangle_{Q4}+|10\rangle_{Q4}\Big)\Bigg]|\varphi_1\rangle_{TU} \\ -&ie^{i\phi_A}\cos\frac{\theta_A}{2}\sin\phi_A\Bigg[\cos\frac{\theta_A}{2}\frac{1}{\sqrt{2}}\Big(|00\rangle_{Q4}-e^{-2i\phi_A}|11\rangle_{Q4}\Big) \\ &\qquad +ie^{-i\phi_A}\sin\frac{\theta_A}{2}\frac{1}{\sqrt{2}}\Big(|01\rangle_{Q4}+|10\rangle_{Q4}\Big)\Bigg]|\varphi_2\rangle_{TU} \\ -&ie^{i\phi_A}\sin\frac{\theta_A}{2}\Bigg[\cos\frac{\theta_A}{2}\frac{1}{\sqrt{2}}\Big(|01\rangle_{Q4}+e^{-2i\phi_A}|10\rangle_{Q4}\Big) \\ &\qquad +ie^{-i\phi_A}\sin\frac{\theta_A}{2}\frac{1}{\sqrt{2}}\Big(|00\rangle_{Q4}+|11\rangle_{Q4}\Big)\Bigg]|\varphi_3\rangle_{TU} \end{aligned} \tag{14}$$

Therefore, Bob obtains the correlated state $|\Phi^+\rangle_{Q4}$ only with probability

$$P_{\Phi^+} = \frac{1}{4}\Big(3+\cos\big(4\phi_A\big)\Big)\cos^4\frac{\theta_A}{2}+\sin^4\frac{\theta_A}{2} \tag{15}$$

and the other results with the respective probabilities

$$\begin{aligned} P_{\Phi^-} &= 2\cos^4\frac{\theta_A}{2}\cos^2\phi_A\sin^2\phi_A \\ P_{\Psi^+} &= \frac{1}{2}\sin^2\theta_A\cos^2\phi_A \\ P_{\Psi^-} &= \frac{1}{2}\sin^2\theta_A\sin^2\phi_A. \end{aligned} \tag{16}$$

Hence, due to Eve's intervention Bob obtains a result uncor-

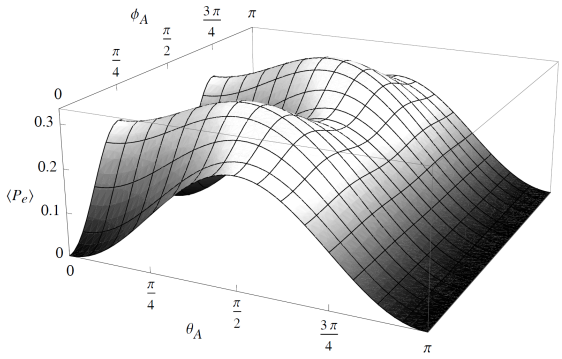


Figure 5. Error probability $\langle P_e \rangle$ depending on θ_A and ϕ_A

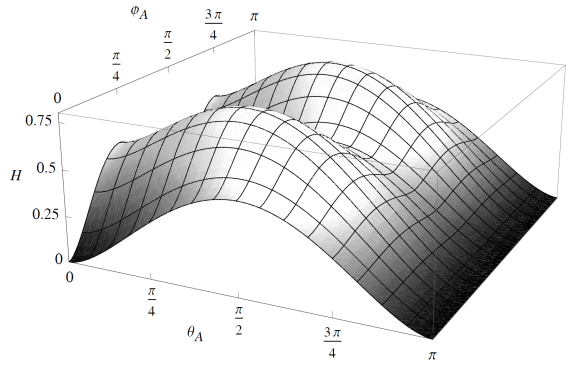


Figure 6. Shannon entropy H of the raw key depending on θ_A and ϕ_A

related to Alice's outcome with probability

$$\begin{aligned} P_e &= P_{\Phi^-} + P_{\Psi^+} + P_{\Psi^-} \\ &= \frac{1}{2}\Big(\sin^2\theta_A + \cos^4\frac{\theta_A}{2}\sin^2\big(2\phi_A\big)\Big). \end{aligned} \tag{17}$$

Assuming that Bob obtains $|\Phi^+\rangle_{Q4}$, i.e., the expected result based on Alice's measurement outcome, Eve obtains either $|\varphi_1\rangle$, $|\varphi_2\rangle$ or $|\varphi_3\rangle$ from her measurement on qubits T and U with the respective probabilities

$$\begin{aligned} P_{\varphi_1} &= \frac{\cos^4\frac{\theta_A}{2}\cos^4\phi_A}{\frac{1}{4}(3+\cos 4\phi_A)\cos^4\frac{\theta_A}{2} + \sin^4\frac{\theta_A}{2}} \\ P_{\varphi_2} &= \frac{\cos^4\frac{\theta_A}{2}\sin^4\phi_A}{\frac{1}{4}(3+\cos 4\phi_A)\cos^4\frac{\theta_A}{2} + \sin^4\frac{\theta_A}{2}} \\ P_{\varphi_3} &= \frac{-\sin^2\frac{\theta_A}{2}}{(3+\cos 4\phi_A)\cos^4\frac{\theta_A}{2} + 4\sin^4\frac{\theta_A}{2}} \end{aligned} \tag{18}$$

Furthermore, in case Bob measures an uncorrelated result, Eve obtains two out of the four auxiliary states $|\varphi_i\rangle$ at random. Hence, due to the basis transformation $T_x\big(\theta_A, \phi_A\big)$, Eve's auxiliary systems are less correlated to Bob's result compared to the application of a simple basis transformation as described in [25] [26]. In other words, Eve's information on Alice's and Bob's result is further reduced compared to the scenarios described therein.

Since Alice applies the basis transformation at random, i.e., with probability 1/2, the average error probability $\langle P_e \rangle_A$ can be directly computed using (17) and its variations based on Alice's measurement result as

$$\langle P_e \rangle_A = \frac{1}{4}\left[\sin^2\theta_A + \cos^4\frac{\theta_A}{2}\sin^2\Big(2\phi_A\Big)\right]. \tag{19}$$

Keeping in mind that Eve does not introduce any error when Alice does not use the basis transformation $T_x\big(\theta_A, \phi_A\big)$, the average collision probability $\langle P_c \rangle$ can be computed as (cf. also (18))

$$\begin{aligned} \langle P_c \rangle_A = \frac{1}{64}\Big(&53 - 4\cos\theta_A + 7\cos\big(2\theta_A\big) \\ &+ 8\cos^4\frac{\theta_A}{2}\cos\big(4\phi_A\big)\Big). \end{aligned} \tag{20}$$

In further consequence, this leads to the Shannon entropy H of the raw key, i.e.,

$$H_A = \frac{1}{2}\left[h\Big(\cos^2\frac{\theta_A}{2}\Big) + \cos^2\frac{\theta_A}{2}\; h\Big(\cos^2\phi_A\Big)\right]. \tag{21}$$

Here, the function $h(x)$ describes the binary entropy, i.e.,

$$h\big(x\big) = -x\log_2 x - \big(1-x\big)\log_2\big(1-x\big) \tag{22}$$

with $\log_2$ the binary logarithm.

As we can directly see from Figure 5, the average error probability $\langle P_e \rangle_A$ has its maximum at 1/3 with

$$\theta_{A_0} \simeq 0.39183\pi \qquad \phi_{A_0} \in \Big\{\frac{\pi}{4}, \frac{3\pi}{4}\Big\}. \tag{23}$$

For this choice of θ_A and ϕ_A we see from Figure 6 that the Shannon entropy is also maximal with $H_A \simeq 0.79248$. Hence, the adversary Eve is left with a mutual information of

$$I_{AE} = 1 - H_A = 0.20752 \tag{24}$$

This value for the mutual information is less than half of Eve's information on the raw key compared to the application of a Hadamard operation (cf. [2] [4] [22] [14]) or the application of a simplified basis transformation (cf. [25] [26]).

Unfortunately, the angle for $\theta_{A_0} \simeq 0.39183\pi$ to reach the maximum value is rather odd and might be difficult to realize in a practical implementation. In this context, difficult to realize in a physical implementation means that a transformation about an angle of $\pi/4$ or $3\pi/8$ is easier to implement in a laboratory than an angle of 0.39183π. Therefore, choosing an angle $\theta_A = 3\pi/8$ for this scenario we can compute from (19) an average error rate of $\langle P_e \rangle_A \simeq 0.33288$ and from (21) the respective Shannon entropy $H_A \simeq 0.79148$ (cf. also Figure 5 and Figure 6), which are both just insignificantly lower than their maximum values. Accordingly, Eve's mutual information on the raw key is $I_{AE} \simeq 0.20852$, which is slightly above the maximum given in (24). Hence, the security of the protocol is drastically increased using a general basis transformation compared to the application of a Hadamard operation.

V. Combined Application of General Basis Transformations

In the previous section, we discussed the application of one general basis transformation $T_x\big(\theta_A, \phi_A\big)$ on Alice's side. It is easy to see that the results for the average error probability $\langle P_e \rangle$ in (19) as well as the Shannon entropy H in (21) are the same if only Bob randomly applies the basis transformation $T_x\big(\theta_B, \phi_B\big)$ on his side.

Hence, a more interesting scenario is the combined random application of two different basis transformations, i.e., $T_x\big(\theta_A, \phi_A\big)$ on Alice's side and $T_x\big(\theta_B, \phi_B\big)$ on Bob's side.

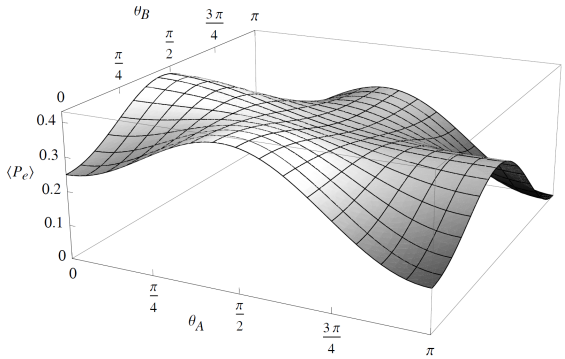


Figure 7. Error probability $\langle P_e \rangle$ depending on θ_A and θ_B. The remaining parameters ϕ_A and ϕ_B are fixed at $\pi/4$.

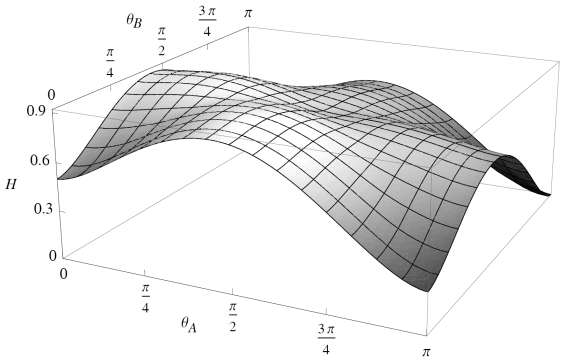


Figure 8. Shannon entropy H of the raw key depending on θ_A and θ_B. The remaining parameters ϕ_A and ϕ_B are fixed at $\pi/4$.

The application of these two different basis transformations alters the state introduced by Eve accordingly to

$$|\delta'\rangle_{1QR4TU} = T_x^{(1)}\big(\theta_A, \phi_A\big)\; T_x^{(4)}\big(\theta_B, \phi_B\big)\; |\delta\rangle_{1QR4TU} \quad (25)$$

where again the superscripts "(1)" and "(4)" indicate that $T_x\big(\theta_A, \phi_A\big)$ is applied on qubit 1 and $T_x\big(\theta_B, \phi_B\big)$ on qubit 4, respectively. Following the protocol, Alice has to undo Bob's transformation using $T_x^{-1}\big(\theta_B, \phi_B\big)$ before she can perform her Bell state measurement. Similar to the application of one basis transformation described above, Alice obtains all four Bell states with equal probability from her measurement. The state of the remaining qubits changes in a way analogous to (13) above and Bob has to reverse Alice's transformation using $T_x^{-1}\big(\theta_A, \phi_A\big)$. Hence, when Bob performs his measurement on qubits Q and 4, he does not obtain a result correlated to Alice's outcome, but all four possible Bell states with different probabilities such that an error is introduced in the protocol. As already discussed in the previous section, the results from Eve's measurement on qubits T and U are not fully correlated to Alice's and Bob's results and therefore Eve's information on the raw key bits is further reduced compared to the application of only one transformation.

