International Journal on Advances in Systems and Measurements

2014 vol. 7 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 7, Number 1 & 2, 2014

António Dourado, University of Coimbra, Portugal
Daniela Dragomirescu, LAAS-CNRS / University of Toulouse, France
Matthew Dunlop, Virginia Tech, USA
Mohamed Eltoweissy, Pacific Northwest National Laboratory / Virginia Tech, USA
Paulo Felisberto, LARSyS, University of Algarve, Portugal
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
João V. Gomes, University of Beira Interior, 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
Daisuke Mashima, Georgia Institute of Technology, USA
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
Arvind K. Srivastav, NanoSonix Inc., USA
Grigore Stamatescu, University Politehnica of Bucharest, Romania
Raluca-Ioana Stefan-van Staden, National Institute of Research for Electrochemistry and Condensed Matter, Romania
Pavel Šteffan, Brno University of Technology, Czech Republic
Monika Steinberg, University of Applied Sciences and Arts Hanover, Germany
Chelakara S. Subramanian, Florida Institute of Technology, USA
Sofiene Tahar, Concordia University, Canada
Jaw-Luen Tang, National Chung Cheng University, Taiwan
Muhammad Tariq, Waseda University, Japan
Roald Taymanov, D.I.Mendeleyev Institute for Metrology, St.Petersburg, Russia
Francesco Tiezzi, IMT Institute for Advanced Studies Lucca, Italy
Theo Tryfonas, University of Bristol, UK
Wilfried Uhring, University of Strasbourg // CNRS, France
Guillaume Valadon, French Network and Information and Security Agency, France
Eloisa Vargiu, Barcelona Digital - Barcelona, Spain
Miroslav Velev, Aries Design Automation, USA
Dario Vieira, EFREI, France
Stephen White, University of Huddersfield, UK
Shengnan Wu, American Airlines, USA
Xiaodong Xu, Beijing University of Posts & Telecommunications, China
Ravi M. Yadahalli, PES Institute of Technology and Management, India
Yanyan (Linda) Yang, University of Portsmouth, UK
Shigeru Yamashita, Ritsumeikan University, Japan
Patrick Meumeu Yomsi, INRIA Nancy-Grand Est, France
Alberto Yúfera, Centro Nacional de Microelectronica (CNM-CSIC) - Sevilla, Spain
Sergey Y. Yurish, IFSA, Spain
David Zammit-Mangion, University of Malta, Malta
Guigen Zhang, Clemson University, USA
Weiping Zhang, Shanghai Jiao Tong University, P. R. China
J Zheng-Johansson, Institute of Fundamental Physic Research, Sweden

International Journal on Advances in Systems and Measurements
Volume 7, Numbers 1 & 2, 2014

CONTENTS

Generic Frameworks for a Matrix of RFID Readers Based Interactions

Nicolas Géraud
Dasein Interactions,
4 pl Jean Achard, 38000 GRENOBLE
Email: nicolas.geraud@dasein-interactions.fr
Maxime Louvel
Univ. Grenoble Alpes, F-38000 Grenoble, France
CEA, LETI, MINATEC Campus, 17 rue des Martyrs, 38000 Grenoble, France
Email: maxime.louvel@cea.fr
François Pacull
Univ. Grenoble Alpes, F-38000 Grenoble, France
CEA, LETI, MINATEC Campus, 17 rue des Martyrs, 38000 Grenoble, France
Email: francois.pacull@cea.fr

***Abstract*—The paper presents first a framework to develop applications on a very innovative hardware associating hundreds of RFID readers and a high resolution display within a table. The framework is built on top of a rule-based coordination middleware that provides mechanisms to handle combinations of events, generated by the RFID readers. This framework offers the basic blocks to fully support the hardware. The paper demonstrates the interest and the possibilities of the framework through simple examples. In a second part, the flexibility of the approach is illustrated by combining this *"interface"* framework with *"rendering"* frameworks built on top of the same coordination middleware. As a result, we exemplify the re-usability approach through two scenarios (case-studies) belonging to very different application domains: help to decision making and urban mediation.**

***Keywords*-Coordination Middleware; RFID; Data aggregation.**

I. Introduction

Sensor networks are continuously growing and bringing new designs and usages. The increasing number of devices implied at the same time and the increasingly complex interactions required by the usages do not ease the task of the application programmers. There is a need for a middleware layer, offering as basic blocks high level mechanisms, in order to move most of the complexity from the application to the middleware. This paper illustrates this with an innovative smart table hosting a high resolution display and a matrix of several hundreds of RFID readers. The usage of this table is multiple when it is question of interaction, mediation and collaboration between several users. A first experience has been described in [1]. This paper goes further and shows the re-usability and extensibility of software components built with the proposed middleware. A completely different application domain is considered as a second case study.

The paper is organised as follows. Section II describes the hardware embedded in the table. The table allows to detect the identity and the position of RFID tagged objects put on the table, and to display arbitrary pictures on the HD screen. Section III presents the rule based middleware and the frameworks built on top, which offers to the application designer the basic interactions involving objects equipped with RFID tags and graphical engines managing 2D and 3D graphical objects displayed as feedback to the users. Then, Section IV puts these frameworks in context to show how interactions can be build. Section V then offers a discussion on the proposed software environment and puts it in perspective of related works. Section VI illustrates two complex applications to help decision making and urban mediation. Finally, Section VII concludes the paper.

II. Hardware

To illustrate the capability of our middleware to manage complex events detection, this paper describes our experiment with an original hardware. This hardware combines within a table, a RFID based location system and a HD screen that is used as a dynamic tablecloth.

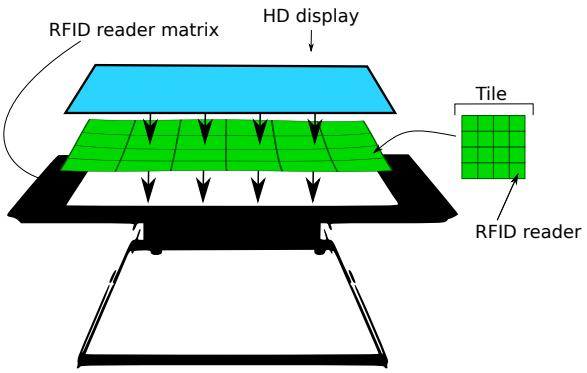


Fig. 1. Description of the table

Fig. 1 describes the table, composed of two layers. The first layer is a 42" screen able to display with a HD resolution of 1080p. This screen is seen as a classical LCD display and can thus be connected to a computer or smaller footprint board (e.g., a raspberry pi board). Under this display layer, there is a set of RFID readers organised as a matrix of 6 x 4 tiles, with each tile containing itself a matrix of 4 x 4 RFID readers. As

a result there are 24 x 16 (384) RFID readers distributed in the table. The size of a RFID reader antenna is 3.3 x 3.3 cm.

This table works with classical RFID tags that can be attached to any physical object. The raw information received for each RFID reader is the set of detected tags along with the corresponding signal strengths. This information is collected via Ethernet. Each tile has its own IP address and gives information for the 16 RFID readers constituting the tile.

There are two interesting functioning modes of this table. In the first mode (push), the tiles are autonomous and send automatically information each time a RFID is seen. In the second mode (pull), each tile can be interrogated in order to have the information corresponding to the RFID readers it contains.

With this hardware, the applicative fields are quite infinite provided that the middleware offers the required abstraction layer and a powerful mechanism to define the coordination schemes we want to put in place.

III. SOFTWARE

The presented hardware allows a lot of interactions through tangible objects. It needs a high level middleware able to quickly react to the context defined by the set of objects present on the table at the same time. Applications for this hardware typically combine RFID tag location, co-location (several tags), proximity, distance, sequence of tags put on the table. Moreover it is possible to use other interfaces connected to the system (e.g., 3d mouse, cameras). This section firstly introduces the middleware we use. For a more detailed description of this middleware, the reader may refer to [2] or [3], where it has been used in the building automation context. Then, the section presents the frameworks we developed on top to ease the creation of applications involving the table.

A. Coordination Middleware

This middleware, called LINC, is an evolution of earlier middlewares [4], [5]. LINC has been specifically re-designed to tackle lightweight embedded systems. It provides a uniform abstraction layer that eases the integration and coordination of the different components (software and hardware). It relies on the *Associative memory* paradigm implemented in our case as a distributed set of bags containing resources (tuples). Following Linda [6] approach the bags are accessed through three operations:

- `rd()` that takes as parameter a partially instantiated tuple and returns from the bag a fully instantiated tuple whose fields match the input pattern;
- `put()` that takes as parameter a fully instantiated tuple and inserts it in the bag;
- `get()` that takes as parameter a fully instantiated tuple, verifies its presence in the bag and consumes it in an atomic way.

For a matrix of RFID readers like the one in the table, bags `RawInformation` and `Position` may contain raw data such as `(tagid, tileid, readerid)` or more refined data as `(tagid, posX, posY)`. Depending on the usage (calibration or real application) they both have an interest. Once the location is computed according to the raw data, metadata may be considered from the association between `(physicalTagId, tagId)` or `(tagId, objectId)`.

The `put()` operation can insert tuples into bags configuring the readers operating mode or other configuration parameters. Finally, some bags may be used to control the videos that are displayed on the table screen.

In addition, bags can be grouped inside objects for identification purpose. For instance, the object modelling the RFID readers will contain all the bags allowing its management.

The operations `rd()`, `get()` and `put()` are used in production rules [7] to express the way these resources are used in the classical *pre-condition* and *performance* phases.

Precondition phase: It relies on a sequence of rd() operations to find and detect the presence of resources in several bags. This can be sensed values, result of service calls or states stored in tuplespaces or databases.

The particularity of the precondition phase is that:

- the result of a `rd()` operation can be used to define some fields of the subsequent `rd()` operation;
- a `rd()` is blocked until a resource corresponding to the pattern is available.

Performance phase: It combines the operations `rd()`, `get()` and `put()` to respectively verify that some resources found in the precondition phase are still present, consume some resources and insert new resources. In this phase, the operations are embedded in *distributed transactions* [8]. This ensures several properties that go beyond traditional production rules. In particular, it ensures that:

- the conditions responsible of firing the rule (precondition) are still valid in the performance phase;
- the different involved bags are effectively all accessible.

These properties are very important since they allow to verify that a set of objects are actually present "at the same time" on the table.

B. Frameworks

In LINC the approach is to define frameworks dedicated to a specific aspect of the underlying hardware or the legacy software component that is encapsulated. The frameworks provide basic blocks (LINC objects) that are then used to define the applications. The more application neutral the objects are, the more reusable they are.

Here, we consider three frameworks. The first framework is responsible for the interaction through the RFID readers embedded in the table. It has been written from scratch since it is related to a very specific hardware. It consists mainly of one LINC object called *RFID*.

The second framework is responsible for managing what is displayed on the screen. This is a standard LINC framework already used in other applications. It is composed of two objects: *`Display`* and *`2D_Engine`*. It is responsible for rendering more or less complex 2D information on a screen such as: videos, static or animated drawing, text, etc.

The third framework encapsulates off-the-self 3D engines able to render scenes containing moving 3D objects and managing textures, points of view, lights, etc. It contains two

objects *3D_OSG* and *3D_Ogre* encapsulating respectively *OpenSceneGraph* [9], [10] and *Ogre* [11], two well known 3D engines built on top of OpenGL [12]. *3D_Ogre* has been developed first, then we did the same for *3D_OSG*, keeping the same bags in order to make them interchangeable.

C. First framework: RFID Table

Object RFID: This object models the RFID readers matrix. It contains the following bags:

- `Position (tagId, posX, posY)`: contains the position of the tag (0,0 defines the top left position);
- `LogicalTag (physicalTagId, tagId)`: stores the association of a physical tagId with a more meaningful logical id, e.g., ("030209348393", "video1");
- `TagStatus(tagId, status)`: contains the status of a tag: `"in"` if detected by a RFID reader or `"out"` if not seen for a given time;
- `Mapping (tagId, objectId)`: keeps the association physical object and RFID tag that is attached to it (e.g., an hourglass used to symbolise a timer);
- `Type (tagId, type)`: maintains association of a tagId with a type of tagged object (e.g., physical object, video, action card, badge);
- `Area (areaId, areaDefinition)`: contains areas on the table defined as a set of points forming a polygon;
- `PositionArea (tagId, areaId)`: contains the tagId contained in a given area.