Due to the fact that Alice as well as Bob choose at random whether they apply their respective basis transformation, the average error probability is calculated over all four scenarios: no transformation is applied, either Alice or Bob applies $T_x\big(\theta_A, \phi_A\big)$ or $T_x\big(\theta_B, \phi_B\big)$, respectively, or both transformations are applied. Therefore, using the results from (19) above, the overall error probability can be computed as

$$\begin{aligned}
\langle P_e \rangle_{AB} = &\frac{1}{8}\left[\sin^2\theta_A + \cos^4\frac{\theta_A}{2}\sin^2\Big(2\phi_A\Big)\right] \\
&+\frac{1}{8}\left[\sin^2\theta_B + \cos^4\frac{\theta_B}{2}\sin^2\Big(2\phi_B\Big)\right] \\
&+\frac{1}{16}\left[\sin^2\Big(\theta_A + \theta_B\Big)\right. \\
&\qquad\left.+\cos^4\frac{\theta_A + \theta_B}{2}\sin^2\Big(2\big(\phi_A + \phi_B\big)\Big)\right] \\
&+\frac{1}{16}\left[\sin^2\Big(\theta_A - \theta_B\Big)\right. \\
&\qquad\left.+\cos^4\frac{\theta_A - \theta_B}{2}\sin^2\Big(2\big(\phi_A - \phi_B\big)\Big)\right]
\end{aligned} \quad (26)$$

having its maximum at $\langle P_e \rangle_{AB} \simeq 0.41071$. One possibility to reach the maximum is to choose the angles

$$\begin{aligned}
\theta_A &= 0 & \theta_B &\simeq 0.45437\pi \\
\phi_A &= \frac{\pi}{4} & \phi_B &= \frac{\pi}{4}.
\end{aligned} \quad (27)$$

In fact, as long as $\phi_A = \pi/4$ or $\phi_A = 3\pi/4$ the value of ϕ_B can be chosen freely to reach the maximum. Therefore, the graph of the average error probability plotted in Figure 7 uses $\phi_A = \phi_B = \pi/4$.

Following the same argumentation and using (21) from above, the Shannon entropy can be calculated as

$$\begin{aligned}
H_{AB} = &\frac{1}{4}\left[h\Big(\cos^2\frac{\theta_A}{2}\Big) + \cos^2\frac{\theta_A}{2}\; h\Big(\cos^2\phi_A\Big)\right] \\
&+\frac{1}{4}\left[h\Big(\cos^2\frac{\theta_B}{2}\Big) + \cos^2\frac{\theta_B}{2}\; h\Big(\cos^2\phi_B\Big)\right] \\
&+\frac{1}{8}\left[h\Big(\cos^2\frac{\theta_A + \theta_B}{2}\Big)\right. \\
&\qquad\left.+\cos^2\frac{\theta_A + \theta_B}{2}\; h\Big(\cos^2\big(\phi_A + \phi_B\big)\Big)\right] \\
&+\frac{1}{8}\left[h\Big(\cos^2\frac{\theta_A - \theta_B}{2}\Big)\right. \\
&\qquad\left.+\cos^2\frac{\theta_A - \theta_B}{2}\; h\Big(\cos^2\big(\phi_A - \phi_B\big)\Big)\right]
\end{aligned} \quad (28)$$

having its maximum at $H_{AB} \simeq 0.9452$ (cf. Figure 8 for a plot of (28) taking $\phi_A = \phi_B = \pi/4$). This maximum is reached, for example, using

$$\begin{aligned}
\theta_{AB_0} &\simeq -0.18865\pi & \theta_{AB_0} &\simeq 0.42765\pi \\
\phi_{AB_0} &\simeq -0.22405\pi & \phi_{AB_0} &\simeq 0.36218\pi.
\end{aligned} \quad (29)$$

The maximal Shannon entropy can also be reached using other values but they are not as nicely distributed as in the case of the average error probability.

Looking again at set of values for $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$, which are more suitable for a physical implementation than the values mentioned above, one possibility for Alice and Bob is to choose

$$\begin{aligned}
\theta_A &= -\frac{3\pi}{16} & \theta_B &= \frac{7\pi}{16} \\
\phi_A &= -\frac{\pi}{4} & \phi_B &= \frac{3\pi}{8}
\end{aligned} \quad (30)$$

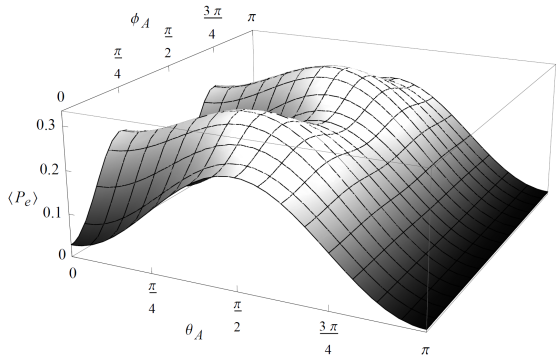


Figure 9. Error probability $\langle P_e \rangle$ depending on θ_A and ϕ_A. Here, a standard deviation $(\delta\varphi) = \pi/20$ of the angles is taken into account.

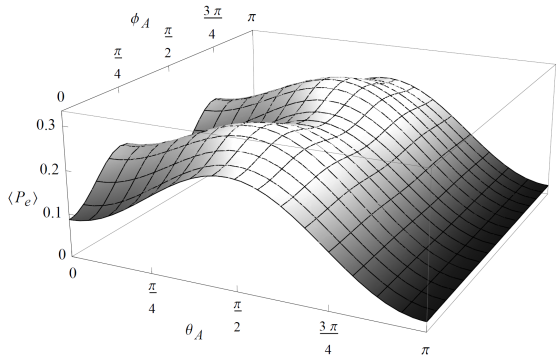


Figure 10. Error probability $\langle P_e \rangle$ depending on θ_A and ϕ_A. Here, a standard deviation $(\delta\varphi) = \pi/10$ of the angles is taken into account.

leading to an almost optimal Shannon entropy $H_{AB} \simeq 0.9399$ and a average respective error probability $\langle P_e \rangle_{AB} \simeq 0.39288$. Keeping ϕ_A and ϕ_B fixed – as already discussed in the previous section – such that

$$\begin{aligned} \theta_A &= \frac{3\pi}{16} \qquad \theta_B = \frac{7\pi}{16} \\ \phi_A &= \frac{\pi}{4} \qquad \phi_B = \frac{\pi}{4} \end{aligned} \tag{31}$$

the same average error probability $\langle P_e \rangle_{AB} \simeq 0.39288$ and a slightly smaller Shannon entropy $H_{AB} \simeq 0.91223$ compared to the previous values are achieved. Hence, we see that using a set of parameters more suitable for a physical implementation still results in a high error rate and leaves Eve's mutual information I_{AE} below 10%.

VI. Robustness of the Optimal Angles

As already pointed out above, the optimal values for the angles $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$ are rather odd and might not be easy to create in a laboratory. Especially when looking at the combined application of basis transformations at Alice's and Bob's side, it will be very difficult to implement the exact angles given in (29) to achieve the optimal values for $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$. Furthermore, due to physical limitations the apparatus, which is used to adjust the angles $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$ in the laboratory can in general not be considered perfect. To model an error introduced by this imperfect apparatus, we will use a Gaussian distribution to describe the angles $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$. In this context, we will look in detail at two rather small standard deviations from the optimal angles, i.e., in the order of 5% and 10% of π, and how this deviation from the optimal angle affects the security of the protocol.

In detail, a Gaussian distribution for some angle x can be described as

$$f\Big[x, x_0, (\delta x)\Big] = \frac{1}{\sqrt{2\pi}(\delta x)}\, e^{-\frac{(x-x_0)^2}{2(\delta x)^2}} \tag{32}$$

with x_0 the expected value (e.g., the optimal angle for some configuration) and (δx) the standard deviation (the deviation from that optimal angle). Accordingly, the mean value is described by the area under the curve, and is computed by the integral

$$\int_{-\infty}^{\infty} f\Big[x, x_0, (\delta x)\Big] dx = 1. \tag{33}$$

Based on this definition, the mean value for the cosine function $\cos(\lambda x)$ of some angle x and a real number λ can be computed directly as

$$\begin{aligned} \overline{\cos(\lambda x)} &= \int_{-\infty}^{\infty} f\Big[x, x_0, (\delta x)\Big] \cos(\lambda x) dx \\ &= e^{-\lambda^2 \frac{(\delta x)^2}{2}} \cos(\lambda x_0). \end{aligned} \tag{34}$$

Taking this approach into account, we can rephrase the calculations leading to the expected error probability $\langle P_e \rangle_A$ given in (19) and $\langle P_e \rangle_{AB}$ given in (26). This leads to a representation of the expected error probability depending on the deviation from the optimal value for the angles $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$, respectively. The computation of the Shannon entropy H_A described in (21) and H_{AB} described in (28) using this approach is more complex due to the application of the binary logarithm when computing the binary entropy h. Hence, we will not provide it here.

First, we describe this extension with regards to the expected error probability $\langle P_e \rangle_A$ in (19). Therefore, we use the equalities

$$\begin{aligned} \sin^2(x) &= \frac{1}{2}\Big[1 - \cos(2x)\Big] \qquad \text{and} \\ \cos^4(x) &= \frac{1}{8}\Big[\cos(4x) + 4\cos(2x) + 3\Big] \end{aligned} \tag{35}$$

as well as the definition in (34) above. After a few computations we see that

$$\begin{aligned} \overline{\langle P_e \rangle_A} &= \frac{1}{4}\Bigg[\frac{1}{2}\Big(1 - \overline{\cos(2\theta_A)}\Big) \\ &\quad + \frac{1}{8}\Big(\overline{\cos(2\theta_A)} + 4\,\overline{\cos(\theta_A)} + 3\Big) \\ &\quad \times \frac{1}{2}\Big(1 - \overline{\cos(4\phi_A)}\Big)\Bigg] \\ &= \frac{1}{4}\Bigg[\frac{1}{2}\Big(1 - e^{-2(\delta\varphi)^2}\cos(2\theta_{A_0})\Big) \\ &\quad + \frac{1}{8}\Big(e^{-2(\delta\varphi)^2}\cos(2\theta_{A_0}) \\ &\quad + 4\, e^{-\frac{1}{2}(\delta\varphi)^2}\cos(\theta_{A_0}) + 3\Big) \\ &\quad \times \frac{1}{2}\Big(1 - e^{-8(\delta\varphi)^2}\cos(4\phi_{A_0})\Big)\Bigg] \end{aligned} \tag{36}$$

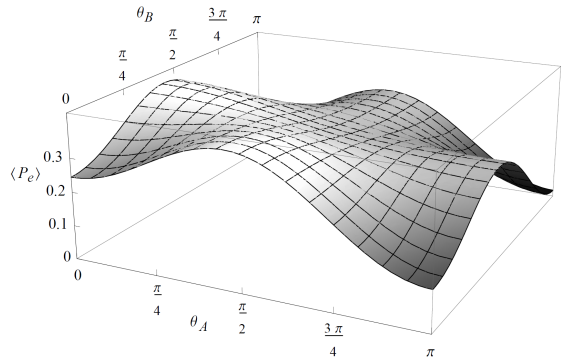


Figure 11. Error probability $\langle P_e \rangle$ depending on θ_A and θ_B. The remaining parameters ϕ_A and ϕ_B are fixed at $\pi/4$. Here, a standard deviation $(\delta\varphi) = \pi/20$ of the angles is taken into account.

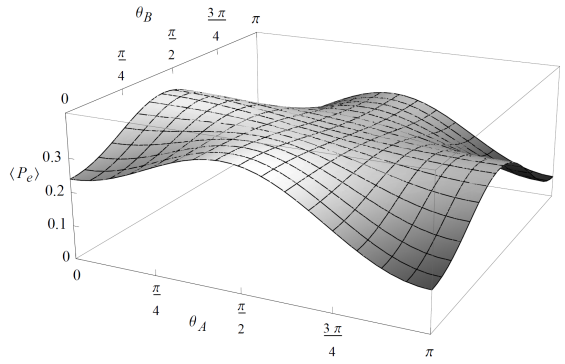


Figure 12. Error probability $\langle P_e \rangle$ depending on θ_A and θ_B. The remaining parameters ϕ_A and ϕ_B are fixed at $\pi/4$. Here, a standard deviation $(\delta\varphi) = \pi/10$ of the angles is taken into account.

For reasons of simplicity, we use the same standard deviation for both angles θ_A and ϕ_A such that $(\delta\theta_A) = (\delta\phi_A) = (\delta\varphi)$.

As we can conclude from (36), a deviation from the optimal angles θ_{A_0} and ϕ_{A_0} results in a reduced expected error probability $\overline{\langle P_e \rangle_A}$ (cf. also Figure 9 and Figure 10). Additionally, the expected error probability does not reach 0 any more due to the attenuation by the Gaussian distribution. Considering, for example, a standard deviation $(\delta\varphi) = \pi/20$, the maximum is slightly reduced by 4% (compared to (19)) from 1/3 to $\overline{\langle P_e \rangle_A} \simeq 0.3194$. This value is achieved using

$$\theta_{A_0} \simeq 0.40108\pi \qquad \phi_{A_0} \in \left\{\frac{\pi}{4}, \frac{3\pi}{4}\right\}. \tag{37}$$

Furthermore, taking a bigger standard deviation $(\delta\varphi) = \pi/10$, the maximum is reduced by almost 14% to $\overline{\langle P_e \rangle_A} \simeq 0.28826$.

It is also easy to see from (36) that the more precise the apparatus works, i.e., the smaller $(\delta\varphi)$ becomes, the closer the values $\overline{\langle P_e \rangle_A}$ and $\langle P_e \rangle_A$ get. Hence, we reach the limit

$$\lim_{(\delta\varphi)\to 0} \overline{\langle P_e \rangle_A} = \frac{1}{4}\left[\sin^2\theta_{A_0} + \cos^4\frac{\theta_{A_0}}{2}\sin^2\Big(2\phi_{A_0}\Big)\right] \tag{38}$$

which directly corresponds to $\langle P_e \rangle_A$ in (19).

Similarly, looking at the expected error probability $\langle P_e \rangle_{AB}$ in (26) when two different basis transformations are applied at Alice's and Bob's side, we can rewrite (26) such that

$$\begin{aligned}\langle P_e \rangle_{AB} = \frac{1}{2}\langle P_e \rangle_A + \frac{1}{2}\langle P_e \rangle_B \\ + \frac{1}{4}\langle P_e \rangle_{A+B} + \frac{1}{4}\langle P_e \rangle_{A-B}\end{aligned} \tag{39}$$

where

$$\begin{aligned}\langle P_e \rangle_{A+B} = \frac{1}{4}\Bigg[\sin^2\Big(\theta_A + \theta_B\Big) + \\ \cos^4\frac{\theta_A + \theta_B}{2}\sin^2\Big(2(\phi_A + \phi_B)\Big)\Bigg]\end{aligned} \tag{40}$$

and $\langle P_e \rangle_{A-B}$ accordingly. Based on these two equations, we can directly calculate the expected error probability $\overline{\langle P_e \rangle_{AB}}$ as

$$\begin{aligned}\overline{\langle P_e \rangle_{AB}} = \frac{1}{2}\overline{\langle P_e \rangle_A} + \frac{1}{2}\overline{\langle P_e \rangle_B} \\ + \frac{1}{4}\overline{\langle P_e \rangle_{A+B}} + \frac{1}{4}\overline{\langle P_e \rangle_{A-B}}.\end{aligned} \tag{41}$$

In this case, we again use the same standard deviation for all angles, such that $(\delta\theta_A) = (\delta\phi_A) = (\delta\theta_B) = (\delta\phi_B) = (\delta\varphi)$. An explicit representation (as we have provided it in (36) for $\overline{\langle P_e \rangle_A}$) of the above expression would be rather lengthy and therefore is not provided here. Nevertheless, the terms are similar to the result in (36) and we can directly compute the new maxima of the expected error probability. Considering again a standard deviation $(\delta\varphi) = \pi/20$, the maximum is slightly reduced by approximately 4% from 0.41071 to $\overline{\langle P_e \rangle_{AB}} \simeq 0.39599$ compared to (26). This value is achieved using

$$\begin{aligned}\theta_{A_0} = 0 \qquad \theta_{B_0} \simeq 0.45264\pi \\ \phi_{A_0} = \frac{\pi}{4} \qquad \phi_{B_0} = \frac{\pi}{4}.\end{aligned} \tag{42}$$

Applying a bigger standard deviation of $(\delta\varphi) = \pi/10$, these values just slightly change, i.e.,

$$\begin{aligned}\theta_{A_0} = 0 \qquad \theta_{B_0} \simeq 0.44703\pi \\ \phi_{A_0} = \frac{\pi}{4} \qquad \phi_{B_0} = \frac{\pi}{4}.\end{aligned} \tag{43}$$

and the maximum is further decreased by approximately 11% to $\overline{\langle P_e \rangle_{AB}} \simeq 0.36444$.

Analogous to (38), it is easy to see that also the expected error probability $\overline{\langle P_e \rangle_{AB}}$ for the combined application of two basis transformations reaches a limit when $(\delta\varphi)$ approaches 0, which corresponds to $\langle P_e \rangle_{AB}$ from (26) above, i.e.,

$$\begin{aligned}\lim_{(\delta\varphi)\to 0} \overline{\langle P_e \rangle_{AB}} = \lim_{(\delta\varphi)\to 0} \frac{1}{2}\overline{\langle P_e \rangle_A} \\ + \lim_{(\delta\varphi)\to 0} \frac{1}{2}\overline{\langle P_e \rangle_B} \\ + \lim_{(\delta\varphi)\to 0} \frac{1}{4}\overline{\langle P_e \rangle_{A+B}} \\ + \lim_{(\delta\varphi)\to 0} \frac{1}{4}\overline{\langle P_e \rangle_{A-B}}.\end{aligned} \tag{44}$$

As already pointed out above, when it comes to the computation of the Shannon entropy H, the terms are rather complex to evaluate symbolically due to the application of the binary entropy. Based on the above computations in (36) and (41) in context with the expected error probability, we can assume that also the graphs describing the Shannon entropy will be similar to Figure 6 and Figure 8. Due to the application of the Gaussian

TABLE I. OVERVIEW OF THE ERROR RATE $\langle P_E \rangle$ AND EVE'S INFORMATION I_{AE} ON THE RAW KEY BITS FOR DIFFERENT VALUES OF $\theta_{A,B}$ AND $\phi_{A,B}$.

	$\phi_A = 0$	$\phi_A = \frac{\pi}{2}$	$\phi_A = \frac{\pi}{4}$
$\phi_B = 0$	$\theta_A = 0, \theta_B = 0$ $\langle P_e \rangle = 0$ $I_{AE} = 1$	$\theta_A = \frac{\pi}{2}, \theta_B = 0$ $\langle P_e \rangle = 0.25$ $I_{AE} = 0.5$	$\theta_A = \frac{3\pi}{8}, \theta_B = 0$ $\langle P_e \rangle \simeq 0.333$ $I_{AE} \simeq 0.208$
$\phi_B = \frac{\pi}{2}$		$\theta_A = \frac{\pi}{2}, \theta_B = \frac{\pi}{4}$ $\langle P_e \rangle = 0.25$ $I_{AE} \simeq 0.45$	$\theta_A = 0, \theta_B = \frac{\pi}{2}$ $\langle P_e \rangle \simeq 0.406$ $I_{AE} = 0.125$
$\phi_B = \frac{\pi}{4}$			$\theta_A = \frac{3\pi}{16}, \theta_B = \frac{7\pi}{16}$ $\langle P_e \rangle = 0.393$ $I_{AE} = 0.088$

TABLE II. OVERVIEW OF THE MEAN VALUE OF THE ERROR RATE $\overline{\langle P_E \rangle}$ FOR DIFFERENT STANDARD VARIATIONS ($\delta\varphi$) AND DIFFERENT VALUES OF $\theta_{A,B}$ AND $\phi_{A,B}$.

	$\theta_A = \frac{3\pi}{8}, \theta_B = 0$ $\phi_A = \frac{\pi}{4}, \phi_B = 0$	$\theta_A = 0, \theta_B = \frac{\pi}{2}$ $\phi_A = \frac{\pi}{4}, \phi_B = \frac{\pi}{2}$	$\theta_A = \frac{3\pi}{16}, \theta_B = \frac{7\pi}{16}$ $\phi_A = \frac{\pi}{4}, \phi_B = \frac{\pi}{4}$
$(\delta\varphi) = 0$	$\overline{\langle P_e \rangle} \simeq 0.333$	$\overline{\langle P_e \rangle} \simeq 0.406$	$\overline{\langle P_e \rangle} \simeq 0.393$
$(\delta\varphi) = \frac{\pi}{20}$	$\overline{\langle P_e \rangle} \simeq 0.318$	$\overline{\langle P_e \rangle} \simeq 0.386$	$\overline{\langle P_e \rangle} \simeq 0.381$
$(\delta\varphi) = \frac{\pi}{10}$	$\overline{\langle P_e \rangle} \simeq 0.286$	$\overline{\langle P_e \rangle} \simeq 0.359$	$\overline{\langle P_e \rangle} \simeq 0.355$

distribution (and the respective standard deviation) for the angles $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$ the graphs will be attenuated like it is depicted for the error probability in Figure 9 to Figure 12. Thus, the maximum Shannon entropy will also be decreased, which means that the maximum of the adversary's information I_{AE} will be increased. As we have seen above, even if we consider a rather large deviation of $\pi/10$, the variation of the Shannon entropy will be around 15%. Hence, we can assume that the increase of the adversary's information will not become critical in such a way that the protocol becomes insecure.

VII. Security Implications

The results presented in the previous sections have direct implications on the security of QKD protocols based on entanglement swapping. Where in some QKD protocols [14] [17] [18] a random application of a Hadamard operation is used to detect an eavesdropper and secure the protocol, the above results indicate that the Hadamard operation is not the optimal choice. Using the Hadamard operation leaves an adversary with a mutual information $I_{AE} = 0.5$ and an expected error probability $\langle P_e \rangle = 0.25$ (cf. Table I), which is comparable to standard prepare and measure protocols [2]–[4].

Giving Alice an increased degree of freedom, i.e., choosing both angles θ_A and ϕ_A of the basis transformation freely, she is able to further decrease the adversary's information about the raw key bits. By shifting ϕ_A from $\pi/2$ to $\pi/4$ and θ_A from $\pi/2$ or $\pi/4$ to $3\pi/8$, the adversary's information is reduced to $I_{AE} \simeq 0.208$ (cf. (21)). This is a reduction by almost 60% compared to QKD schemes described in [2]–[4] [14] [18] and more than 50% compared to the combined application of two different basis transformations (cf. also [25] [26]). At the same time, the expected error probability is increased by one third to $\langle P_e \rangle_A \simeq 0.333$ (cf. (19)). Hence, an adversary does not only obtain fewer information about the raw key bits but also introduces more errors and therefore is easier to detect.

Following these arguments, the best strategy for Alice and Bob is to apply different basis transformations at random to reduce the adversary's information to a minimum. As already pointed out above, the minimum of $I_{AE} \simeq 0.0548$ is reached with a rather odd configuration for $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$ as described Section V. Hence, it is important to look at configurations more suitable for physical implementations, i.e., configurations of $\theta_{\{A,B\}}$ and $\phi_{\{A,B\}}$ described by simpler fractions of π as given in (30) and (31). In this case, we showed that $\phi_{\{A,B\}}$ can be fixed at $\phi_A = \phi_B = \pi/4$ and with $\theta_A = 3\pi/16$ and $\theta_B = 7\pi/16$ almost maximal values can be achieved resulting in $I_{AE} \simeq 0.088$ and $\langle P_e \rangle_{AB} \simeq 0.393$ (cf. (31) and also Table I).

Regarding physical implementations, another – even simpler – configuration can be found, involving only $\pi/2$ and $\pi/4$ rotations (cf. Table I). In this case, $\theta_A = 0$, $\phi_A = \pi/4$ and $\theta_B = \phi_B = \pi/2$, which leaves the expected error probability at $\langle P_e \rangle_{AB} \simeq 0.406$. The adversary's information is nowhere near the minimum but still rather low at $I_{AE} = 0.125$.

Although the configurations described above are much simpler with regards to the angles that have to be prepared, we also pointed out that a potential deviation from these angles has to be taken into account. This deviation is coming from the imperfect configuration of the physical apparatus and can be modeled using a Gaussian distribution. Fortunately, the above configurations are very robust in withstanding this variance such that even a large deviation of $\pi/10$ does not cause a large variation in the expected error rate. For example, with a deviation of $(\delta\varphi) = \pi/10$ the error probability $\overline{\langle P_e \rangle_{AB}}$ is decreased only by about 11% compared to the optimal error probability $\langle P_e \rangle_{AB}$. This also holds compared to the simpler configurations above, as described in Table II. Hence, even if the angles can not be configured precisely, the expected error probability is not drastically decreased and the security of the protocol is not jeopardized.

In terms of security, these results represent a huge advantage over existing QKD protocols based on entanglement swapping [14], [17], [18] or standard prepare and measure protocols [2]–[4]. As pointed out, such protocols usually have an expected error probability of $\langle P_e \rangle = 0.25$ and a mutual information $I_{AE} = 0.5$. Due to the four degrees of freedom, the error rate is between one third ($\langle P_e \rangle_{AB} \simeq 0.333$) and more than one half ($\langle P_e \rangle_{AB} = 0.411$) higher in the scenarios described here than in the standard protocols, which makes it easier to detect an adversary.