The detection of the tags placed on the table is done by a driver that handles the events sent by the different RFID readers (used in push mode). This information is decoded and the different bags are filled with the corresponding resources. When a tag is detected, the driver computes its position on the table (X,Y) and adds the resource `(tagId, posX, posY)` in the bag `Position`. A tag is detected by one or several RFID readers of the table. To improve the precision of the detected location we can use the signal strength provided by the readers. Thus, we have higher precision than the size of an RFID reader antenna. Practically, we can consider a step equal to the third of the antenna size (around 1.1 cm).

As a RFID reader continuously sends the tag information and as the information slightly vary, a filtering is applied to avoid inserting new resources when it is not necessary. Hence, a resource is inserted only when a significant change in the location is effective. In addition, the `status` of the tag, `"in"` if the tag is still on the table or `"out"` if it has been removed (i.e., not be seen for a given time), is inserted as a resource `(tagId, status)` in the bag `TagStatus` each time the `status` changes.

The bags `Type`, `Area` or `LogicalTag` are configuration bags and their usage is described here after.

Introduction to rules: The described middleware allows to express with its rule based language actions to be performed (performance phase) when some conditions (precondition phase) are verified. The actions performed are embedded in transactions enclosed in {}. As `rd()` actions may be included in these transactions, it is possible to ensure that resources found in the precondition are still valid in the performance.

```
1 ["RFID","TagStatus"].rd(tagId, "in") &
2 # other preconditions
3 ::
4 {
5  ["RFID","TagStatus"].rd(tagId, "in");
6    # other actions
7 }.
```

Listing 1. Ensure tag is still there at performance phase

Listing 1 presents an example of rule, where the precondition and performance part are respectively before and after the `"::"`. To simplify the example, we only show a single operation in the precondition and performance phase but both may contain several additional tokens.

The first token (line 1) reads in the bag `TagStatus` of the object `RFID` all the tags with status "in". This allows to detect new tags placed on the table and then to manage the corresponding scenario. Line 5 guaranties that the `tagId` is still on the table when the performance phase is executed. Since actions in the performance are embedded in transactions the other actions can only be done if the tag is still there. Note that this approach simplifies a lot the management of events:

- events are detected in preconditions;
- when performances are executed, guarantying that the condition related to the event is still valid only requires to add a `rd()` in the performance part.

Initialisation rules: Listing 2 presents an initialisation rule. No precondition is defined, this rule is always executed and only once at the application launch time.

```
1 ::
2 {
3  ["RFID","LogicalTag"].put("9e7f9cce9","tag_video_table");
4  ["RFID","Area"].put("zoneA","0,0;0,54,12,54;12,66;66,0");
5  ["RFID","Type"].put("t_video_presentation_table","video");
6 }.
```

Listing 2. Initialisation rule

Here we initialise the bags `LogicalTag`, `Area` and `Type`.

In the first bag, we associate the physical tagid `"94e7f89cce9"` to the more user friendly logical tag `"tag_video_table"`. This allows to manipulate in the rules an id that is human readable. In addition, several physical tags can be associated to the same logical tag for backup reason or to offer to several people the possibility to trigger the same action with different objects or cards.

In the second bag, we define a `"zoneA"` as a list of points defining a polygon. This is taken into account by the driver to populate the `PositionArea` bag.

In the third bag, we associate a `type` to a tag. The `type` allows to define a specific context around this tag to verify that it is correctly used. For instance, a tag associated to a voting card cannot be placed everywhere on the table but in a given area. Another usage is to give information to the driver about the sampling frequency for a given tag or if the change in the location is large enough to be reported or not.

Defining action area: To better organise the table, area (i.e., zone of the table) can be used. An area is defined by adding a resource `(areaId, areaDefinition)` in the bag `Area`. The `areaDefinition` is a set of points defining a polygon. When the RFID driver detects a new position for a tag, at the same time it inserts the corresponding resource

in the bag `Position`, it scans all the defined areas and adds the resources `(tagId, areaId)` in the bag `Area`. In the same manner, when the driver inserts a resource `(tagId, "out")` in the bag `Status` it removes all the resources corresponding to the tag in the bag `Area`. This simplifies the application designer's task since she can directly write a rule that starts with a token reading in the bag `Area`.

D. Second framework: 2D Rendering Engine

This framework is a generic 2D rendering engine. Its role is to manage what is displayed on a screen. It includes an object more oriented to video rendering and another that display arbitrary 2D fixed or animated graphical objects. The target, can be, as in our case, the screen included in the table, but also a smart TV, a regular computer screen, a tablet or a smartphone.

Object `Display`: The first object of the framework manages the displays on the screen. It contains the following bags (non exhaustive list):

- `videoPlayer (playerId, videoname, posX, posY, width, height, orientation, soundTrack)`: this bag accepts only the `put()` operations and launches a video player displaying the video corresponding to the filename with the given geometry, with or without sound track;
- `video (videoname, status)`: maintains the status of the video among `started`, `finished`, `paused`;
- `videoPlayerCommand(videoname, command)`: this bag accepts only the `put()` operations and the following commands: `"stop"`, `"pause"`, `"resume"`, `"fs_on"`, `"fs_off"` (fs is for full screen).

A simple usage of this object is described in Listing 3. This initialisation rule starts the video called *video_table* presenting the table on the top left corner of the screen. When the performance is executing, a video player is started and configured to display the video with the resolution (640x480) at position (0,0). The status of the video is set to `"started"`.

```
::
{
 ["Display","video"].put("video_table","started");
 ["Display","videoPlayer"].put("vlc","video_table",
      "0","0","640","480", "True");
}.
```

Listing 3. Start presentation video of the table

To easily support any kind of video player, the framework uses an external Linux process. The role of this process is to display a video according to a media definition file containing the basic information needed to define the layout, the position, the fact that the sound track is on or off. The display driver saves the PID of the process started in order to interact with it independently of the video player used.

Listing 4 shows how to stop a video. The precondition waits that the stop card is placed on the table. It then reads the `videoId` of the started video. The performance actually stops the video just by sending the signal `SIGKILL` to the PID playing the video.

```
["RFID","TagStatus"].rd("tag_stop_video", "in")&
["Display","video"].rd(videoId,"started")
::
{
 ["Display","videoPlayerCommand"].put(videoId, "stop");
}.
```

Listing 4. Stopping a video with a control card

Object `2D engine`: The second object of the framework is a 2D engine. It is in charge of displaying the background of the table. It is also in charge of the displayed animations. The current version relies on a Scalable Vector Graphics [13] (SVG) engine to define 2D animations that will be displayed in a simple web browser that is opened in full screen on the table display.

The 2D engine object contains the following bags (non exhaustive list):

- `Background (imagefile)`: When a resource (i.e., an image file) is inserted it replaces the current background of the table;
- `Media (tagId, filename)`: associates a `tagId` to a `filename`;
- `Sprites (spriteId,x,y, svgfile)`: Allows to display a sprite (SVG image) at the position `x,y` on the table screen;
- `MoveSpriteGrid (spriteid,x,y,duration, nbsteps,renderlist)`: Allows to define an animation for the sprite defined by `spriteid`. The `duration` of the animation using; `nbsteps` steps and using successively the SVG patterns defined in `renderlist`;
- `Visibility (spriteId,percent)`: defines the opacity and the visibility of the sprite.

This object actually contains more bags that allow not only to define a background but also sprites that can be animated on top of this background. All the SVG attributes may be dynamically modified.

The animations are done at the level of the object that returns an html file when invoked through URL. This HTML file is built from static information (HTML [14] and SVG [13] templates) present in the file system and contextual information present in the bags.

In addition, SVG templates are filled by javascripts [15] to bring the dynamic aspects through animated SVG entities. Finally, through SVG and javascripts it is possible to attach URL based interactions to classical events *mouseclick*, *mouseover*, etc. These URL calls either insert or read resources in bags dedicated to user interactions. Resources are put in bags to capture the inputs from the users. Furthermore, resources are regularly read in bags to obtain updated information. The resources are returned to the web browser as json structure [16] easily understandable by javascripts. Thus, with devices allowing user interactions (e.g., tablet and phone) we can go very far in term of user interface.

The main advantage of this approach is that we use the full power of nowadays web technologies without paying an heavy cost at the rendering level. As most of the current equipments are surprisingly able to manage quite complex web pages, this framework can deal with almost all the user life

objects owning a screen. For instance, we have built a user interface including SVG drawing for home automation with a simple kindle paperwhite e-reader. Thus, you can have a tablet-based interface always available in your living room with an autonomy of a month.

E. Third Framework: 3D Rendering

In the same way, we have developed a generic 3D rendering engine that allows to display arbitrary 3D scenes.

Object `3D engine`: This object encapsulates the 3D engine Open Scene Graph [10]. It runs on a multicore laptop and the output is displayed either on an external large screen or a video projector. The role of this object is to display complex interactive 3D scenes. To deal with performance, scalability and high quality user experience the LINC object has been decomposed in two parts. First, we have a set of bags that are used to store the basic information about the manipulated entities.

- `Entity (entity, file)`: contains 3D models of buildings that are the same as the ones used for printing the buildings on the 3D printer. Thus, no extra effort is required for the 3D virtual scene.
- `Light (aspect, r, g, b)`: this bag allows to insert different types of light that are used for rendering the scene.
- `Mode (key, mode)`: this bag allows to define the visualisation modes. Currently, we consider objective or subjective view.
- `Location (entity, x, y, z)`: this bag keeps tracks of the location of the different entities displayed in the 3D scene.

In addition, we have a bag `Command (command, entity, p1, p2, p3)` that is associated to the 3D engine launched as an independent process. This bag is regularly interrogated by the 3D engine and the commands are collected one by one and executed in the context of the 3D scene. The reason of such architecture is first to dedicate one of the CPU cores to OpenSceneGraph managing the 3D scene. Second, changes are only done when an update of the scene is required, if no command is present nothing needs to be recomputed and changed. Third, this allows the 3D engine to pace the rhythm of command executions. If the rendering is very complex then the frequency of update will slow down accordingly, the bag working as a buffer. Thus, the user experience does not suffer of the possible saturation of the 3D engine.

To deal with priorities, we do not consider a single `Command` bag but two. The second bag is associated to high priority commands. Thus, for instance, we can manage in priority the commands linked to the displacement of the user in a subjective view while adding new 3D objects in the background scene is managed when possible.

F. Other available frameworks

Other frameworks, not used in this paper use cases, have been built on top of the LINC middleware. Following the same patterns, objects are linked to a dedicated context. They define specific frameworks ready to be used to design new applications.

For instance, to integrate sensors and actuators we have built a framework that considers the main standards for wired and wireless technologies. Each technology is managed by a dedicated object acting as a gateway. All the objects share the same set of bags to hide the heterogeneity. In addition, we have developed another framework to integrate camera (movement detection, face detection), light systems or mobile robots. We also developed a framework for managing the dynamic creation of scenarios and their management in term of context and priorities. The three of them have been used in the context of building automation [3], [17]. Finally, we have also recently been working on a voice framework to integrate voice recognition and text to speech engine.

IV. Frameworks in context

After presenting the global architecture, this section shows how interactions involving objects from different frameworks can be easily encoded with rules, through several examples.

A. Architecture

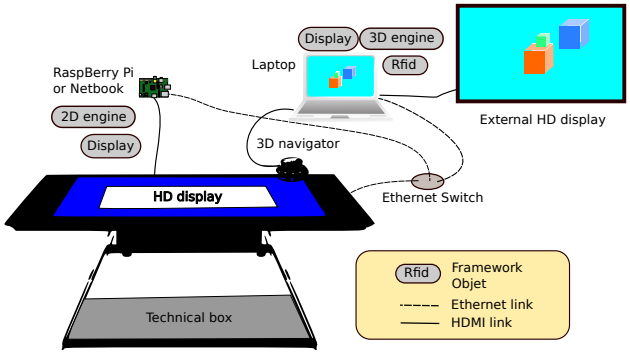


Fig. 2. Global picture.