VIII. Conclusion

In this article, we discussed the effects of basis transformations on the security of QKD protocols based on entanglement swapping. Additionally, we looked at the robustness of these QKD protocols against an imperfect preparation of these basis transformations. We showed that the Hadamard operation, a transformation from the Z- into the X-basis often used in prepare and measure protocols, is not optimal in connection with entanglement swapping based protocols. Starting from a general basis transformation described by two angles θ and ϕ, we analyzed the effects on the security when the adversary follows a collective attack strategy. We showed that the application of a basis transformation by one of the communication parties decreases the adversary's information to $I_{AE} \simeq 0.2075$, which is less than half of the information compared to an application of the Hadamard operation. At

the same time, the average error probability introduced by the presence of the adversary increases to $\langle P_e \rangle = 1/3$. Hence, the application of one general basis transformation is more effective, i.e., reveals even less information to the adversary, than the application of a simplified basis transformation as given in [25] [26]. A combined application of two different basis transformations further reduces the adversary's information to about $I_{AE} \simeq 0.0548$ at an average error probability of $\langle P_e \rangle \simeq 0.4107$.

Since the configuration of the angles θ and ϕ to reach these maximal values is not very suitable for a physical implementation, we also showed that values for $\langle P_e \rangle$ and I_{AE}, which are almost maximal, can be reached with more convenient values for θ and ϕ. In this case, the adversary's information is still $I_{AE} < 0.1$ with an expected error probability $\langle P_e \rangle \simeq 0.393$ for a combined application of two basis transformations.

To take the effects of an imperfect preparation of these angles into account, the angles are described using a Gaussian distribution. Based on that model, the effects of two certain values for the standard deviation on the expected error probability are analyzed. In this context, we showed that the variation in the expected error probability is with 11% - 14% rather low even for a large deviation of $\pi/10$. With regards to the application of the Gaussian distribution, we showed that also for these more practical values of θ and ϕ the variation of the expected error probability as well as the increase of the adversary's information is rather low. Hence, we can conclude that the protocol is robust against this kind of error and the gain of the adversary's information will not become critical in such a way that the protocol becomes insecure.

These results have a direct impact on the security of such protocols. Due to the reduced information of an adversary and the high error probability introduced during the attack strategy, Alice and Bob are able to accept higher error thresholds compared to standard entanglement-based QKD protocols.

REFERENCES

[1] S. Schauer and M. Suda, "Optimal Choice of Basis Transformations for Entanglement Swapping Based QKD Protocols," in ICQNM 2014, The Eighth International Conference on Quantum, Nano and Micro Technologies. IARIA, 2014, pp. 8–13.

[2] C. H. Bennett and G. Brassard, "Public Key Distribution and Coin Tossing," in Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing. IEEE Press, 1984, pp. 175–179.

[3] A. Ekert, "Quantum Cryptography Based on Bell's Theorem," Phys. Rev. Lett., vol. 67, no. 6, 1991, pp. 661–663.

[4] C. H. Bennett, G. Brassard, and N. D. Mermin, "Quantum Cryptography without Bell's Theorem," Phys. Rev. Lett., vol. 68, no. 5, 1992, pp. 557–559.

[5] D. Bruss, "Optimal Eavesdropping in Quantum Cryptography with Six States," Phys. Rev. Lett, vol. 81, no. 14, 1998, pp. 3018–3021.

[6] A. Muller, H. Zbinden, and N. Gisin, "Quantum Cryptography over 23 km in Installed Under-Lake Telecom Fibre," Europhys. Lett., vol. 33, no. 5, 1996, pp. 335–339.

[7] A. Poppe et al., "Practical Quantum Key Distribution with Polarization Entangled Photons," Optics Express, vol. 12, no. 16, 2004, pp. 3865–3871.

[8] A. Poppe, M. Peev, and O. Maurhart, "Outline of the SECOQC Quantum-Key-Distribution Network in Vienna," Int. J. of Quant. Inf., vol. 6, no. 2, 2008, pp. 209–218.

[9] M. Peev et al., "The SECOQC Quantum Key Distribution Network in Vienna," New Journal of Physics, vol. 11, no. 7, 2009, p. 075001.

[10] N. Lütkenhaus, "Security Against Eavesdropping Attacks in Quantum Cryptography," Phys. Rev. A, vol. 54, no. 1, 1996, pp. 97–111.

[11] ——, "Security Against Individual Attacks for Realistic Quantum Key Distribution," Phys. Rev. A, vol. 61, no. 5, 2000, p. 052304.

[12] P. Shor and J. Preskill, "Simple Proof of Security of the BB84 Quantum Key Distribution Protocol," Phys. Rev. Lett., vol. 85, no. 2, 2000, pp. 441–444.

[13] A. Cabello, "Quantum Key Distribution without Alternative Measurements," Phys. Rev. A, vol. 61, no. 5, 2000, p. 052312.

[14] ——, "Reply to "Comment on "Quantum Key Distribution without Alternative Measurements"","," Phys. Rev. A, vol. 63, no. 3, 2001, p. 036302.

[15] ——, "Multiparty Key Distribution and Secret Sharing Based on Entanglement Swapping," quant-ph/0009025 v1, 2000.

[16] F.-G. Deng, G. L. Long, and X.-S. Liu, "Two-step quantum direct communication protocol using the Einstein-Podolsky-Rosen pair block," Phys. Rev. A, vol. 68, no. 4, 2003, p. 042317.

[17] D. Song, "Secure Key Distribution by Swapping Quantum Entanglement," Phys. Rev. A, vol. 69, no. 3, 2004, p. 034301.

[18] C. Li, Z. Wang, C.-F. Wu, H.-S. Song, and L. Zhou, "Certain Quantum Key Distribution achieved by using Bell States," International Journal of Quantum Information, vol. 4, no. 6, 2006, pp. 899–906.

[19] C. H. Bennett et al., "Teleporting an Unknown Quantum State via Dual Classical and EPR Channels," Phys. Rev. Lett., vol. 70, no. 13, 1993, pp. 1895–1899.

[20] M. Zukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, ""Event-Ready-Detectors" Bell State Measurement via Entanglement Swapping," Phys. Rev. Lett., vol. 71, no. 26, 1993, pp. 4287–4290.

[21] B. Yurke and D. Stolen, "Einstein-Podolsky-Rosen Effects from Independent Particle Sources," Phys. Rev. Lett., vol. 68, no. 9, 1992, pp. 1251–1254.

[22] Y.-S. Zhang, C.-F. Li, and G.-C. Guo, "Comment on "Quantum Key Distribution without Alternative Measurements"," Phys. Rev. A, vol. 63, no. 3, 2001, p. 036301.

[23] S. Schauer and M. Suda, "A Novel Attack Strategy on Entanglement Swapping QKD Protocols," Int. J. of Quant. Inf., vol. 6, no. 4, 2008, pp. 841–858.

[24] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information. Cambridge University Press, 2000.

[25] S. Schauer and M. Suda, "Security of Entanglement Swapping QKD Protocols against Collective Attacks," in ICQNM 2012 , The Sixth International Conference on Quantum, Nano and Micro Technologies. IARIA, 2012, pp. 60–64.

[26] ——, "Application of the Simulation Attack on Entanglement Swapping Based QKD and QSS Protocols," International Journal on Advances in Systems and Measurements, vol. 6, no. 1&2, 2013, pp. 137–148.

Furnace Operational Parameters and Reproducible Annealing of Thin Films

Victor Ovchinnikov
Department of Aalto Nanofab
School of Electrical Engineering, Aalto University
Espoo, Finland
e-mail: Victor.Ovchinnikov@aalto.fi

***Abstract*—Annealing of thin silver films on oxidized silicon substrates in different furnaces is studied. It is shown that identical temperatures and durations of thermal treatment do not guarantee reproducibility, i.e., the annealing provides different results, e.g., shape and size of nanostructures in different furnaces. To clarify the source of the variation, morphology and optical properties of the samples are analyzed. Spectroscopic ellipsometry is used to measure thickness and composition of the oxide layer before and after annealing. Reflectance spectra, obtained for different angles of incidence and polarizations, demonstrate the dependence of sample plasmonic properties on the furnace design. Additionally, a numerical simulation of the heating process in a diffusion furnace has been performed. It is concluded that uncontrollable overheating of silver film with regards to the substrate produced by thermal radiation of the environment leads to variation in annealing results.**

Keywords-silver thin film; diffusion furnace; annealing; nanostructures.

I. INTRODUCTION

Recently, the variation of annealing results for thin silver films heated at identical temperatures and during identical times has been demonstrated by using thermal processing tools of different designs [1].

Annealing is well known and broadly used in microfabrication for controlled heating of inorganic materials to alter their properties. In case of polymer, similar heat treatment is called baking, curing or drying. From the beginning of semiconductor technology, annealing has been used to modify properties of thin films, substrates and interfaces. Annealing is not a major microfabrication method like lithography or etching, however it is always included in fabrication of all micro- and nanodevices. The main application areas of annealing are doping of semiconductors, silicide formation, densification of deposited films, contact resistance decreasing, sample surface conditioning, etc. [2]. Annealing is done by heat treatment equipment, which can have completely different designs: convection and diffusion furnaces, hot plates and rapid thermal processing tools, infrared (IR) and curing ovens and so on. At the same time, annealing is usually characterized by only process temperature and time. Furthermore, temperature can be measured in different places: on a sample surface, in a fixed point of the heated volume, or on heating surfaces. Clearly, results of the annealing of the same samples, at the same time and temperature, but in various furnaces can be different. In this work, we anneal identical samples in identical conditions (time and temperature), but in different furnaces and study the effect of the furnace design on the obtained results.

The paper is organized as follows. In the subsequent Section II, the solved problem is formulated. In Section III, the details of sample preparation are given, the designs of three annealed furnaces are described and the measurement procedures are presented. In Section IV, the results of the work are demonstrated by scanning electron microscope (SEM) images, optical parameters of the layers obtained by spectroscopic ellipsometry, reflectance spectra of the samples for different angles of incidence and polarizations and also by a simulation of the heating process. The effect of furnace design on silver film annealing is discussed in Section V. In Section VI, the conclusions are drawn.

II. PROBLEM STATEMENT

The standard description of annealing in publications includes only temperature and duration of the process [3, 4]. Sometimes information about ambient or gas flow is added [5, 6]. The heat equipment and the sample position in the process chamber are rarely written about [7, 8]. However, different annealing tools deliver heat energy to a sample in different ways, which directly affects the obtained results.

During annealing heat exchange between the sample and a furnace is performed by thermal conductivity, convection and thermal radiation. Depending on the furnace design, one or another heat transfer mode may be dominant. For example, a hot plate mainly heats a sample by thermal conductivity, a diffusion furnace by thermal radiation and convection, an IR oven - by thermal radiation. In all furnaces heat is not only generated, but it is also dissipated. As a result, the sample temperature is controlled by thermal balance between the heating and cooling processes.

Additionally, the sample thermal parameters (emissivity, thermal conductivity and heat capacitance) and sample arrangement in a furnace (position, holder design and shields) affect the heating process dynamics and the sample temperature. The most complicate situation happens in the instance of phase transition of the heated thin film, e.g., melting or recrystallization. As a consequence, the sample emissivity is changed and the new thermal balance is installed.

In this paper, we demonstrate that identical heating ramp, temperature and time of the annealing are not sufficient

conditions for reproducibility of nanostructures fabricated by the annealing of thin silver films. We compare the designs of three annealing tools and analyze the relative strength of different heating modes in all tools. On the basis of optical properties and crystalline structure of the annealed and as-deposited samples we draw conclusions about melting and crystallization of silver nanostructures. To find the temperature field of the furnace and to estimate the real sample temperature we simulate the heating process in the diffusion furnace for different gas flows and sample emissivities. The obtained results are used to find correlation between annealing conditions and properties of silver nanostructures.

III. EXPERIMENTS

Four identical samples were prepared to compare annealing in different furnaces. For this purpose, a 12 nm thick silver film was deposited by electron beam evaporation at a rate 0.2 nm/s. As a substrate was used a 4" silicon wafer with 21 nm layer of thermal oxide. After the deposition the whole wafer was cut in four quotas, which were further processed separately. Annealing was done at 400 ºC during 5 minutes with a heating ramp of 21 ºC/min, and a cooling ramp of 3.6 ºC/min. However, all samples were processed in various furnaces (Fig. 1).

The sample #1 was annealed in the diffusion furnace (Fig. 1a). A 4" silicon wafer on a quartz boat was used as a sample holder, which was located in the centre of the furnace during the experiment. It was assumed that heat exchange through the quartz boat was negligible. The quartz furnace tube had a 4½ inch diameter, was 96 cm in length and with 3 mm thick walls. The resistive heater (orange strips in Fig. 1a) was situated around the tube with a gap of 1 cm. The furnace temperature was controlled according to thermocouple measurements on the tube surface. Room temperature nitrogen with a flow of 8.3×10^{-5} standard m^3/s was introduced in the furnace along its axis.

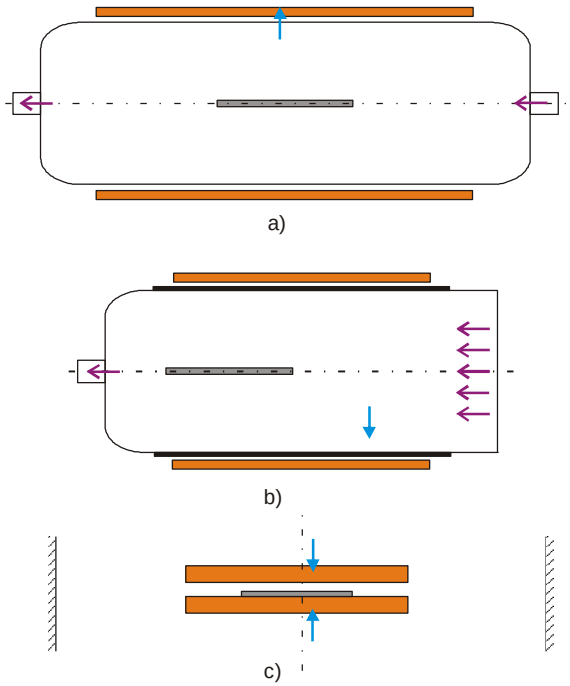


Figure 1. Design of the diffusion furnace (a), the fast ramping furnace (b) and the hot plate (c). Thermocouple positions and gas flows are denoted by blue and vilolet arrows, respectively. Heating surfaces are orange.

The sample #2 was annealed in a fast ramping furnace (Fig. 1b). The temperature, gas flow and process duration were the same as in the diffusion furnace (Fig. 1a). However, in the fast ramping furnace the quartz tube length was 35 cm and the sample position was close to the exhaust of the furnace. Tungsten lamps were used as heaters. The quartz tube was covered by a heat absorbing shield (black lines in Fig. 1b). The gas temperature in the tube was measured by thermocouple and was used for process control. Nitrogen was introduced through an array of holes in the right part of the furnace.

The sample #3 was annealed between two hot plates in vacuum (Fig. 1c). The diameter of both hot plates was 10 cm and they were separated by a 2.5 mm gap. The chamber wall temperature was close to room temperature.

The sample #4 is deposited silver film. The silver films were deposited in the e-beam evaporation system IM-9912 (Instrumentti Mattila Oy) at a base pressure of 2.7×10^{-5} Pa and at room temperature of the substrate. Annealing of the sample #1 was done in the diffusion furnace THERMCO Mini Brute MB-71. Annealing of the sample #2 was done in the fast ramping furnace PEO-601 from ATV Technology GmBH. Annealing of the sample #3 was done in the wafer bonder AML AWB-04 from Microengineering Ltd.

For selective etching of the samples were used diluted nitric acid (HNO_3 min. 69% from Honeywell) HNO_3:H_2O = 1:1 and diluted buffered hydrofluoric acid (BHF) BHF:H_2O = 1:3. As BHF was used standard ammonium fluoride etching mixture AF 90-10 LST from Honeywell. Two small

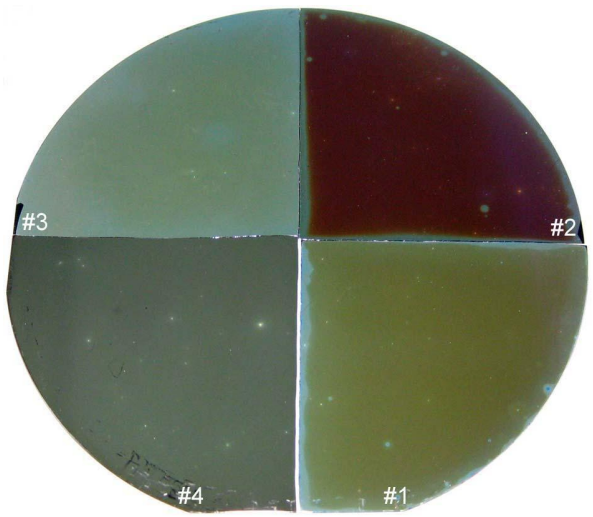


Figure 2. Optical images of the annealed (#1 - #3) and as-deposited (#4) samples.

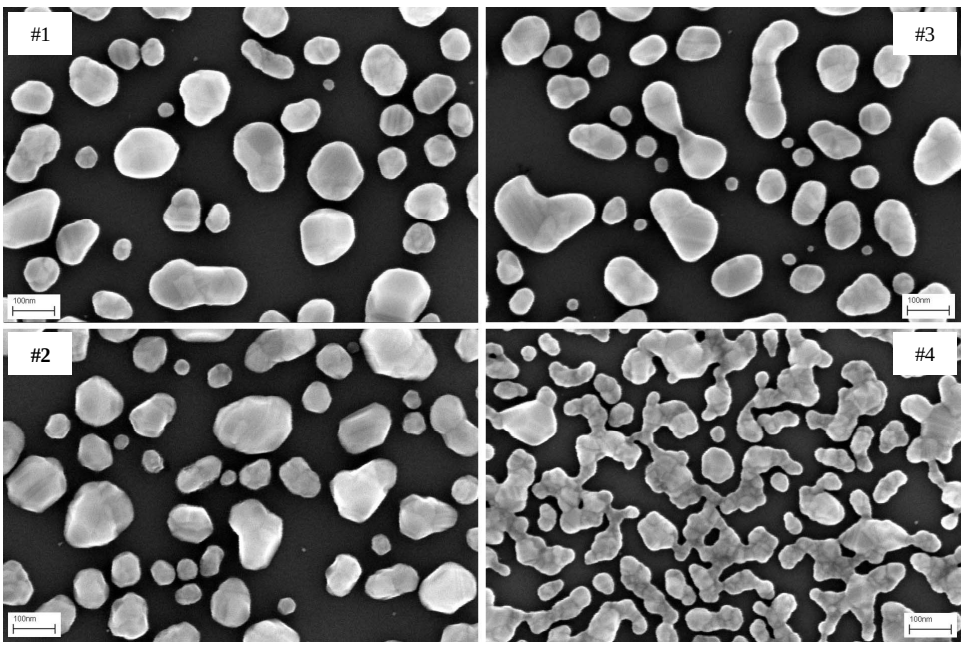


Figure 3. Plan view SEM images of the annealed (#1 - #3) and as-deposited (#4) samples.

chips (1×1 cm^2) were prepared from every annealed sample. The first chip was etched by diluted HNO_3 during 50 seconds without preliminary treatment (HNO_3 processing), the second one was dipped in diluted BHF for 10 seconds, rinsed in deionized water, dried by nitrogen and etched by diluted HNO_3 during 50 seconds (BHF/HNO_3 processing).

Plan view and tilted SEM images of the samples were observed with the Zeiss Supra 40 field emission scanning electron microscope. Reflectance measurements were carried out using the FilmTek 4000 reflectometer in the spectral range of 400–1700 nm or the spectrometer Axiospeed FT (Opton Feintechnik GmbH) in the range of 400–750 nm. Spectroscopic ellipsometry and reflectance measurements in the range of 650–1700 nm were done by spectroscopic ellipsometer SE 805 (SENTECH Instruments GmbH). The crystalline structure of the silver films was estimated by RHEED (reflection high-energy electron diffraction) observations with the help of the diffractometer embedded in a molecular beam epitaxy tool. EDS (Energy-Dispersive X-ray Spectroscopy) analyses were done with the help of a Genesis Apex 4i EDS system.

IV. RESULTS

Fig. 2 shows the optical image of three annealed samples (#1–#3) and deposited silver film (#4). The picture was taken with a digital camera with a flash. Despite identical temperature and time of annealing all samples have different colored surfaces. The sample #1 is yellow-green, the sample #2 is brown-red, the sample #3 is yellow-blue and the as-deposited sample is grey. Bulk silver is a perfect reflector, however, nanostructured silver possesses plasmon resonances, which modify reflection spectra of the samples [7, 9, 10]. Therefore, the obtained colour variation could be explained by silver nanostructures formed on the sample surface instead of the continuous film. For a detailed understanding of the effect of annealing conditions on film transformation, the structure and optical properties of the prepared samples were studied by SEM, spectroscopic ellipsometry and reflectometry.

A. Morphology of silver nanostructures

To justify the formation of silver nanostructures all samples were observed in SEM (Fig. 3). The as-deposited silver film (sample #4) is already discontinuous and has lace like structure. Silver covers a relatively large part of the sample surface in comparison with annealed films. The annealed samples have close values of silver areal density and nanostructure sizes, but the shape of the nanoislands depends on annealing conditions. The sample #2

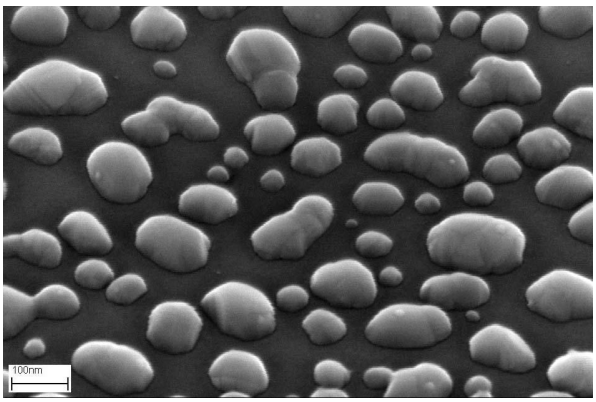


Figure 4. Tilted SEM image of the sample #2.

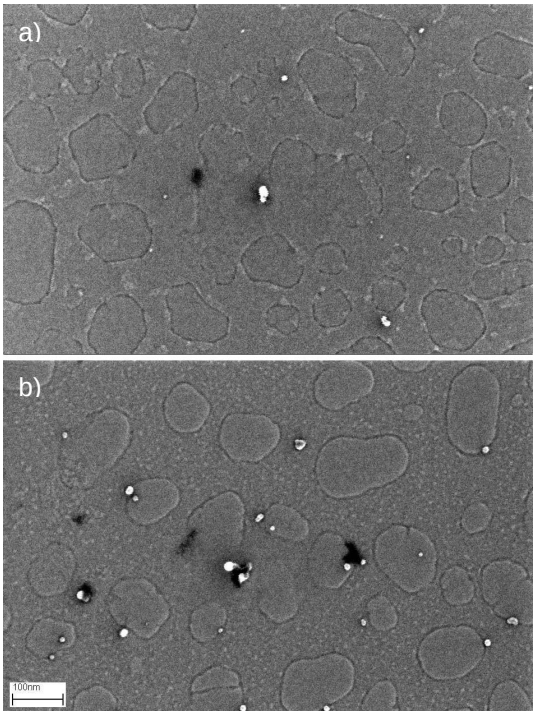


Figure 5. Plan SEM images of the sample #2 after HNO_3 (a) and BHF/HNO_3 (b) treatments, respectively.

demonstrates the most irregular islands with straight flats on some of them. The sample #3 has roundish nanostructures with large shape deviation and the sample #1 shows an intermediate picture between the previous cases. The tilted SEM image of the sample #2 is shown in Fig. 4. The silver islands have the shape of a distorted and bended ellipsoid with a flat bottom. The height of all annealed nanostructures is around 30 nm.

To investigate the modification of the SiO_2 layer below the silver nanostructures after annealing we selectively removed Ag by diluted nitric acid. The acid does not react with Si and stoichiometric SiO_2. The etched samples were studied by SEM and spectroscopic ellepsometry. SEM investigation of samples #1, #3 and #4 did not reveal anything on the sample surfaces. However, SEM images of HNO_3 and BHF/HNO_3 processed chips from the sample #2 demonstrate surface modification (Fig. 5a and Fig. 5b). The SEM plan view taken in the "in lens" mode shows dark contours of nanostructures on the lighter background. In the "in lens" tilted image and plan view taken in the "secondary electrons" mode, the mentioned contours were not observed. The surface of the sample #2 was also scanned by an atomic force microscope and contours were not found. "In lens" mode provides better resolution, but it is more sensitive to electrical charge on the sample surface than "secondary electrons" mode. Therefore, we can conclude that the black contours in Fig. 5 coincide with electrical charge variation, which in turn appears due to changing of local chemical

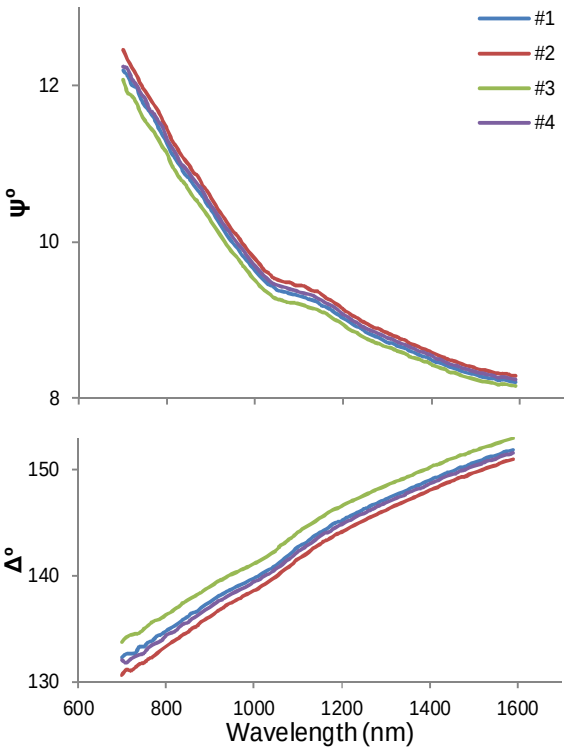


Figure 6. ψ, Δ spectra at 70° after HNO_3 treatment.