Fig. 2 presents the current hardware and software setting. It contains the table described in Section II. In addition, there are two computing resources: a laptop and a raspberry pi or a netbook depending on the complexity we want to manage on the screen table. In real situation, they are embedded inside the technical box at the base of the table, hidden from the users. The table's screen is connected to the raspberry pi with a HDMI cable, offering a 1080p HD resolution. The Ethernet switch defines a local area network connecting the matrix of RFID readers, the raspberry pi/netbook and the laptop. From the software point of view, the objects of the different frameworks are distributed among the two computing resources.

The software configuration can be changed according to the application. For instance, this paper details two applications (in Section VI).

For the first application, the `RFID` and `Display` objects run on the laptop, the `2D engine` and another `Display` objects run on the Raspberry pi. For the second application, in addition a `3D engine` object runs on the laptop. The

raspberry pi is replaced by a netbook to use more intensively the `2D engine` and the `Display` objects.

Some LINC objects launch background processes that interact with them. For instance, the `2D engine` object launches a web browser, displayed in full screen, responsible for displaying the generated HTML + SVG files. The object `3D engine` launches OpenSceneGraph engine also in fullscreen. Finally, the object `Display` launches on demand one or more instances of VLC multimedia viewer.

B. Examples of simple interactions through rules

To change the background with a card: In this example (Listing 5), the background displayed on the screen is changed when a card of type `background` is put anywhere on the table. The first line of the precondition makes the rule fire only for tags that are known to define a background. Then (line 2), whenever a card of this type is on the table, line 3 finds the `filename` image corresponding to the `tagid`. Then, in the the performance phase the action (line 6) changes the background of the table with the content of filename. After the change, it is not necessary to let the card on the table.

```
1 ["RFID","Type"].rd(tagId, "background")&
2 ["RFID","TagStatus"].rd(tagId, "in")&
3 ["2D_Engine","Media"].rd(tagId, filename)
4 ::
5 {
6  ["2D_Engine","background"].put(filename);
7 }.
```

Listing 5. Rule to change the background when a card is put on the table

You can define as many backgrounds as you want, you just need to insert a resource defining the type of your RFID tag as a `"background"` in the bag `"Type"` and a resource to associate the tag id to an image filename in the bag `"Media"`.

To display a video at the location defined by the card: This example aims at starting a video when a card of type `video` is put. The card's position defines the top-left corner of the video. Listing 6 gives the rule implementing this scenario. As previously, lines 1 and 2 make the rules fires when a video card is put on the table. Line 3 gives the card's position. Finally, line 4 finds the video to be displayed from the tag id. The performance then embeds in one transaction:

- ensuring that the card is still there (line 7);
- starting of the video player with (posX,posY) (line 8);
- saving state "started" for the video (line 9).

```
1  ["RFID","Type"].rd(tagId, "video")&
2  ["RFID","TagStatus"].rd(tagId,"in")&
3  ["RFID","Position"].rd(tagId, posX, posY) &
4  ["RFID","Mapping"].rd(tagId, videoId) &
5  ::
6  {
7   ["RFID","TagStatus"].rd(tagId, "in");
8   ["Display","videoPlayer"].put("vlc", videoId, posX,
        posY, "640", "480", "True");
9   ["Display","videoPlayer"].put(videoId, "started");
10 }.
```

Listing 6. Display video at card's position

To switch a video currently played as a full screen video: This scenario uses in addition to the card that started a video as in previous scenario, a card that defines the modality `full-screen`.

Listing 7 implements such scenario. Lines 1-3 check that both cards are on the table. Line 4 fires when the video has been started (by rule in Listing 6). The performance phase checks that both cards are still on the table and the video is still in started state. In these conditions, we switch the video player to full screen by adding a resource in the bag `videoPlayerCommand` (line 10).

```
1  ["RFID","TagStatus"].rd("tag_fullScreen", "in")&
2  ["RFID","Type"].rd(videoId, "video") &
3  ["RFID","TagStatus"].rd(videoId, "in")&
4  ["Display","videoPlayer"].rd(videoId, "started")&
5  ::
6  {
7   ["RFID", "TagStatus".rd(videoId, "in") ;
8   ["RFID","TagStatus"].rd("tag_fullScreen", "in");
9   ["Display","videoPlayer"].rd(videoId, "started");
10  ["Display","videoPlayerCommand"].put(player, "fs_on");
11 }.
```

Listing 7. Put video in full-screen

It is then necessary to write a rule (Listing 8) to quit the full screen mode if the full screen card is removed from the table. This rule is triggered when the full screen control card is `"out"` (line 1). As previously, the performance checks the player and the cards' status and adds a resource in the bag `videoPlayerCommand` (line 10).

```
1  ["RFID","TagStatus"].rd("tag_fullScreen", "out")&
2  ["RFID","Type"].rd(videoId, "video") &
3  ["RFID","TagStatus"].rd(videoId, "in")&
4  ["Display","VideoPlayer"].rd(videoId, "started")&
5  ::
6  {
7   ["RFID", "TagStatus".rd(videoId, "in") ;
8   ["RFID","TagStatus"].rd("tag_fullScreen", "out");
9   ["Display","VideoPlayer"].rd(videoId, "started");
10  ["Display","VideoPlayerCommand"].put(player,"fs_off");
11 }.
```

Listing 8. Quit full-screen

V. Discussion and related works

So far we have illustrated the simplicity of writing interactions with the proposed frameworks. The detailed examples showed how information coming from distributed sources may be aggregated as a complex distributed event.

In the literature such task is usually implemented with a publish-subscribe approach [18], where subscribers register to specific events generated by publishers (RFID readers in this paper). This has been applied for instance in the context of sensor networks [19]. With such a system it is possible to write code that would be similar to the precondition part of the rules presented in this section. For instance, to react to a tag detected in a specific area or to an external event. However, in a publish-subscribe approach when the system has to react upon a set of events or to be sure that the events are still valid when the actions have to be executed, the amount of additional code is not negligible.

With the framework developed on top of our middleware expressing an event as "one card is put in a specific area" and "another card of a specific type is put at the same time anywhere else" is simply a sequence of `rd()` tokens. In addition, defining what to do if a card is put on the table and immediately removed is possible thanks to the distributed transaction offered in the performance phase that

aborts a transaction if the conditions are not currently true. In a publish/subscribe approach it would correspond to remove events that are no longer true but are still present in the system. Dealing with that is not impossible, but it is for sure not an easy task.

Other works have used as us resources and tuple spaces to facilitate the coordination in a distributed system such as [20] or [21]. These work focus on providing context awareness to mobile applications. However, we believe that these approaches still rely too much on traditional object oriented paradigms to be flexible enough to address the hardware used in this paper. Other promising approaches for coordination of complex systems have been proposed in [22] and [23]. For instance, in [22], the authors propose to use the chemical reaction paradigm to model the tuples evolution. The basic idea is to let the tuples evolve and auto-regulate as in chemical reaction. Even though these approaches could help coping with complexity of interactions brought with the table, we see two limitations. Firstly, self evolving coordination may be too complex to build specific interactions such as the ones described in this paper. Secondly, this work seems to be only at the theoretical level since no implementation has been done.

VI. Applications

This section presents two applications using the table: the first one helps decision making, the second deals with urban mediation. In both cases, we show how the frameworks detailed previously have been used to build the applications. The main advantage of our approach is that the objects of the frameworks are very generic and thus highly reusable. The specificities of the applications are in the coordination rules.

A. Help for decision making

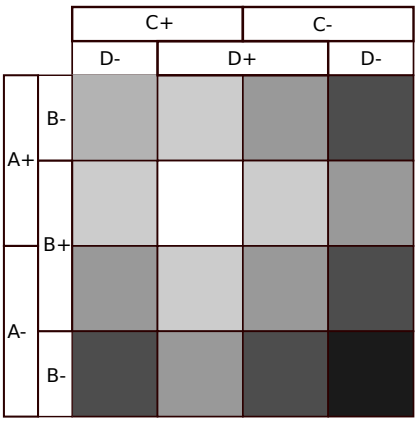


Fig. 3. Veitch Diagram

The first application uses quite complex interactions between several users around this table. The application allows to collect the opinion of a panel of people to elect the best equipment, concept or decision according to a set of criteria. In the present example, the panelists are asked to give their opinions about a set of smartphones according to the following criteria: aesthetic, user interface, size and autonomy. These criteria are denoted A, B, C and D, respectively, and can take the value positive or negative depending on the majority of votes from the panelists. The resulting information is displayed as a Veitch diagram as shown in Fig. 3.

The white cell contains the best choices that received 4 positive opinions. The adjacent cell contain choices that received 3 positive opinions. The darker a cell is, the more negative it is. The black bottom right cell contains the worst choice with 4 negative opinions. This section now details what is an interaction session and gives a few hints on the implementation.

1) Interactions: At the beginning, each panellist has a badge representing his/her identity. The master of session presents a smartphone and may display a video on the table by putting the corresponding card on it. Some modifier cards added to the table may modify the display either by switching to fullscreen or by launching a second video player with a 180 rotation to adapt to situation where people are all around the table. In this case, the second video uses the same video flow (without sound track) and is synchronised with the first one. Once the presentation is done, the vote may start. The master of session places the card corresponding to the criterion (e.g., aesthetic) and then the table display shows two areas, one green to collect the badges of panelists liking the smartphone design and one red for those that are not enthusiastic. An additional video or photo specific to this criterion may also be displayed. Then, the master of session triggers the vote by placing an hourglass (tagged with an RFID) on the table. A timer indicates at each corner of the table the remaining time for the vote. Each panellist put her badge on the table according to her opinion. A circle is displayed around the badge to return a feedback to the user. Different modalities may be configured at the beginning of the session to control the vote:

- the duration of a vote phase;
- missing vote is considered as negative or positive;
- a vote is definitive or not.

Once the timer reaches zero the votes are stored for further processing and the master of session can go to the next criterion. When all the criteria have been considered, the master of session can go to the next smartphone. At any moment, the master of session may place a card on the table to display or print current the status of the Veitch diagram.

2) Implementation: The full application described here may be implemented by using small variation of the basic interactions involving `Display`, `2D_Engine` and `Rfid` objects presented previously.

A specific `Application` object has been added. It contains bags to store panelists identities, votes and current step in the session (smartphone number and criteria number). This object is dedicated to the application and is the only one that is not re-usable. Associated with the coordination rules, this defines the logic of the applications. All the other objects are just a mean to access to the external world: table, screen, etc.

The basic settings, configuring a working session, are done through initialisation rules that define modalities such as default value of missing vote or the identity of the panelists. Note that using the proposed framework is very appropriate

since adding new features simply requires to add new rules. Existing rules can continue to work without concern. For instance, we can use an initial round getting the identities of the panelists rather than using a configuration rule. This can be done without any other impact than replacing the initialisation rule with the following rule required to obtain the identity of the panelists.

Registration: The following rule (Listing 9) starts when the *registration card* is placed on the table. It basically stores the `tagId` of all the users who placed their id badge on the table, in the bag `Users` of the object `Application`.

```
["RFID","Status"].rd("tag_registration","in") &
["RFID","Status"].rd(tagId,"in") &
["RFID","Type"].rd(tagId,"badge") &
::
{
 ["RFID","Status"].rd("tag_registration","in") &
 ["RFID","Status"].rd(tagId,"in") &
 ["Application","Users"].put(tagId)
}.
```

Listing 9. Registration

When the *registration card* is removed, the resource `("tag_registration","in")` is removed from the bag `Status` replaced by `("tag_registration","out")`. This immediately stops the effect of the registration rule: performances will fail on the first token because of the absence of the resource.

As a result, when the card is removed, all the registered users are in the bag `Users` of the object `Application`.

Vote: A session of vote would then start with the following rule:

```
["RFID","Status"].rd("tag_vote","in") &
::
{
 ["Application","Step"].put("vote_started")
 ["2D_Engine","Background"].put("vote_bg")
}.
```

Listing 10. Vote starting

It defines that we are in the step `"vote_started"` and changes the table background creating a zone for the positive votes in green and a zone for the negative votes in red.