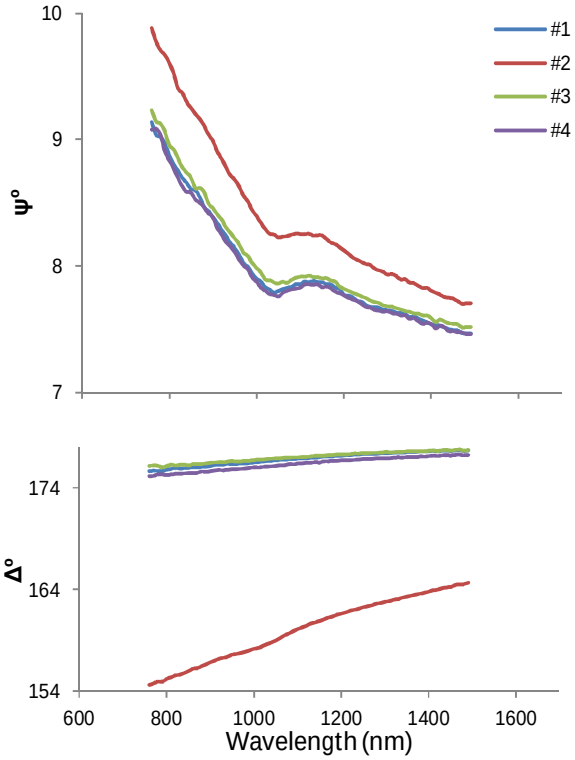


Figure 7. ψ, Δ spectra at 70° after BHF/HNO3 treatment.

TABLE I. SAMPLE DETAILS

Sample	SiO2 sublayer details after HNO3 processing					Original peak, nm	BHF peak, nm	Blueshift, nm
	Thickness, nm	*h1, nm*	*h1 composition*	*h2, nm*	*Thickness loss, nm*			
#1	18.8	12.2	SiO_2	6.6	2.2	439	425	14
#2	19.1	12.6	2% of Si in SiO_2	6.5	1.9	494	443	51
#3	17.6	10.5	SiO_2	7.1	3.4	430	411	19
#4	18.6	11.4	SiO_2	7.2	2.4	-	-	-

composition. Contours in Fig. 5b are more contrast and smoother than in Fig. 5a. Furthermore, there are bright spots in the field of Fig. 5b, which can correspond to the pinholes in the silicon oxide layer.

The RHEED showed relatively sharp, continuous Laue circles in addition to amorphous background patterns for the sample #3. Therefore, this sample contains separate crystalline particles, but their orientation varies from island to island [11]. For other samples, the intensity and sharpness of the diffraction patterns were weaker and decreased in the following order: sample #1, as-deposited sample, sample #2. In other words, the sample #2 contains nanoislands with the most disordered crystalline structure.

B. Properties of oxide sublayer

Spectroscopic ellipsometry is based on measurement of ellipsometric angles ψ, Δ for different wavelengths. The sample is described by a simplified model from several optical layers and ψ, Δ are calculated for the model. After that the matching between measured and calculated ψ, Δ is done for different parameters of the optical layers. Unfortunately, this approach is valid only for systems described by Fresnel equations. Silver nanostructures cause light scattering requiring application of Mie theory [12] and cannot be simulated by Fresnel equations. However, samples with removed silver, i.e., a Si substrate with residual SiO_2 layer can be analyzed by spectroscopic ellipsometry.

The obtained spectra of ψ, Δ after HNO_3 and BHF/HNO_3 treatments are given in Fig. 6 and Fig. 7, respectively. The samples after HNO_3 processing demonstrate small difference in ψ, Δ spectra (Fig. 6). However, BHF/HNO_3 processing results in a big difference between spectrum of the sample #2 and other spectra (Fig. 7). The reconstruction of the sample layers after HNO_3 and BHF/HNO_3 treatments was done using the optical models shown in Fig. 8a and Fig. 8b, respectively. Before this, EDS analyses were performed to ascertain the presence of silver in the etched samples. Traces of Ag were found both after HNO_3, and after BHF/HNO_3 processing. Therefore, the former SiO_2 layer is enriched by

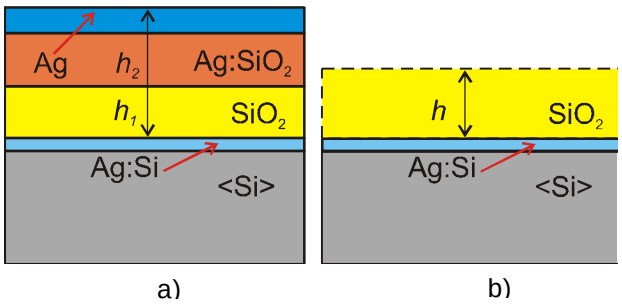


Figure 8. Optical models of oxide sublayer after HNO_3 (a) and BHF/HNO_3 (b) treatments, respectively.

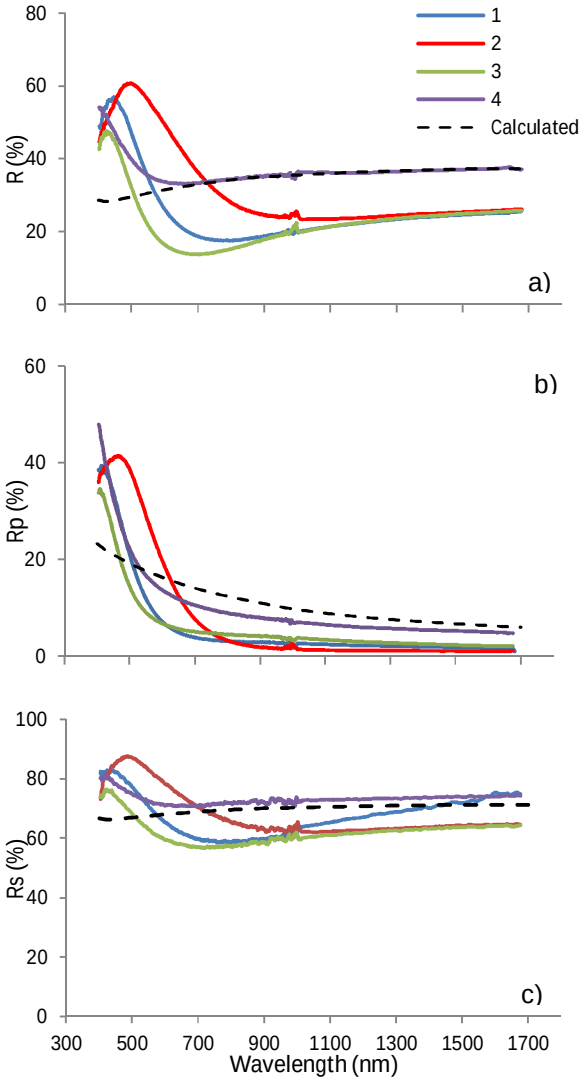


Figure 9. Reflection spectra at normal (a) and inclined (70°) light incidence for *p*-(b) and *s*- polarization (c). Dashed lines show calculated spectra.

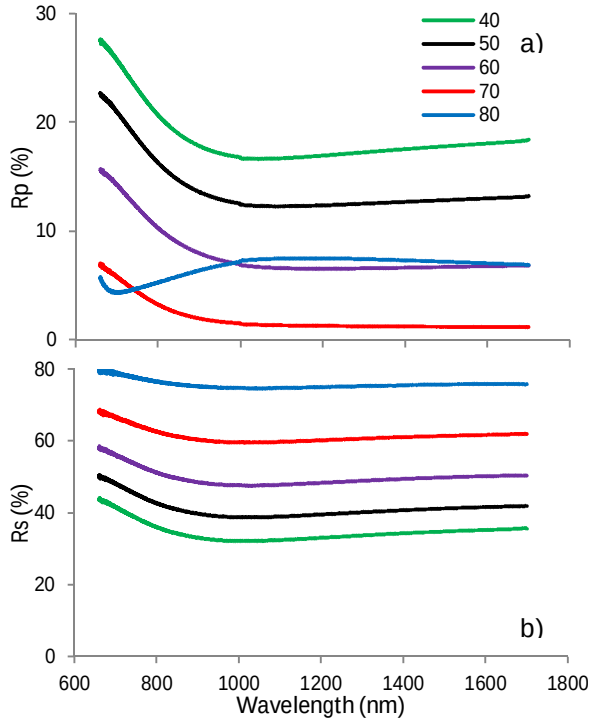


Figure 10. Reflection spectra at different angles of incidence for *p*-(a) and *s*- polarization (b).

Ag and consists of pure SiO_2 (thickness h_1) and composite Ag-SiO_2 (thickness h_2) sublayers. Additionally, a composite layer (35% of Ag in Si) with a thickness of 0.35 nm is required between the substrate and SiO_2 layer to provide the best matching (Fig. 8). The Si and SiO_2 layers with silver inclusions (Ag-Si and Ag-SiO_2) were described with the help of effective medium approximation (Bruggeman model).

The results obtained after HNO_3 processing are given in Table I. For all samples the Ag-rich layer has the same composition (11% of Ag in SiO_2). Thicknesses of the pure SiO_2 layer h_1 and Ag-SiO_2 layer h_2 are changed from sample to sample. Due to this the total thickness of oxide sublayer after HNO_3 processing is varied, but it is always less than SiO_2 thickness (21nm) before Ag deposition. The highest thickness loss was observed in the sample #3 (Table I). The sample #2 differs from others by the presence of Si-rich oxide (2% of Si in SiO_2) instead of stoichiometric SiO_2.

After BHF/HNO_3 processing the SiO_2 layer in the samples #1, #3, #4 was removed and the samples turned into bare Si substrates with thin surface layers. The composition of these layers cannot be found by means of ellipsometry [13]. The sample #2 has a residual SiO_2 layer with a thickness of 10.0 nm and an Ag-Si layer (19% of Ag in Si) at the interface with a thickness of 0.15 nm.

C. Optical properties

It has been already mentioned that colour variation of the samples could be explained by their reflection spectra, which are connected with plasmonic properties of the nanostructures. Fig. 9a demonstrates reflection spectra of the annealed and as-deposited samples at normal light incidence. In Fig. 9b and Fig. 9c, the same spectra are given at inclined light incidence (70°) and for *p*- and *s*- polarization, respectively. According to surface colour, the sample #2 has the main peak at the longest wavelength of 497 nm, the sample #3 at the shortest wavelength of 425 nm and the sample #1 at the intermediate wavelength of 448 nm for normal light incident (Fig. 9a). The as-deposited sample #4 has no reflection peaks in the range of the measurements, but it has trough at the wavelength of 654 nm. On the other hand, the sample #2 has no troughs at all and the samples #3 and #1 have troughs at 694 nm and 767 nm, respectively.

For *p*-polarized light strong reflection is observed only in the visible range (below 800 nm). IR reflectance falls down to 2% for all annealed samples and to 5% for the as-deposited sample. Peaks of reflection for *p*-polarization shift to shorter wavelength and for the sample #2 the peak is observed at 470 nm. For *s*-polarized light spectra of the annealed samples coincide with each other in the IR range (above 1200 nm), which justifies the suggestion concerning identical silver areal density. Blueshift of the peak positions between Fig. 9a and Fig. 9c is equal 12 nm for samples #2,#3 and 4 nm for the sample #1, respectively.

Angle dependence of reflection was studied in near IR range. Fig. 10 demonstrates the reflectance of the sample #2 (the behavior of other samples is similar) for both polarizations in the range of 650–1700 nm. Reflectance of *p*-polarized light falls down with increasing the incident angle

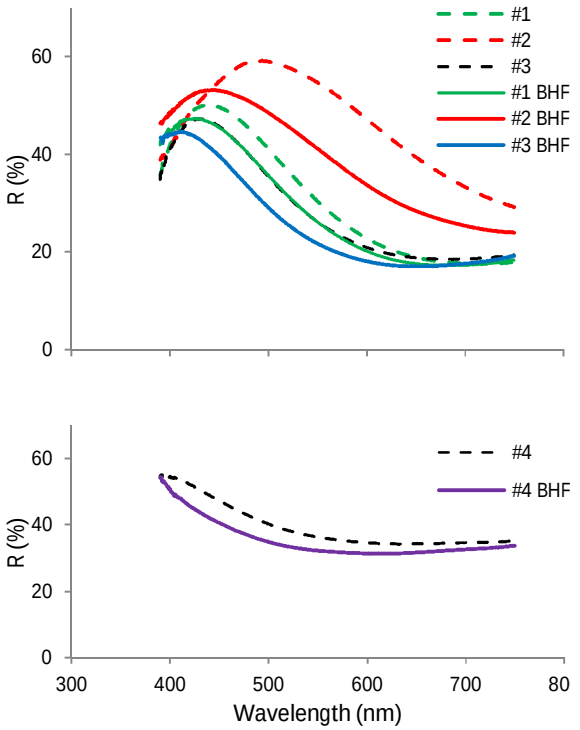


Figure 11. Reflection spectra at normal light incidence before and after BHF treatment.

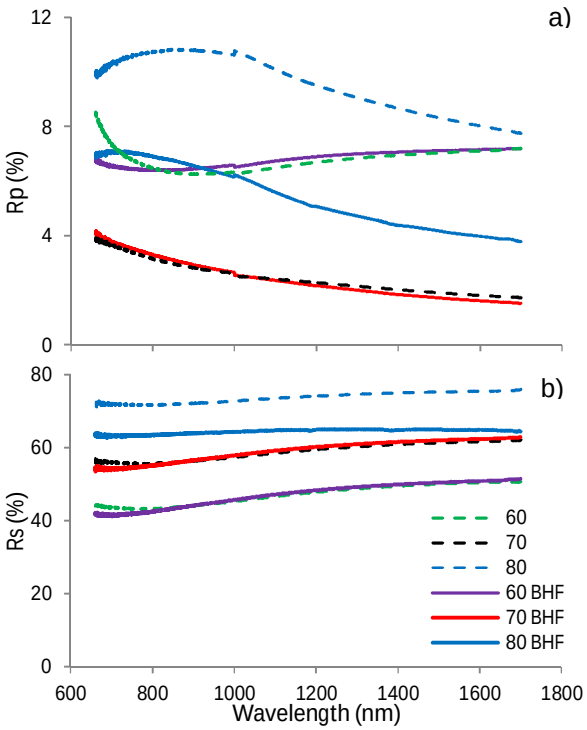


Figure 12. Reflection spectra at different angles of incidence before and after BHF treatment for *p*-polarisaion (a) and *s*-polarization (b).

and reaches its minimum at 70º. After that the wavelength behavior of reflection is changed and the spectrum at 80º looks like a mirror reflection of the 60º spectrum. At the same time, trough positions are redshifted with increasing incident angle. For *s*-polarization the spectrum shape and trough position (1050 nm) are independent from the angle of incidence and reflectance intensity growths with increasing the incident angle.

In IR range scattering is negligible and sample reflection can be described by Fresnel equations. The proposed model consists of a 21 nm thick oxide layer and a Bruggeman Ag-air layer. Reflectance spectra of the sample #4 were used for matching with the optical model, because its silver layer is closest to a continuous film. It was found that the sample #4 can be approximated by a 37 nm thick Ag-air layer (31.5% of Ag). Dashed lines in Fig. 9 show spectra calculated with the help of the obtained model.

For one set of samples (#1–#4) BHF/HNO_3 processing was stopped after BHF etching. After that the SEM investigation did not show any difference between BHF processed and just annealed samples. However, reflectance spectra of all samples were modified in a similar way (Fig.11). After BHF processing the spectrum peaks were shifted to shorter wavelengths and their intensity decreased (Table I).

Reflectance in IR range is not sensitive to BHF processing, excluding the angle of incidence 80º and *p*-polarization (Fig. 12). The spectrum for this angle is shifted down (reflectance decreased) and preserves invariable shape.

D. Simulations

The purpose of simulations in this work is to find out the effect of different heat transfer modes on sample heating in the diffusion furnace (Fig. 1a). 3D simulations of the annealing process were done with the help of software COMSOL Multiphysics 3.5a. Gas flow in the furnace is non-isothermal, which assumes the coupling of fluid dynamics and heat transfer equations in the whole volume of the 96 cm long furnace. Preliminary simulations demonstrate that a converging solution can be obtained for mesh element size less than 5 mm near the surface of the heated wafer (it is the most problematic place for modeling). In this case, the required numbers of mesh elements and degrees of freedom are 70000 and 455000, respectively. The corresponding solution time and memory use for this model are tens of hours and 15 Gb, respectively. However, finding suitable mesh parameters and proper stabilization techniques requires multiple attempts, which leads to high computational load and makes this approach unpractical.

Taking the above mentioned into consideration, the simulations were done in two phases. Firstly, the temperature and velocity fields inside the empty furnace were found. For this purpose two transient models were used in coupled mode: a general heat transfer model and a weakly compressible Navier–Stokes model for non-isothermal flow. The first one calculates gas temperature distribution in the furnace volume due to thermal conduction and convection. Boundary conditions are a fixed temperature of 400 ºC for quartz tube walls and room temperature for input gas. At the gas outlet from the furnace heat exchange was provided by

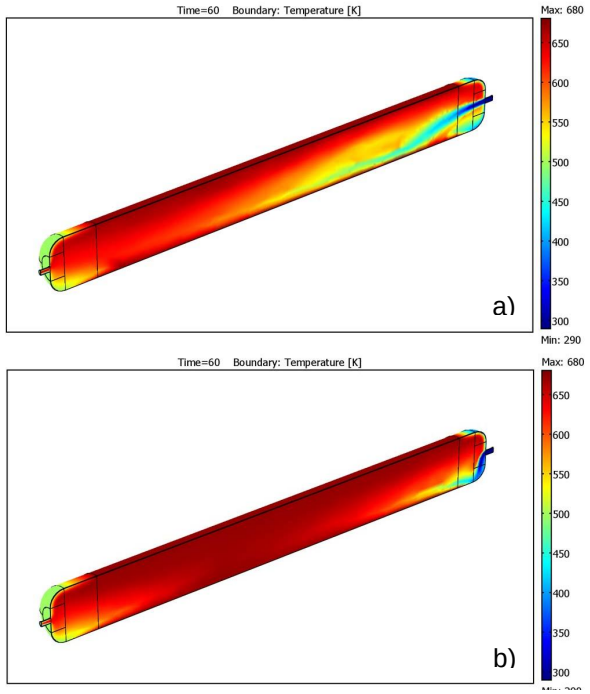


Figure 13. Temperature fields of the diffusion furnace for high (a) and low (b) nitrogen flows. The gas inlet is on the right.

convective flux. The second model calculates gas velocity distribution in the furnace volume caused by inlet pressure and non-uniform temperature. Boundary conditions are laminar inlet flow and atmospheric pressure without viscous stress at the outlet. Appeared gravitational force due to gas density variation was taken into account as the vertical volume force.

At the second phase, the obtained temperature and velocity of gas were used as inlet boundary conditions for the simulation of silicon wafer heating in the hot cylindrical tube. The rest of the boundary conditions were the same as at the first phase. All heat transfer modes, including sample, tube and gas thermal conduction, convection in nitrogen and surface-to-surface radiation were taken into account.

Fig. 13 demonstrates the simulation results obtained at the first phase. Temperature distributions in the diffusion furnace were calculated for the quartz tube with a temperature of 400 ºC and for gas flows 8.3×10^{-5} standard m^3/s (Fig. 13a) and 1.0×10^{-5} standard m^3/s (Fig. 13b), respectively. The large flow of cold gas creates non-uniform temperature distribution inside the tube and gas temperature in the middle of the furnace (below the sample holder) can be 150 ºC lower than the tube temperature (Fig. 13a). At the same place gas velocity reaches a maximum value of 0.18 m/s. At the small flow of nitrogen (Fig. 13b), temperature variation and gas velocity in the centre of the furnace do not exceed 15 ºC and 0.03 m/s, respectively.

Temperature and velocity fields near the wafer are illustrated in Fig. 14a and Fig. 14b, respectively. They are obtained at the second phase of simulations (temperature and

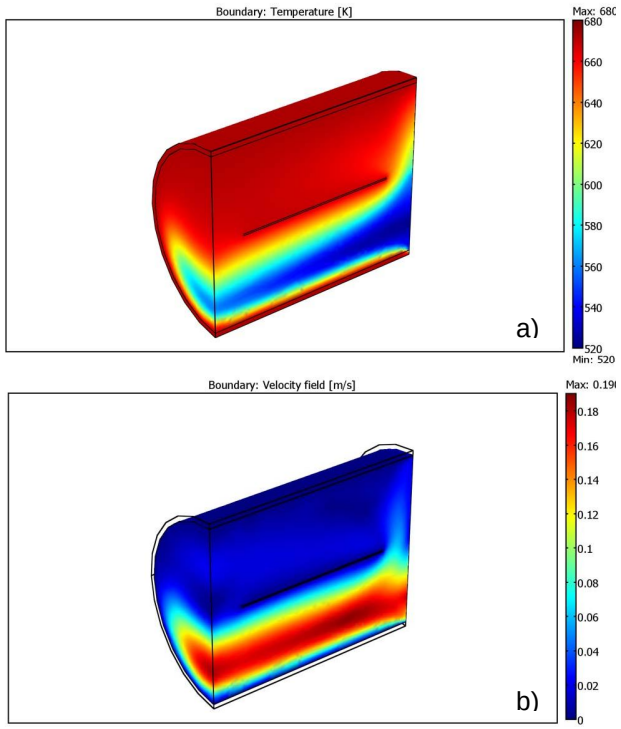


Figure 14. Temperature (a) and velocity (b) fields near the wafer for high nitrogen flow and ε=1. Gas moves from right to left.

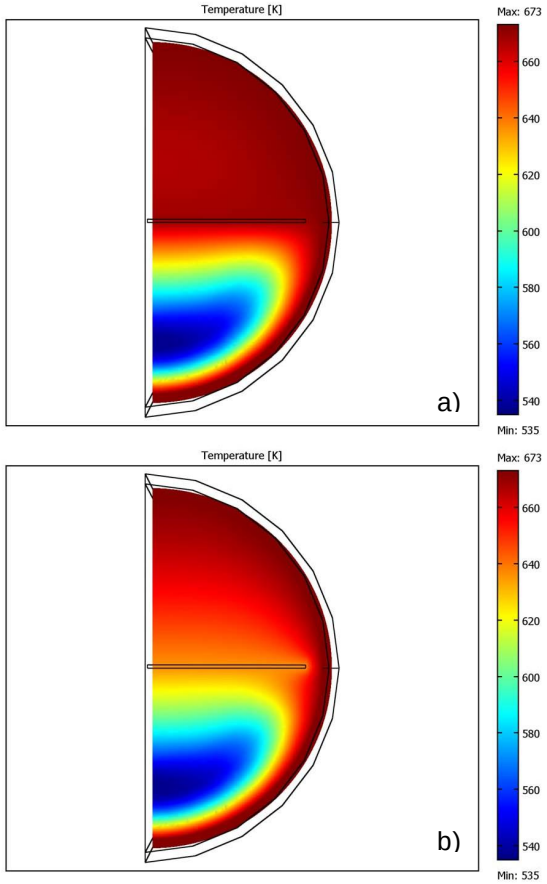


Figure 15. Vertical cross sections of the temperature field at high nitroghen flow in the centre of the furnace for ε=1 (a)and ε=0 (b).

velocity of the gas at the entrance are taken from Fig. 13a) for nitrogen flow 8.3×10^{-5} standard m^3/s and sample emissivity ε=1. The internal furnace volume is divided by the wafer holder in two parts – the upper one with high temperature and low velocity and the lower one with low temperature and high velocity. In the upper volume gas has a temperature of 398 ºC and slowly moves with a velocity of 0.02 m/s. In the lower volume high temperature and velocity gradient exist. However, the wafer temperature variation does not exceed 1ºC due to high thermal conductivity of silicon. In the present experiment, the wafer temperature depends on tube temperature, nitrogen flow and wafer emissivity ε. Cross sections of the temperature fields in the centre of the furnace for ε=1 and ε=0 are given in Fig. 15a and Fig. 15b, respectively. The temperature of the heat absorbing sample (ε=1) is 35 ºC higher than the temperature of the reflective sample (ε=0). As a consequence, the temperature distribution in the upper volume is more uniform for ε=1.

V. DISCUSSION

Annealing of thin silver films is complicated due to three circumstances. Firstly, silver films and nanostructures are melted at low temperatures [5, 14, 15]. In our previous study

[7], it was shown that this melting point is close to 250 ºC. However, this transformation happens only once and the second heating of the sample does not change morphology of the silver nanostructures. Secondly, liquid silver has a tendency to form spherical shapes of nanoislands due to low cohesive forces to SiO_2 surface. Thirdly, silver is the best plasmonic material [9] and the silver nanostructures appeared after breaking apart the continuous film, modify optical properties of the sample surface [10].