The vote round stops either when the delay associated to the round has expired with the following rule:

```
["Application","Step"].rd("vote_started") &
SLEEP: 30
::
{
 ["Application","Step"].get("vote_started")
 ["2D_Engine","Background"].put("end_vote_bg")
}.
```

Listing 11. Vote end (time out)

or when the master decides the end of the round, for instance because everybody have voted. This is done when she removes the cards associated to the `tag_vote`.

```
["Application","Step"].rd("vote_started") &
["RFID","Status"].rd("tag_vote","out") &
::
{
 ["Application","Step"].get("vote_started")
 ["RFID","Status"].rd("tag_vote","out") &
 ["2D_Engine","Background"].put("end_vote_bg")
}.
```

Listing 12. Vote end (master decision)

At the application level it does not matter which of the 2 rules has been applied. As both of them consume the resource `vote_started` the execution of one will prevent the other to be executed.

The combination of the two rules defines that we are no longer in the step `"vote_started"` and changes the table background to indicate the end of the round.

The rule managing the vote itself is given in Listing 13.

```
["Application","Step"].rd("Vote_started") &
["RFID","Status"].rd(tag_product,"in") &
["RFID","Type"].rd(tag_product,"product") &
["RFID","Status"].rd(tag_property,"in") &
["RFID","Type"].rd(tag_property,"property") &
["Application","Users"].rd(user_tagId) &
["RFID","Status"].rd(user_tagId,"in") &
::
{
 ["Application","Step"].rd("vote_started")
 ["RFID","Status"].rd(tag_product,"in")
 ["RFID","Status"].rd(tag_property,"in")
 ["RFID","Status"].rd(user_tagId,"in")
 ["RFID","Area"].rd("positive",user_tagId)
 ["Application","Vote"].put(tag_product,
     tag_property,user_tagId, "+")
}
{
 ["Application","Step"].rd("vote_started")
 ["RFID","Status"].rd(tag_product,"in")
 ["RFID","Status"].rd(tag_property,"in")
 ["RFID","Status"].rd(user_tagId,"in")
 ["RFID","Area"].rd("negative",user_tagId)
 ["Application","Vote"].put(tag_product,
     tag_property,user_tagId, "-")
}.
```

Listing 13. vote

The first token of the precondition and of each transaction in the performance phase is a guard ensuring the rule is only active during the vote. (lines 1, 10 and 18) When the vote ends, the resource `("vote_started")` is removed from the bag `Step`. As a result, all the transactions will fail on the action `rd("vote_started")`. The `rd()` operations on `(tag_product,"in")` and `(tag_property,"in")` (lines 2 and 4 in the precondition) ensure that the vote is considered for the current property of the current product. Token in line 6 of the precondition returns all the users registered for the vote. This is a very natural manner to avoid considering votes by an unregistered person. Obviously, as this information is stored in a bag, it would be possible at any time to register or unregister a user. Finally, the last token of the precondition waits for each user tag to be put on the table. For each of these tags, a performance is triggered. The performance is composed of two very similar transactions. The first four tokens (lines 10 to 13 for the first transaction) are guards to ensure that the vote is still open, for the current product, the current property and that the user tag is still on the table. The last two tokens of the transactions define the vote and save it. A vote card can be in only one area, and the two areas cover all the table. Then, the two transactions are exclusive. The first transaction succeeds for positive votes, while the second transaction succeeds for negative votes (enforced by lines 14 and 22). Hence we ensure that one and only one transaction will succeed, counting the vote correctly.

When all the votes have been done for a criterion, the users can remove their voting card.

This rule applies for all the products and the criteria inside a product since the context is given by the corresponding cards placed on the table.

Then, the master simply removes the criterion card and replaces it by the card for the next criterion. This will trigger a new branch in the precondition, at line 4. The value of `tag_property` now contains the value of the new property. Once all the criteria have been voted for a product, the master replaces the product card by another product card and the vote may continues with a new round of properties.

The goal of this section was not to describe all the rules involved in the application but to show how problems that seems quite complex may be managed with only a few number of rules. Moreover, as the application can be decomposed in steps and each rule associated to a step is controlled by the presence of some specific resources in some bags, we can easily avoid the "unwanted" competition between rules. Thus, it is very easy to guaranty that a set of rules verified in isolation will not introduce flows in the whole set of rules defining the full application.

B. Urban Mediation

The context of this second application is the mediation in between people who design urban infrastructures, deciders of the project and inhabitants who live or will live inside or nearby the area. The table is used both as a ground for a physical model of the urban sector under consideration and as an interaction medium to augment the model with 3D virtual representation of the same urban scene as shown in Fig. 4.

Physical building models can be put on the table to figure out the area to be considered. These buildings are made with a 3D printer and they are equipped with Rfid tags to allow interaction with the table.

It is possible thanks to the framework `2D Rendering` to display various useful information on the table screen.

- Static information:
  - maps or aerial view;
  - parking lots, road, green spaces;
  - electricity, communication, water or drain networks;
  - underground transportation;
  - meta information such as reserved location;
  - specific difficulties linked to the ground nature (floodplain, rocky soil, historic relic).
- dynamic information:
  - graphs and statistical information;
  - animation to render wind direction (venturi effect), sound propagation;
  - historical traffic data;
  - videos.

An additional external screen or a projector is used to render the urban area as a virtual 3D scene. This allows the user to either have an objective view, flying over the full scene or a subjective view, moving around directly within the scene. This uses the framework `3D Rendering`. In Fig. 4, an objective view of the area is displayed.

This section now details how the user may interact with the application and then gives some hints on the implementation.

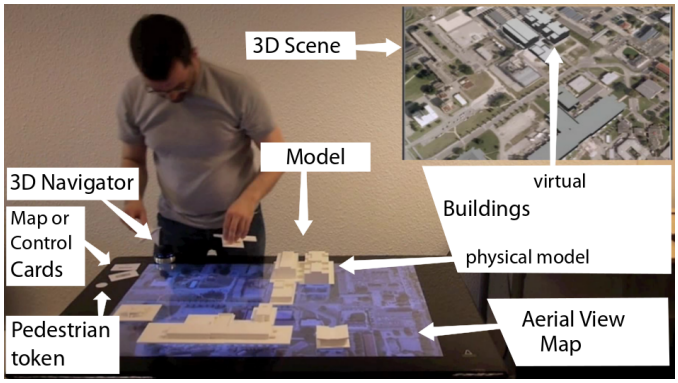


Fig. 4. Urban mediation.

1) Interaction: In this application we do not have a strict succession of steps as in the previous application. On the contrary, we just have different *"modes"* that can be freely set at any moment by the user through the use of a context card placed on the table.

For instance, a mode populates the physical model (on the table) and thus the virtual scene (on the wall) with buildings and other urban elements.

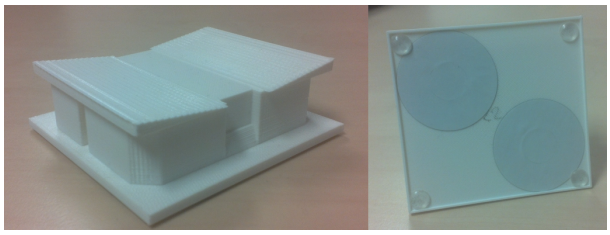


Fig. 5. Printed building and its verso equipped with RFID tags.

This is done by putting on the table 3D building models as the one shown in Fig. 5. Each building has on the verso two RFID tags placed respectively in the top left and bottom right corners. These tags are used for the identification, the location on the table and the orientation of the building.

Given the respective positions of the two tags, the barycentre corresponds to the centre of the building that is required to compute the actual position in the virtual scene. Using two tags per building model also allows to have a rough idea of the orientation of the building. Given the resolution of the table we can consider steps of 45 degrees.

We define a single logical tag (concatenation of the two physical ones) that uniquely identifies the building. Then, we associate this logical tag to the physical building model, its 3D model file, its location, its orientation and its virtual representation in the 3D scene. This bridges the physical world and the virtual one.

The 3D printed shapes represent, in the current urban scene, the real buildings and the buildings for which the plans are already known. However, sometime, for discussion in the early phases, we want to have an idea of the impact of a possible building. In this case, we use different black boxes as shown the Fig. 6 prefiguring buildings of different shapes and sizes. These blocks are also equipped with two tags.

Fig. 6. Generic shapes for prefiguring future buildings footprint.

In order to enrich the urban scene, we can also use some *"ink pads"* that once applied on the table create 2D markers on the background and elements in the 3D virtual scene. This concerns for instance trees, hedgerow and other scenery elements.

The background can also be modified by placing cards on the table. Backgrounds can be superposed in order to reveal various additional informations as described in the previous section.

It is also possible to move a token representing a pedestrian within the physical model and acting on her field of vision with the 3D Navigator. This is a device that includes the functionality of a classical joystick in X and Y combined with informations push and pull to add the Z axis. In addition, you can rotate it and you have one button on each side.

As a result, we can see on an external screen, the 3D virtual scene actually seen by the pedestrian. Moving the token allows to progress in the 3D scene, visiting the different places. Rotation of the pedestrian is done through the 3D navigator. The pedestrian can thus turn on herself getting a 360 view. A vision cone is displayed on the table to show the current pedestrian field of view. The same device is used to manage the head movement up, down, right and left. Finally, the two buttons on the right and left sides of the device allow the pedestrian to respectively go up and down. This allows to consider the view from the different floors of a building. The corresponding floor number is displayed on the table near the vision cone.

All the modifications of the physical model are immediately echoed on the 3D virtual scene and thus, the potential impact can be better evaluated.

Switching back to objective view is done via a special card placed on the table. In this mode, the 3D navigator is used to fly over the scene with the same capability of movements.

In objective mode, it is also possible to enrich the 3D scene according to lights and sun position. It is thus possible to see the impact of a building in term of shadow in the neighbouring according to the season, the hour, etc.

It is also possible to associate a texture to the black boxes to test alternative to a project and how the future building can be better integrated with the existing ones.

Finally, it is possible to launch video on the table screen or on the external one.

2) Implementation: The application uses a combination of the three frameworks described in this paper. These frameworks provides very generic and reusable objects: a large part of them is already used in the previous application. Some of the interactions and the application logic of this second application is given in the following as illustration.

The first rule manages the 3D printed buildings.

```
["RFID","TagType"].rd(id,"building") &
["RFID","TagPosition"].rd(id,x,y) &
COMPUTE: x1, y1, z1 = transform(x, y) &
["OSG","Entity"].rd(id,filename) ;
::
{
 ["OSG","Command"].put("entity",id,filename,"","") ;
 ["OSG","Command"].put("translate",id,x1,y1,z1) ;
 ["OSG","Command"].put("scale",id,"1","1","1") ;
}.
```

Listing 14. Buildings

The precondition waits for all the tags of type `building` in order to manage only tags corresponding to the 3D printed building models. It then reads the location of the model on the table. From the 2D coordinates it computes the corresponding 3D coordinates in the virtual scene through the method `transform()`. The method `transform()` is written in python [24] code and may be very simple if we just do a simple translation for `x` and `y` and set `z` to 0. It can be more complex if we refer to a model of the ground that in addition can give the `z` coordinate according to `x` and `y`. The last token allows to obtain the 3D model file associated to the building.

The performance phase then inserts in the `Command` bag the three commands required to create in 3D virtual scene with the representation of the building. The first one creates the entity according to its 3D model if it is not already done. The second places the entity at the right place. The third changes the scale of the object in the 3 dimensions. Indeed, by default, when a new graphical object is created it has an initial scale of 0 to stay hidden. Then we can make all the transformations (translate, rotate, etc.) before acting on the scale in order to make it appears at the correct place.

Moving a building will trigger another instance of this rule with a new resource for the tag position and the building will be moved accordingly in the 3D virtual scene. Removing a building is detected by the status of the tag that becomes `out` and in this case, we just set the scale of the object to 0 in order to hide the building.

The second rule manages scenery elements that can be added to the 3D virtual scene. It is almost similar to the previously presented rule concerning buildings. The main difference is that we have not one physical tagged object per virtual one. Thus, we use a tag equipped *"ink pad"* for each type of element we want to consider. In the following example, we consider a small tree. Each time this the pad is put on the table, a new instance of the small tree is created as if a physical 3D object where placed. As a result, a 3D virtual small tree appears at the corresponding location in the 3D virtual scene.

```
["RFID","TagPosition"].rd("small_tree",x,y) &
COMPUTE: x1, y1, z1 = transform(x, y) &
COMPUTE: id = "elt_"+x+"_"+y &
::
{
 ["OSG","Entity"].put(id,"arbre.osg") ;
 ["OSG","Command"].put("entity",id,"tree1.osg","","") ;
 ["OSG","Command"].put("translate",id,x1,y1,z1) ;
 ["OSG","Command"].put("scale",id,"1","1","1") ;
}.
```

Listing 15. Small tree ink pad

The precondition phase reads the 2D coordinates of the pad on the table. Each pad has an unique identifier, here it is `small_tree`. The 3D coordinates are computed as in the previous rule. The last token creates an `id` that will be associated to the virtual entity that is going to be created. This entity is in fact an instance of the type of tree associated to the pad. The id is of the form `elt_<x>_<y>` and is unique for a given location. Indeed, we consider that we can have a single scenery element at a given place.

In the performance phase, we declare a new entity referred by its `id` and associated to it the corresponding 3D model here in the format `.osg` native to OpenSceneGraph. We then insert in the `Command` bag the three same commands as the previous rule to make the trees appear in the scene.

This rule is very generic and we can manage, with small variants, pads associated to various scenery elements. For instance, we can act on the 3D model by using another model file or we can act on the size by using different scale factors. In LINC it is possible to generate dynamically new rules, thus it is very simple to add new ink pad on the fly via a simple web interface.

To remove an object created by a tag equipped ink pad, we can use a tag equipped rubber as shown in the following rule.

```
["RFID","TagPosition"].rd("rubber",x,y) &
COMPUTE: id = "elt_"+x+"_"+y &
["OSG","Entity"].rd(id,filename) &
::
{
 ["RFID","TagPosition"].get("rubber",x,y) &
 ["OSG","Entity"].get(entityname,filename) ;
 ["OSG","Command"].put("scale",entityname,"0","0","0") ;
}.
```

Listing 16. Rubber

Each time the rubber is detected at a place, we compute (line 2) the `id` of a potential element that would have been created by a pad. If the entity exists (line 3) then we can go further and remove the element. This is done in the performance part where we consume the information about the presence of the rubber that is no longer required. Keeping this information would prevent to put later another element at this place, because it would be automatically immediately removed. The entity itself is removed from the corresponding bag and finally we hide the virtual element of the 3D scene by setting a scale to 0. This creates an orphan hidden virtual element that will be garbaged if we create a new scenery entity at the same place since it will have the same id.

If the rubber is placed on an empty location, nothing is done and when it is removed from the table the resource `("rubber",<x>,<y>)` is removed from the bag `TagPosition` preventing any erase action to be done at this place.

The full application contains two tens of rules following more or less the same scheme. Inputs from the table and the 3D navigator are combined and used to act on the table screen and/or the 3D virtual scene. Here too, as the scenario depends on a specific combination of tags read by the table, we can ensure that each scenario is guaranteed to be executed with no impact from the others. This decreases the risk of unexpected behaviour.

Another advantage of this approach is the possibility to use additional interfaces at a little cost. For instance, we have used a 3D navigator to basically get the information X, Y, Z, tilt, pan, roll that could also be easily obtained with a combination of 3 axis magnetometers, accelerometers and gyroscopes. As the interface with the 3D navigator is a set of bags receiving the position information, we can replace the 3D navigator with another device based on magnetometers, accelerometers and gyroscopes without changing the coordination rules defining the application.

VII. Conclusion

This paper has presented an innovative hardware and several frameworks easing the development of applications. The hardware, is a table combining a full HD display and a set of 384 RFID readers allowing to return the location of several tens of object tagged with RFIDs.

The frameworks are built on top of our in house rule-based middleware LINC. LINC relies on bags containing resources modelling our system, production rules and distributed transactions. A framework has been defined to map events coming from the table into resources stored in bags allowing the resources to be accessible with simple rules. This allows to react to event composing several RFID tags and to embed events verification in distributed transactions.

In addition, we have defined frameworks to offer a visual feedback to the user via displays. This includes 2D and 3D graphical objects rendering and multimedia contents such as videos.

This paper has shown firstly in isolation, through simple examples the genericity of these frameworks. Then, we have described how they may be combined and specialised via coordination rules to target two different application domains. This shows how the combination of the hardware, the middleware and the high level frameworks helps designing applications while offering a high degree of re usability of the frameworks components. The amount of work is decreased since a large part of the application is already available in the existing frameworks.

Future work is to enrich the tool kit around this table. We can integrate more external devices, sensors and actuators. For instance, cameras to deduce the number of people around the table (e.g., counting the detected faces), sensors to define the distance of the users from the table. We can also add voice interface. All these additional informations combined with the ones returned by the table may offer a richer user experience in order to target other application domains.

ACKNOWLEDGMENT

This work has been partially funded by the FP7 SCUBA project under grant nb 288079 and FUI Rapsodie project under grant nb F1209039V.

REFERENCES

[1] M. Louvel and F. Pacull, "A coordinated matrix of RFID readers as interactions input," in *SENSORDEVICES 2013, The Fourth International Conference on Sensor Device Technologies and Applications*, 2013, pp. 91–96.

[2] M. Louvel and F. Pacull, "LINC: A compact yet powerful coordination environment," in *Coordination Models and Languages*, ser. Lecture Notes in Computer Science, E. Kuhn and R. Pugliese, Eds. Springer Berlin Heidelberg, 2014, pp. 83–98.

[3] L.-F. Ducreux, C. Guyon-Gardeux, S. Lesecq, F. Pacull, and S. R. Thior, "Resource-based middleware in the context of heterogeneous building automation systems," in *IECON 2012, The 38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2012, pp. 4847–4852.

[4] J.-M. Andreoli, F. Pacull, D. Pagani, and R. Pareschi, "Multiparty negotiation of dynamic distributed object services," *Journal of Science of Computer Programming*, vol. 31, pp. 179–203, 1998.

[5] D. Arregui, C. Fernström, F. Pacull, G. Rondeau, and J. Willamowski, "STITCH: Middleware for ubiquitous applications," in *sOc 2003, The second International Smart Object Conference*, 2003.

[6] N. Carriero and D. Gelernter, "Linda in context," *Commun. ACM*, vol. 32, pp. 444–458, April 1989.

[7] T. A. Cooper and N. Wogrin, *Rule Based Programming with OPS5*. Morgan Kaufmann, July 1988.

[8] P. A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency control and recovery in database systems*. Boston, MA, USA: Addison-Wesley Longman Publishing, 1987.

[9] R. Wang and X. Qian, *OpenSceneGraph 3.0: Beginner's Guide*. Packt Publishing, 2010.

[10] R. Wang and X. Qian, *OpenSceneGraph 3 Cookbook*. Packt Publishing, 2012.

[11] F. Kerger, *OGRE 3D 1.7 Beginner's Guide*. Packt Publishing, 2010.

[12] M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[13] J. C. Mong and D. F. Brailsford, "Using SVG as the rendering model for structured and graphically complex web material," in *DocEng 2003, The 2003 ACM symposium on Document engineering*, New York, NY, USA: ACM, 2003, pp. 88–91. [Online]. Available: http://doi.acm.org/10.1145/958220.958236

[14] "HTML5," http://www.w3.org/html/wg/drafts/html/CR/.

[15] J. J. Garrett, "Ajax: A new approach to web applications," http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications, 2005.

[16] T. Brown, *Dynamic apache with ajax and json*, 1st ed. O'Reilly, 2006.

[17] F. Pacull, L.-F. Ducreux, S. Thior, H. Moner, D. Pusceddu, O. Yaakoubi, C. Guyon-Gardeux, S. Fedor, S. Lesecq, M. Boubekeur, and D. Pesch, "Self-organisation for building automation systems: Middleware LINC as an integration tool," in *IECON 2013, The 39th Annual Conference of the IEEE Industrial Electronics Society*, Nov 2013, pp. 7726–7732.

[18] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003.

[19] E. Souto, G. Guimarães, G. Vasconcelos, M. Vieira, N. Rosa, C. Ferraz, and J. Kelner, "Mires: a publish/subscribe middleware for sensor networks," *Personal and Ubiquitous Computing*, vol. 10, no. 1, pp. 37–44, 2006.

[20] J. Barbosa, F. Dillenburg, G. Lermen, A. Garzão, C. Costa, and J. Rosa, "Towards a programming model for context-aware applications," *Computer Languages, Systems & Structures*, vol. 38, no. 3, pp. 199–213, 2012.

[21] C. Julien and G.-C. Roman, "Egospaces: Facilitating rapid development of context-aware mobile applications," *IEEE Transactions on Software Engineering*, vol. 32, no. 5, pp. 281–298, 2006.

[22] M. Viroli, M. Casadei, S. Montagna, and F. Zambonelli, "Spatial coordination of pervasive services through chemical-inspired tuple spaces," *ACM Trans. Auton. Adapt. Syst.*, vol. 6, no. 2, pp. 14:1–14:24, Jun. 2011. [Online]. Available: http://doi.acm.org/10.1145/1968513.1968517

[23] M. Viroli, D. Pianini, and J. Beal, "Linda in space-time: an adaptive coordination model for mobile ad-hoc environments," in *Coordination Models and Languages*. Springer, 2012, pp. 212–229.

[24] M. Lutz, *Programming Python*. O'Reilly Media, Inc., 2006.

Fiber-Coupled Microcavity Probe – A Novel Optical Biosensor for Near-Field Real-Time Monitoring of Biomolecular Interactions

Nichaluk Leartprapun[1], Zachary Ballard[1], and Jimmy Xu[1, 2]
[1]School of Engineering
Brown University
Providence, USA
[2]WCU Program
Seoul National University
Seoul, South Korea
e-mail: Nichaluk_Leartprapun@brown.edu, Zachary_Ballard@brown.edu, Jimmy_Xu@brown.edu

Abstract – **We report on a novel optical biosensor design for near-field sensing based on a fiber-coupled microcavity. The device operates on the bases of sensing in the evanescent near-field zone via the amplitude and phase modulation of the reflected interference patterns. The integrated probe design offers a low-cost, robust and easy-to-fabricate alternative for label-free real-time biosensing. The first generation, pre-optimization device has already demonstrated sensitivity in the range of 10^{-4} and 10^{-5} refractive index units and 40 nm per refractive index unit. We have also demonstrated the potential use of this device in specific binding assays or concentration analyses by monitoring in real-time the solution-phase self-assembly of 5 Å aminosilane monolayer from varying bulk concentration. The application of this device as a new platform for point-of-care calibration-free concentration analysis is promising.**

Keywords – biosensor, microprobe, microcavity, fiber-optics, real-time sensing, silanization.

I. INTRODUCTION

Despite the prospects that optical biosensors could make sensitive and accurate point-of-care diagnostics a reality, bringing the technologies out of the lab setting (optical table) and into scalable devices for field use remains a tremendous challenge to many existing state-of-the-art optical biosensor platforms. We previously contributed a work-in-progress report on a novel label-free optical biosensor with a probe structure that is low-cost, robust, easy-to-fabricate, and has the potential for in-vivo or in-situ sensing and probing of bio-environments in the field settings [1] [2].