The first sign of not identical annealing conditions in the studied furnaces is different sample colours (Fig. 2) and the corresponding changing of reflection spectra after annealing (Fig. 9). The reflection spectra demonstrate strong plasmon properties of the silver nanostructures formed after silver film annealing. The troughs in the range 690–1050 nm correspond to dipole plasmon resonance and peaks at 410–500 nm correspond to quadrupole resonance [10]. The non-annealed sample #4 possesses only very weak dipolar plasmon resonance (see spectrum of as-deposited sample).

The second consequence of not identical annealing is variation in chemical composition and thickness of the oxide sublayer below the silver nanostructures from sample to sample. The resulting thicknesses of the SiO_2 and Ag-SiO_2 layers (Table I) obtained after HNO_3 processing are defined by concentration and distribution of silver in the oxide matrix (Fig. 8). Nitric acid cannot remove silver from SiO_2, if silver concentration is below the corresponding threshold (11% in our samples). Due to this, the Ag-SiO_2 layer left after etching has a silver concentration below 11% (Fig. 8). Therefore, the uppermost Ag-rich SiO_2 (more than 11% of Ag) is removed and thickness loss is higher for samples with higher Ag concentration. The sample #3 has maximal thickness loss and contains maximum amount of silver in oxide.

The sample #2 has minimal thickness loss and contains 2% of excess Si in the lower part of the oxide sublayer (Table I). On the other hand, the sample #2 demonstrates black contours after HNO_3 processing (Fig. 5). One might suppose that excess silicon may be concentrated in these contours, corresponding to removed Ag islands. Electrical field of plasmon oscillations is strongest along the contact line between silver and oxide. Therefore, Si enrichment may be connected with light stimulated diffusion around contact line. Furthermore, silicon can diffuse through deposited silver and can be oxidized on top of it [16]. In our case it means that Si can diffuse through the interface Ag-Si layer (Fig. 8) and is oxidized on top of it. Both light stimulated diffusion and interface stimulated diffusion can lead to compensation of thickness loss.

Section III mentioned that *p*-polarized light is reflected in different way for small (less than 70º) and large (more than 70 º) angles of incidence. We believe that it can be related to Brewster's angle of silicon (74º at 1200 nm) and there are two reasons for this. Firstly, *p*-polarized light is not reflected, but only refracted at Brewster's angle. This was observed in our measurements at 70º (Fig. 9b). Only in this configuration the right position of quadrupole resonance can be visible, because the scattering from silver nanostructures is not disturbed by the reflection from the substrate. Secondly,

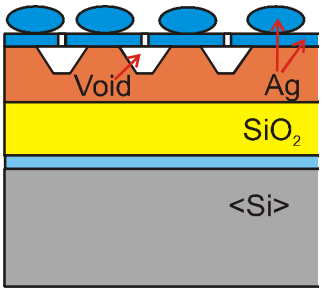


Figure 16. Void layer formation after BHF processing.

there is a jump in the reflection phase at Brewster's angle, i.e., for smaller angles of incidence original and reflected lights have a phase shift of 180º, but for larger angles the phase shift is 0º. It is illustrated by distinguished behavior of the reflection spectrum for 80º in Fig. 10a. Therefore, reflection from Si/SiO_2 interface plays a crucial role in modification of the observed spectra and redshift of troughs for *p*-polarized light with increasing of the incident angle may be explained by destructive interference (Fig. 10a).

To some extent plasmon properties can be estimated by the difference between calculated Fresnel equations and measured spectrum, i.e., the larger difference, the stronger plasmon resonance. Based on this criteria, the strongest plasmon resonances are observed in the samples #1 and #3 (Fig. 9).

In Section III, we have shown that all annealed samples have similar values of silver areal density and nanostructure sizes. Therefore, relatively large redshift of peaks and troughs in Fig. 9 cannot be only explained by the changing of island geometry. Due to the identity of the studied samples, the spectrum variations can be also connected with material modification, e.g., changing of Ag or SiO_2 dielectric functions. Spectral peak and trough broadening (sample #2 has the broadest peak) tells about an increase of the imaginary part of Ag dielectric function. Peak shift is connected with changing of a real part of the dielectric function, i.e., refractive index [9, 12]. It is clearly demonstrated by blueshift of plasmon resonances in the experiment with BHF etching (Fig. 11 and Table I). Due to pinholes in silver residues between the nanostructures (Fig. 16), SiO_2 is partially etched and voids are formed below the Ag nanostructures. It results in decrease of the effective refractive index n of the substrate (n_{air}=1, n_{SiO2}=1.45) and a corresponding shift of the plasmon resonance. Additionally, the same voids can increase scattering of the light travelling in the SiO_2 layer, which leads to uniform decrease in intensity of the light reflected at 80º (Fig. 12).

Typically, annealing is used to improve and restore crystalline structure. However, there are reports about increased defect concentration in melted silver samples [17]. Our RHEED observations also showed that the crystalline structure of the annealed sample #2 is worse than the structure of the as-deposited one. Taking into account the broadest reflection peak and the absence of a dipolar trough in the sample #2, we can conclude that this sample has the

highest disorder of crystalline structure among the studied samples.

In the diffusion furnace (Fig. 1a) the target temperature 400 ºC was supported on the external side of the quartz tube. In the fast ramping furnace (Fig. 1b) the target temperature 400 ºC was supported inside the furnace, at 1cm above the bottom of the quartz tube. According to Fig. 14a, the measured temperature in this point can be 150 ºC lower than the tube temperature, i.e., in our experiment the tube temperature of the fast ramping furnace could be close to 550 ºC. Nitrogen flow 8.3×10^{-5} standard m^3/s is very low for the fast ramping furnace (Fig. 1b) and provides laminar gas flow inside the tube. In the case of the diffusion furnace (Fig. 1a), the same nitrogen flow is too high and provides turbulent gas flow in the lower part of the tube (Fig. 13a). Higher temperature of the absorber shield ($\varepsilon\sim1$) around the quartz tube makes thermal radiation in the fast ramping furnace much higher than in the diffusion one.

In the case of a thin silver layer on silicon, most of radiation energy is absorbed in the silver and during heating up in the laminar gas flow (the fast ramping furnace) the silver temperature is higher than the temperature of the substrate. In turbulent gas flow (the diffusion furnace), intensive heat exchange between the silver and nitrogen prevents overheating of the silver nanostructures.

After melting silver starts to form droplets due to surface tension forces and decreases silver areal density. However, absorbed thermal radiation is proportional to silver areal density or absorbing cross-section. Thus, geometry change decreases radiative heat transfer to the silver. The cold substrate cools down silver nanostructures and causes their rapid solidification. The quenching happens without proper crystallization and silver solidifies in amorphous phase (sample #2).

In the case of low radiative heat transfer (samples #1, #3), melting happens at higher substrate temperature and without silver overheating. Depending on conductive and radiative heat fluxes the melted silver is cooled with a much lower rate and solidifies in polycrystalline phase. In our study, the sample #3 has the best crystalline structure due to lower cooling rate between two hot plates in vacuum. One of the reasons for quenching in this case is the reduction of the surface energy [18]. Another reason is the heating of silver nanostructures by conductive flux through thermal contact with substrate. Silver melting acquires additional heat flux from the substrate to the nanostructure. This heat flux increases the temperature drop on the interface between the substrate and the silver droplet, which in turn leads to decreasing of the silver temperature and quenching.

VI. Conclusion and Future Work

Annealing of identical samples at identical times and temperatures, but in different furnaces leads to different results. We have demonstrated that optical properties and morphology of silver nanostructures produced by annealing of thin film are very sensitive to the heat delivering method. Relative strength of the heat transfer modes affects the wavelength of plasmon resonance, nanostructure geometry and chemical composition of the oxide sublayer. The effect of furnace operational parameters (gas flow, sample and thermocouple position, sample and environment emissivity) on annealing results has been confirmed. Radiation heating of silver can be very strong and provides overheating of film with regards to the substrate. It results in silver melting and droplet formation. The appearing of nanostructures and shrinking of silver areal density lead to a decrease of radiation heating. As a result, melted structures are quenched to solid state with irregular shape and high crystalline disorder. The effect depends on the rate of solidification and explains the variation of annealing results from furnace to furnace.

The results presented here have demonstrated the significance of all furnace operational parameters and can be used for controllable heat processing of different materials. However, the work can be further developed for various furnace designs and thin film materials. Furthermore, accuracy and validity of the process simulations can be improved. Proper understanding of film transformation during annealing opens an effective way for the formation of nanostructures of different shapes, e.g., arrays of spherical nanoislands.

Acknowledgment

This research was undertaken at the Micronova Nanofabrication Centre of Aalto University.

References

[1] V. Ovchinnikov, “Analysis of Furnace Operational Parameters for Controllable Annealing of Thin Films,” Proceedings of ICQNM 2014, ThinkMind Digital Library (ISBN: 978-1-61208-380-3), pp. 32-37.

[2] Handbook of Semiconductor Manufacturing Technology, 2nd edition edited by R. Doering and Y. Nishi, CRC Press, 2007, 1720p.

[3] S. Franssila, “Introduction to Microfabrication,” 2nd edition, Wiley, 2010, 534p.

[4] V. Ovchinnikov, A. Malinin, S. Novikov, and C. Tuovinen, ”Silicon Nanopillars Formed by Reactive Ion Etching Using a Self-Organized Gold Mask,” Physica Scripta, vol.T79, 1999, pp. 263-265.

[5] S. R. Bhattacharyya et al., “Growth and Melting of Silicon Supported Silver Nanocluster Films,” J. Phys. D: Appl. Phys., vol. 42, 2009, pp. 035306-1 - 035306-9.

[6] D. Adams, T. L. Alford, and J. W. Mayer, “Silver Metallization: Stability and Reliability,” Springer, 2008, 123p.

[7] V. Ovchinnikov, “Effect of Thermal Radiation during Annealing on Self-organization of Thin Silver Films,” Proceedings of ICQNM 2013, ThinkMind Digital Library (ISBN: 978-1-61208-303-2), pp. 1-6.

[8] D. Guo, S. Ikeda, K. Saiki, H. Miyazoe, and K. Terashima, “Effect of annealing on the mobility and morphology of thermally activated pentacene thin film transistors,” J. Appl. Phys., vol. 99, 2006, pp. 094502-1 – 094502-7.

[9] M. A. Garcia, “Surface Plasmons in Metallic Nanoparticles: Fundamentals and Applications,” J. Phys. D: Appl. Phys., vol. 44, 2011, pp. 283001-1 - 283001-20.

[10] V. Ovchinnikov and A. Shevchenko, “Self-Organization-Based Fabrication of Stable Noble-Metal Nanostructures on Large-Area Dielectric Substrates,” Journal of Chemistry, vol. 2013, 2013, Article ID 158431, pp. 1 - 10., http://dx.doi.org/10.1155/2013/158431.

[11] A. Ichimiya and P. I. Cohen, “Reflection High-Energy Electron Diffraction,” Cambridge University Press, 2004, 353p.

[12] E. C. Le Ru and P. G. Etchegoin, “Principles of Surface-Enhanced Raman Spectroscopy and Related Plasmonic Effects,” Elsevier, 2008, 688 p.

[13] H. G. Tompkins, “A User's Guide to Ellipsometry,” Academic Press, 1993, 260p.

[14] O. A. Yeshchenko, I. M. Dmitruk, A. A. Alexeenko, and A. V. Kotko, “Surface Plasmon as a Probe for Melting of Silver Nanoparticles,” Nanotechnology, vol. 21, 2010, pp. 045203-1 - 045203-6.

[15] M. Khan, S. Kumar, M. Ahamed, S. Alrokayan, and M. Salhi, “Structural and Thermal Studies of Silver Nanoparticles and Electrical Transport Study of Their Thin Films,” Nanoscale Research Letters, vol. 6, 2011, pp. 434-1 - 434-8.

[16] A. Hiraki and E. Lugujjo, “Low-Temperature Migration of Silicon in Metal Films on Silicon Substrates Studied by Backscattering Techniques,” J. Vac.Sci.Technol., vol. 9, 1972, pp.155-158.

[17] S. A. Little, T. Begou, R. E. Collins, and S. Marsillac, “Optical Detection of Melting Point Depression for Silver Nanoparticles via in situ Real Time Spectroscopic Ellipsometry,” Appl. Phys. Lett., vol. 100, 2012, pp. 051107-1 -1 051107-4.

[18] E. P. Kitsyuk, D. G. Gromov, E. N. Redichev, and I. V. Sagunova, “Specifics of LowTemperature Melting and Disintegration into Drops of Silver Thin Films,” Protection of Metals and Physical Chemistry of Surfaces, vol. 48, 2012, pp. 304–309.

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