Electromagnetic waves interact with biological matter in various ways ranging from transmission, reflection, absorption, scattering and tunneling [3]. Utilizing different modes of light interactions with biological matter, a wide variety of optical biosensor platforms have been developed that probe the bio-environment via its localized refractive indices without the need for fluorescent markers or dyes. Surface-Enhanced Raman Spectroscopy (SERS) sensors rely on the plasmonic enhancement of light absorption and scattering to detect minute differences in the morphology of the sensing surface [4]. The modulation of wave characteristics of light upon reflections at interfaces allow interferometric biosensors to monitor the refractive index of the samples [5]. Surface Plasmon Resonance (SPR) sensors, one of the sensing platforms that have seen the most commercial success, rely on the absence of reflection during the phenomenon of light coupling into a surface mode at a specific incident angle [6]. State-of-the-art SPR sensor modules have been able to resolve refractive index in the orders of 10^{-7} to 10^{-9} refractive index unit (RIU). Garnering increasing research interests recently, the Whispering Gallery Mode (WGM) sensors are able to detect a single molecule or nanoparticle by coupling light into optical microcavity resonators [7][8]. These optical biosensor platforms are label-free, allowing for a more elegant detection protocol. Many of them are also capable of real-time sensing, an ability crucial to cellular studies and drug discoveries [9][10].

Though many breakthroughs have been made in regard to detection sensitivity, translating these from the laboratory into a portable and robust micro-device has yet seen significant success [2]. Existing high sensitivity biosensor platforms often required precise controls of moving parts on the nano-scale. Furthermore, the extensive and involved micro and nanofabrication required during the fabrication processes of some of these platforms make them much too expensive for most practical uses. Fiber-optic based platforms with sub-micron sized dimensions prove to be one of the more promising technologies in regard to the practicality of the device due to their needle-like geometries [11]. Despite the functional advantages and published sensitivities around 10^{-4} to 10^{-7} RIU, optical fiber based platforms still remain several orders of magnitude less sensitive than other bio-sensing technologies such as Surface Plasmon Resonance sensors, Whispering Gallery Mode sensors, and interferometric sensors [12]. However, novel fiber-based platforms could increase the capacity for sensitivity and serve as a more robust and low-cost biosensor solution. A portable and highly sensitive in-vivo device could have immense impact in driving down the cost

of healthcare and guiding vaccine and medicinal distribution networks [13].

In this work, we present a novel optical biosensor with a probe structure that has the potential for in-vivo or in-situ sensing and probing of bio-environments in real-time. This simple, portable, micro-scale device can detect, via phase-shift and amplitude change of the device response, minute changes in its microcavity's immediate vicinity (evanescence near-field zone). It could be used in diagnostics by detecting the presence of bio-targets such as deoxyribonucleic acid (DNA) and many disease biomarkers. In addition, it could be a possible low-cost tool to aid in the understanding of specificity and affinity of specific binding pairs by extracting kinetic information from real-time analysis. The ability to derive kinetic data from real-time measurements also opens the possibility of using this device as an alternative low-cost platform for calibration-free concentration analysis (CFCA) in point-of-care diagnostics [14][15]. The device integrates several characteristic features of existing more complex platforms such as multiple non-normal reflections and optical coupling from waveguide to microcavities into a single probe structure. The micro-probe consists of a micro-cavity formed at, and optically coupled to the tip of an optical fiber, resulting in a compact, easy-to-fabricate, and bio-compatible design for refractive index sensing. Experiments for the first generation device, with non-optimized probe geometries, have yielded a limit of detection of 10^{-4} to 10^{-5} RIU with the phase shift of 40 nm/RIU.

This article will first describe the fabrication and structure of the fiber-coupled micro-cavity probe. Then, propose the working principles of the device and present the experimental validations of the proposed principles. Finally, the article will demonstrate experimentally the capability of the device in bulk refractive index sensing and real-time monitoring of biomolecular interactions within the evanescence near-field zone.

II. Device Description

The micro-probe is made by tapering hollow borosillicate tubes (1mm OD, .75 mm ID) down to tip diameters of 25-50 µm. The tapering was achieved using a Sutter Instrument P-2000 Micropipette Puller. A 20W Class IV CO_2 laser was used as the heating source while the borosilicate tube was pulled from both sides, as shown in Fig. 1. Custom programs were written for the pipette puller to achieve long and gradual tapers. The tapered tips were then melted with a gentle flame from standard butane lighter, whereupon the molten glass forms a solid glass spherical tip due to its surface tension. The symmetry of the spherical tips was maintained by constant spinning of the tips during the melting process. The total resulting structure consists of a finely tapered air cavity that extends into a solid glass spherical tip, as seen in Fig. 2. Then, standard SMF-28 telecom fibers were stripped, cleaved and inserted into this tapered cavity, eventually becoming wedged. The optical fiber remains stationary due to the cylindrical geometry of the tapered air cavity and fiber, and can be affixed upon its insertion point with simple epoxy. The range of spherical tip diameters tested so far were, but not limited to, 300-500 microns.

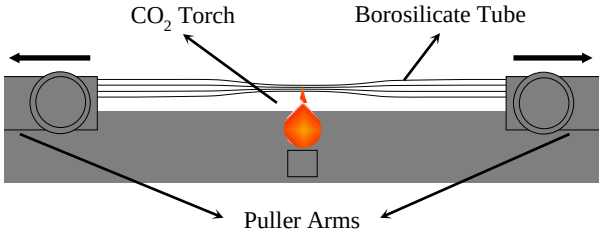


Figure 1. Borosilicate tube tapering process using a Micropipette Puller.

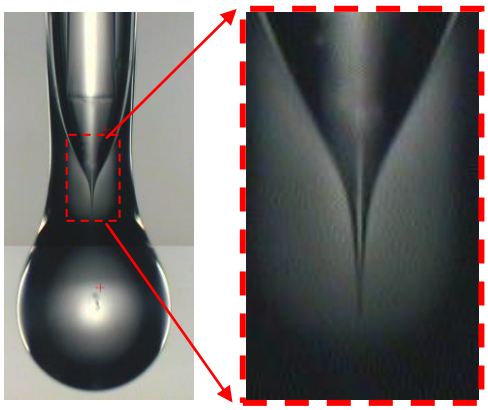


Figure 2. Device structure: Microscope image of SMF-28 fiber (125 µm diameter) wedged in tapered air cavity (red dotted outline) extending into glass spherical tip (410 µm diameter).

III. Working Principles

This section describes the working principles pf the fiber-coupled microcavity probe as a self-reference interferometer. Fizeau interferometry physics of the device is elaborated, followed by the discussion of its use as a sensor. Lastly, the experimental validations of the proposed working principles are presented.

A. *Fizeau Interferometry*

The resulting spherical probe structure contains two effective reflection surfaces for incident light through the optical fiber [16]. The first interface is between the end of the cleaved fiber and the air cavity, and the second interface is between the edge of the solid glass spherical tip and the outside environment, as shown in Fig. 3. Therefore, when incoming light through the fiber enters the device, the reflection from the second interface recombines with the reflection from the first interface as described by the reflection from an effective Fabry-Perot cavity.

The second interface is an effective one that is subjected to change by the material in the evanescence zone of the probe tip. The spherical tip forms a secondary (weak Q) cavity for the entering light scattered from the sharp air-tip.

Surrounding the outer surface of this microcavity is the aforementioned evanescence-zone. A change of material, temperature, humidity, or pH, even a minute one, in this zone would effectively change the size of this secondary cavity, which translates into the amplitude and phase modulation of the reflected light that can be measured from the overall interference pattern.

B. Use as a Sensor

Based on the Fizeau interferometer model, increasing the refractive index of the environment outside of the spherical tip naturally decreases the reflection coefficient for reflections below the critical angle and thus decreases the amplitude of the interference signal. However, this is not the only effect. Experiments show a clear phase change with changing refractive index in the exterior en vironments.

Due to the unique geometry of the sharply tapered air cavity between the two reflection interfaces, this device is able to bend the wave fronts of the fiber-outputted light. Fig. 4 illustrates the simulation of electric field and magnetic field intensity within the air cavity and the glass tip, showing the bent wave fronts as the electromagnetic wave escapes the tapered cavity. This bending of wave fronts allows for multiple reflections at multiple non-normal angles in the glass spherical tip and probe neck, eventually coupling the light back into the optical fiber. This light propagates in the spherical tip with a refractive index n_1 through a series of reflections with non-normal incident angle θ_i, where the reflections that meet the total internal reflection criteria are greatly affected by the Goos-Hänchen effect [17]. Changing the refractive index n_2 at the interface along the spherical tip will modulate the Goos-Hänchen shift, thus, shifting the interference pattern by a phase δ described by,

$$\tan\left(\frac{\delta_S}{2}\right) = \frac{\left(\sin\theta_i^{\,2} - n^2\right)^{\frac{1}{2}}}{\cos\theta_i} \qquad (1a)$$

$$\tan\left(\frac{\delta_P}{2}\right) = \frac{\left(\sin\theta_i^{\,2} - n^2\right)^{\frac{1}{2}}}{n\cos\theta_i} \qquad (1b)$$

where $n = n_2/n_1$.

Treating the device as an effective Fabry-Perot cavity [16], the total reflected intensity could be derived using the Airy Summation Method,

$$I_r(FP) = I_0\left[\frac{r_1^{\,2} + 2\cos(\delta)r_1r_2 + r_2^{\,2}}{1 + 2\cos(\delta)r_1r_2 + r_1^{\,2}r_2^{\,2}}\right] \qquad (2)$$

where r_1 and r_2 represent the reflection coefficient of the two interfaces of an asymmetric Fabry-Perot cavity. Substituting $r_1 = -r_2$ into (2) would recover the standard equation for the reflection of a symmetric Fabry-Perot cavity.

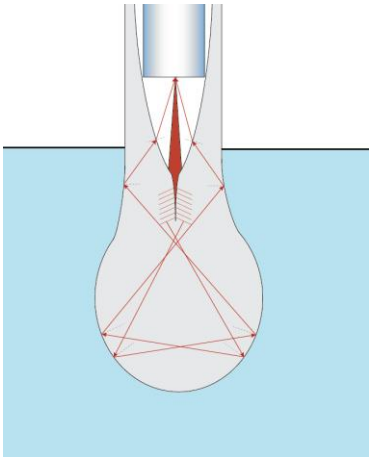


Figure 3. Ray-optics of light propagation in probe geometry.

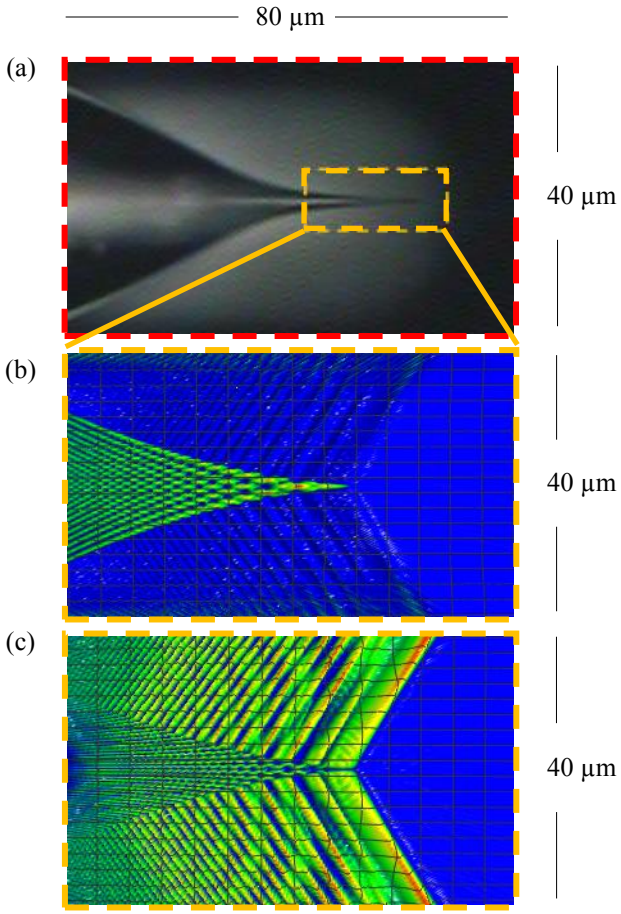


Figure 4. Tapered air cavity: (a) Microscope image of tapered cavity and Optiwave simulation of (b) electric field and (c) magnetic field intensity of light entering and escaping from the cavity.

To examine the effects of reflection angles on the output of the effective Fabry-Perot cavity model, (2) is used to simulate the interference patterns of the reflected signal in three cases: 1) reflection occurs at an angle smaller than the critical angles, 2) reflection occurs at an angle larger than

the critical angles, and 3) reflection occurs at multiple angles both smaller than and larger than the critical angles. Fig. 5 shows the simulated interference patterns of the three cases for n_2 values ranging from 1.00 to 1.30, corresponding to the critical angles from 42 degrees to 60 degrees. When the incident angle θ_i is set to 20°, the only effect of increasing the refractive index n_2 is the decrease in peak-to-peak amplitude due to the lessening of reflection coefficients for reflection below the critical angles. Changing the incident angle to 70°, however, causes the increasing refractive index to lose its effect on the reflection coefficients because of the reflection above the critical angle. Instead, the phase shift towards larger wavelength is observed as a result of increasing optical tunneling into the outside environment. If multiple reflections occur at multiple angles below and above the critical angle, both effects are observed and the reflected light experiences amplitude and phase modulation simultaneously as a result of varying refractive index in the evanescence zone.

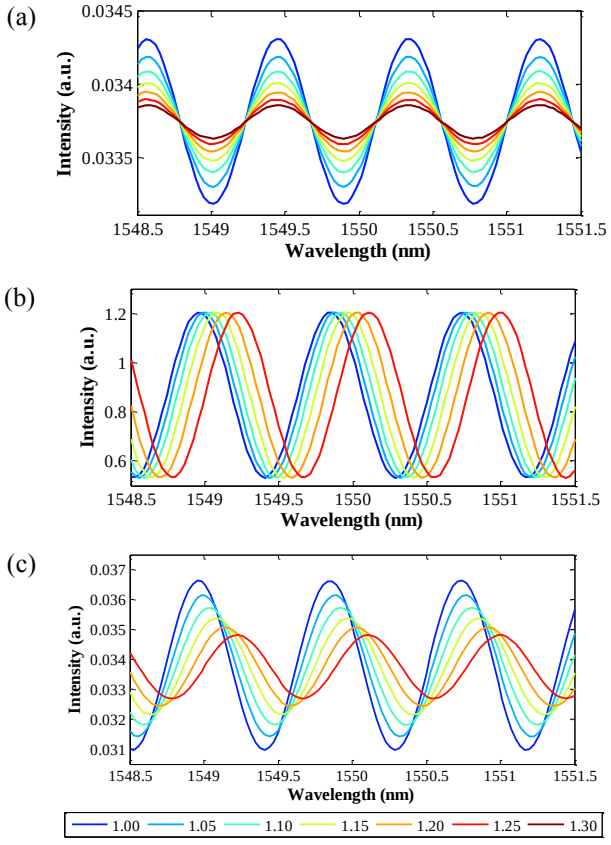


Figure 5. Simulation of interference patterns of reflected light: (a) reflection below critical angles only (θ_i = 20°), (b) reflection above critical angles only (θ_i = 70°) and (c) reflection both below and above critical angles (θ_i = 20° and θ_i = 70°).

C. *Experimental Validation of Working Principle*

To validate the model with the actual device output, Fig. 6 shows the model interference patterns produced from (2) for n_2 of air, water, and ethanol in comparison with the experimentally obtained results for the same environment in Fig. 7. The significant phase shift and amplitude decrease observed in the experimental results clearly indicate that multiple reflections at the incident angles both above and below the critical angles took place. This effective Fabry-Perot cavity model was able to describe the probe output remarkably well despite the complex structure and geometry of the device.

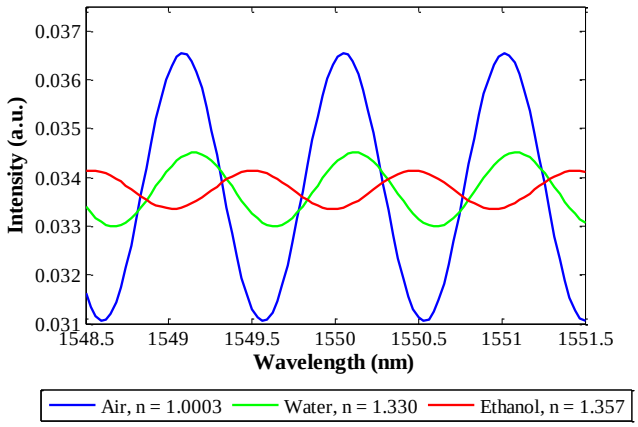


Figure 6. Output of probe model for outside refractive index environments of air, water, and ethanol.

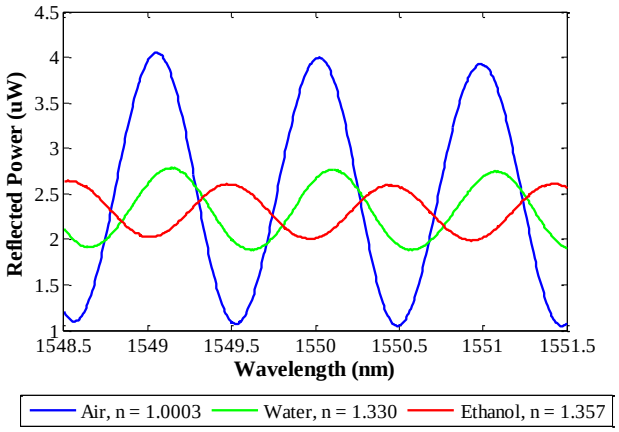


Figure 7. Reflected power (μW) measured for probe during 3 nm sweep in air, water, and ethanol.

The tapered air cavity, which gives rise to multiple reflections at multiple non-normal angles within the glass cavity, sets this fiber-coupled probe apart from conventional self-reference interferometers. The reflection above the critical angle provides an additional refractive index-sensitive effect, the Goos-Hänchen shift, unseen in devices with reflection only at normal angles.

IV. BULK REFRACTIVE INDEX SENSING

This section experimentally demonstrates the use of the fiber-coupled microcavity probe in bulk refractive index sensing. First, the experimental set-up for bulk refractive index measurements is described. Then, the bulk refractive index sensing performance is demonstrated with different volume fractions of ethanol and water mixtures.

A. *Experimental Set-Up*

The experimental set-up to demonstrate the interference properties of the fiber-coupled micro-sphere tip is illustrated in Fig. 8. An Ando AQ4320D Tunable Laser Source was wired to a 2×1 fiber optic coupler. The output of the coupler was sent to the SMF-28 fiber wedged in the micro-sphere tip. The reflected signal from the tip was measured by an Ando AQ6317 Optical Spectrum Analyzer wired to the second output of the coupler. Both the Tunable Laser Source and the Optical Spectrum Analyzer were connected to a PC and controlled by a LabVIEW VI. Spectra collection was also accomplished by the same program.

B. *Refractive Index Sensing with Ethanol-Water Mixture*

For a quantitative assessment of the sensitivity of this device, the probe was exposed to deionized water and then subsequently to incremental concentrations of ethanol-water mixtures ranging from 0 to 40% volume fraction of ethanol. The experiment was performed in a flow system so as to allow the injection of different mixtures without disturbing the probe or exposing it to contaminants that could be present in the air. In each mixture, a 3 nm spectral sweep was performed centered around 1550 nm with 1 mW laser power. The reflected power was measured and found to demonstrate a red spectral shift and a decrease in peak-to-peak amplitude with increasing ethanol concentration, as shown in Fig. 9. This result is in agreement with the model prediction when the refractive index of the immediate vicinity of the spherical tip, n_2, is increased.

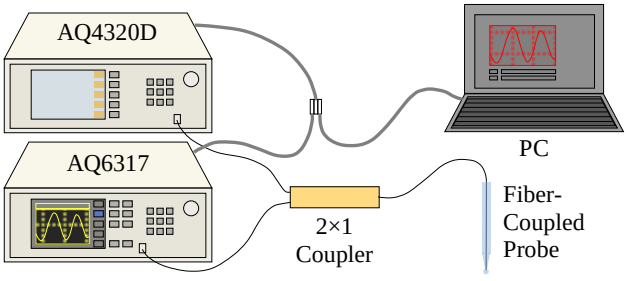


Figure 8. Experimental set-up for fiber-coupled micro-sphere tip interference measurements.

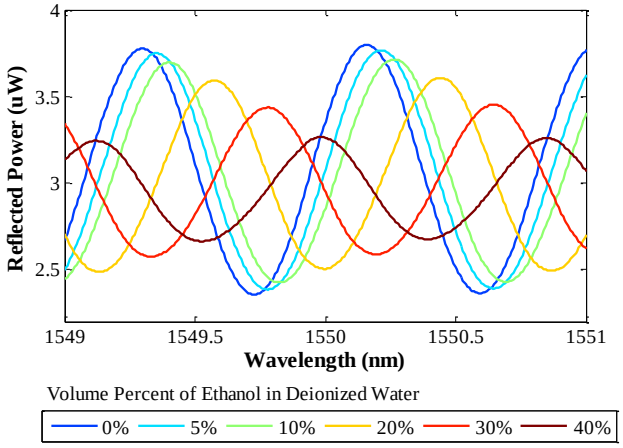


Figure 9. Red spectral shift and amplitude decrease of interference pattern as a result of exposure to 10^{-3} incremental changes in refractive index.

As a control, the second set of measurements was performed in the reverse order from higher to lower concentration of ethanol. The reflected signals were found to match those of the first set of measurements. The refractive index of each ethanol-water mixture was also measured using a Fisher Scientific Abbe Benchtop Refractometer. Fig. 10 plots the change in peak-to-peak amplitude of the spectrums and the shift in peak wavelength as a function of mixture refractive index. Both the amplitude and the phase of the spectrum exhibited a linear change with increasing refractive index. The device demonstrated the sensitivity in spectral shift of 40 nm/RIU.

The Limit of Detection (LOD) of the device was determined by performing the same measurements with smaller refractive index increments ranging from 0 to 2% ethanol volume fraction, as shown in Fig. 11. Although both the amplitude decrease and the red spectral shift were observed in the interference patterns, only the change in the peak-to-peak amplitude was resolvable by the Optical Spectrum Analyzer. Alternatively, when the amplitude change is small, the effect of the spectral shift can be magnified by measuring the reflected power at a single wavelength corresponding to the inflection point of the interference pattern. Fig. 12 plots the change in reflected power at the rising edge of the interference patterns, indicated by the dotted line in Fig. 11, and the change in peak-to-peak amplitude as a function of mixture refractive index. The device demonstrated the ability to detected minute changes in refractive index of the surrounding environment with the LOD in the range of 10^{-4} and 10^{-5} RIU. The optimization of the probe geometry such as the profile of the tapered air cavity could further enhance the LOD of the device.

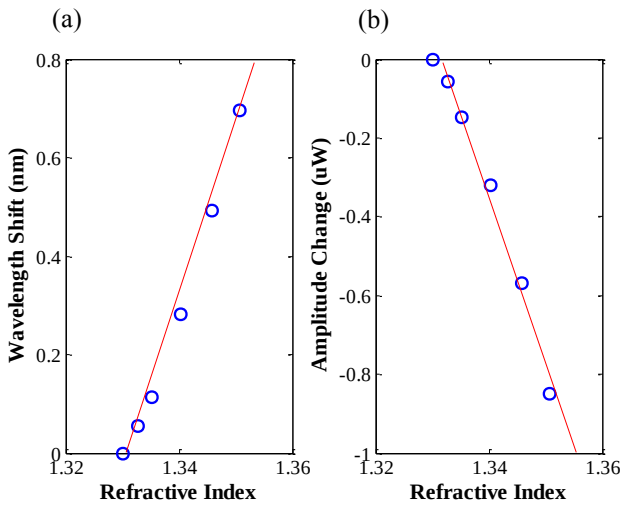


Figure 10. (a) Spectral shift, measured by the change in position of each interference peak, and (b) change in peak-to-peak amplitude of each interference pattern relative to 0% ethanol as a function of refractive index.

V. SURFACE BIOMOLECULAR INTERACTIONS SENSING

This section experimentally describes the use of the fiber-coupled microcavity probe in real-time monitoring of biomolecular interactions on the glass spherical tip. The experimental set-up described in Section IV applies. The reagents for the surface reactions and the experimental procedures are described, followed by the interpretation of the experimental results. Lastly, the proof-of-concept for the device application in concentration analysis is demonstrated.

A. Real-Time Monitoring of Silanization on Probe Tip

One of the advantages of optical biosensors such as the SPR sensors over other types of biosensors is the relative ease of performing real-time monitoring of cellular environments or specific binding events [18][19]. To demonstrate the biosensing capability of the microcavity probe, the formation of self-assembled monolayer of an aminosilane coupling agent was monitored in real-time. The probe surfaces were cleaned in Piranha solution (3:1 v/v of H_2SO_4:H_2O_2) and then placed in a solution of 2% v/v of 3-aminopropyltriethoxysilane (APTES) in acetone [20]. The solution was sealed with a molded polydimethylsiloxane (PDMS) stopper and polytetrafluoroethylene (PTFE) thread seal tape to avoid evaporation. The silanization of APTES refers to the deposition of APTES molecules onto an oxide substrate surface via hydrogen bonding, covalent bonding or electrostatic interactions. It is often used to functionalize inorganic silica substrates for selective binding of organic bio-targets [21]. The probe was incubated in the solution for 12 hours as continuous 3 nm spectral sweeps, centered around 1550 nm with 1 mW laser power, were taken. The interference patterns obtained during the 12-hour APTES silanization exhibit blue spectral shift, as shown in Fig. 13. The change in reflected power at a single wavelength corresponding to the dotted line is also plotted as a function of incubation time.

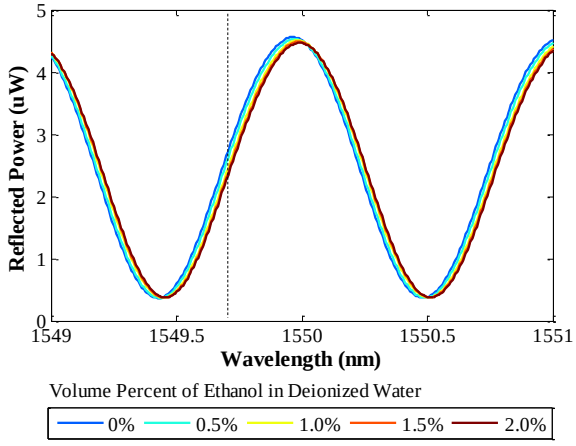


Figure 11. Red spectral shift and amplitude decrease of interference pattern as a result of exposure to 10^{-5} incremental changes in refractive index.

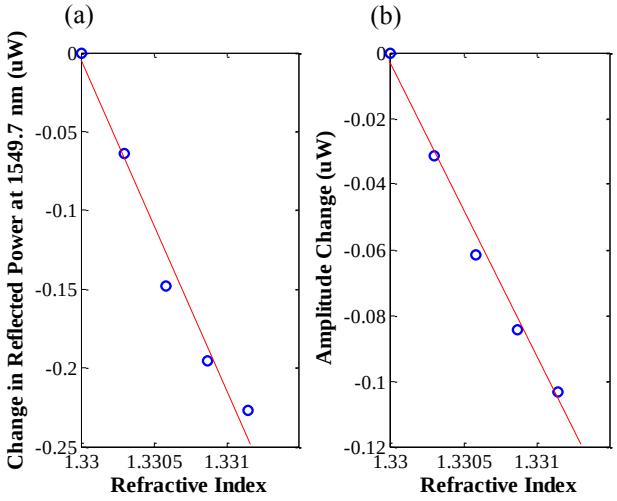


Figure 12. (a) Change in reflected power at 1549.7 nm, and (b) change in peak-to-peak amplitude of each interference pattern relative to 0% ethanol as a function of refractive index.

The change in reflected power at the rising edge illustrates the self-assembly of the APTES monolayer in the first 25-30 minutes, and subsequent intermolecular interactions of APTES molecules on top of this monolayer in the 12-hour period after the monolayer formation. The total spectral shift of approximately 0.2 nm was observed as a result. This observed monolayer binding time-scale (20-30 minutes) agrees with the standard protocol time-scale for a 5 Å monolayer deposition of APTES on a silica substrate from organic solvents [21].

B. Interpreting Blue Spectral Shift

It may be noted that the blue spectral shift as a result of molecular binding on the sensing surface is counter to the

red spectral shifts observed in typical resonance sensors, caused by an effective lengthening of the optical path due to the presence of the monolayer [22][23][24]. This counter-intuitive effect can be explained by the fundamentally different working principle of the fiber-coupled probe, which behaves like a self-reference interferometer, compared to those of the typical resonance sensors.

During the self-assembly of APTES molecules onto the spherical tip, the layer of surface-bounded molecules could be modeled as the third effective cavity, in addition to the air and the glass cavities discussed in Section III. The presence of this effective layer results in an additional effective reflection interface R_3 between the APTES layer and the bulk solution, as shown in Fig. 14. Thus, the total reflected signal is the product of the interference of three electromagnetic waves: $\mathbf{E_1}$, $\mathbf{E_2}$ and $\mathbf{E_3}$. The blue spectral shift could be caused by a lessening of optical tunneling into the bulk solution due to the higher refractive index difference $\Delta n = n_2 - n_3$ between the APTES layer and the bulk solution as more APTES molecules deposited onto the glass spherical tip.

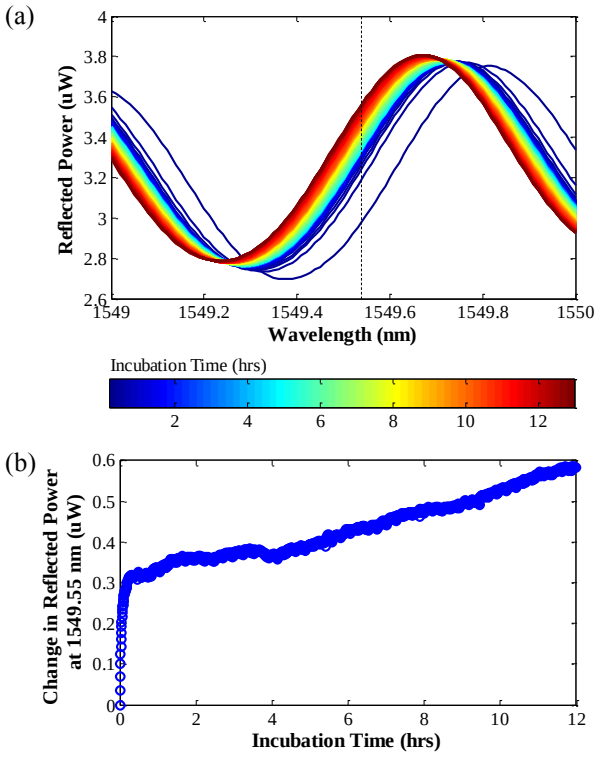


Figure 13. (a) Spectral shift of interference patterns and (b) change in reflected power at 1549.55 nm (dotted line in (a)) as a result of 12 hours incubation in APTES silanization solution.

Fig. 15 shows the predicted interference patterns during the self-assemble of APTES monolayer generated from (2), modified to include the third reflection surface. The refractive index and the thickness of the APTES monolayer were taken to be 1.46 and 5 Å, respectively [21]. The refractive index n_3 of the APTES in acetone solution was measured by the refractometer and assumed to remain constant during the 30 minutes incubation period. The refractive index n_2 was scaled from n_3, when the surface was free of APTES molecules, to 1.46, when the monolayer formation was completed, over the course of 30 minutes. The experimentally obtained interference patterns during the first 30 minutes of incubation in 2% v/v solution of APTES in acetone is also shown for comparison.

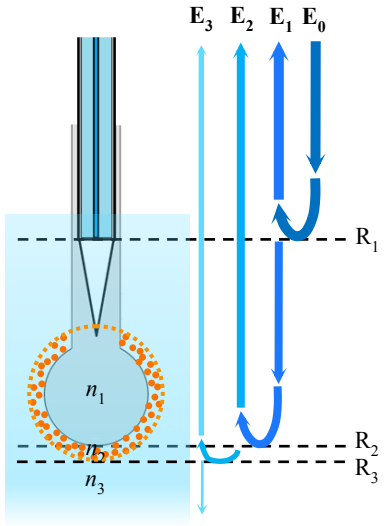


Figure 14. Light propagation in 3-layer Fabry-Perot cavity model with 3 reflection interfaces: R_1 between the cleaved end of the fiber and air, R_2 between the glass tip and surface-bounded molecules, and R_3 between the surface-bounded molecules and bulk environment.

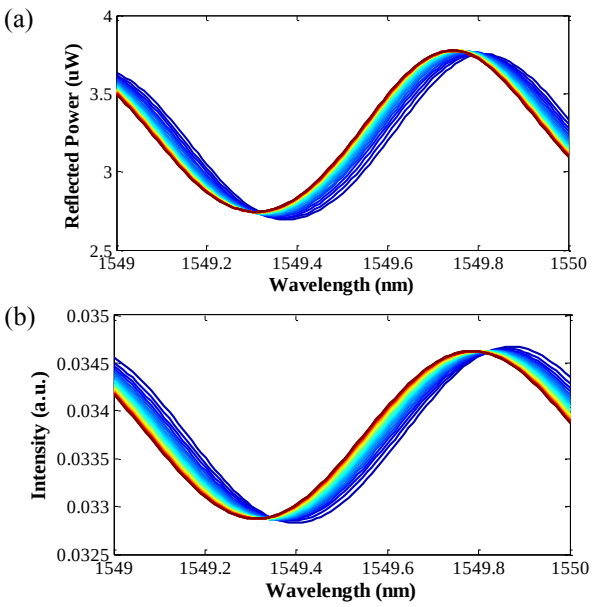


Figure 15. (a) Experimentally obtained spectral shift of interference patterns as a result of 30 minutes APTES monolayer formation and (b) model generated prediction of the same process.

The model successfully predicts the blue spectral shift of approximately the same magnitude as observed in the experimental results. Any discrepancies between the experimental results and the model prediction could be largely due to the distribution of model parameter n_2 over time, which could not perfectly replicate the actually refractive index modulations during the silanization process.

C. Concentration Analysis with Kinetic Data

The ability to perform real-time detection allows the biosensor to monitor the interactions of bio-targets with immobilized binding sites as they occur. Real-time biosensors such as the Biacore SPR modules are able to extract the kinetic constants of the given binding pair and subsequently determine the affinity and concentration of the target molecules [25][26]. To investigate the use of the fiber-coupled probe in concentration determination with kinetic data, the silanization of APTES performed at different bulk concentration ranging from 0.2 to 4% volume fraction of APTES in acetone were monitored. The probe was cleaned in Piranha solution prior to incubation in each silanization solution. The probe was incubated for 30 minutes in each solution, during which continuous 3 nm spectral sweeps, centered on 1550 nm with 1 mW laser power, were taken. The change in reflected power at a single wavelength corresponding to the rising edge of the interference patterns as a function of time is shown in Fig. 16. The change in reflected power was normalized to the value previously obtained from the 2% solution at the 30-minute mark, where the monolayer was assumed to have reached saturation.

The time profile of the normalized change in reflected power is fitted to the adsorption model based on the Langmuir-type physico-chemical reversible process [27]. The model consists of three Langmuir parameters: the adsorption capacity, Γ_{max}, defined as the ratio of the number of bounded molecules to the total number of binding sites at saturation, the association rate constant, k_a, and the dissociation rate constant, k_d; together they define both the reaction kinetics and the equilibrium adsorption isotherm of the process. Equation (3) describes the differential rate of adsorption based on the Langmuir adsorption model, where $\Gamma(t)$ is the time dependent ratio of number of bounded molecules to the total number of binding sites and C_0 describe the bulk concentration of the target molecules. The three Langmuir parameters were determined from the experimental data graphically as described in [27]. Solving (3) for $\Gamma(t)$, the model was fitted to the experimental data with C_0 as the only changing variable for different APTES concentrations.

$$\frac{d\Gamma(t)}{dt} = k_a C_0 \left(1 - \frac{\Gamma(t)}{\Gamma_{max}}\right) - k_d \Gamma(t) \qquad (3)$$

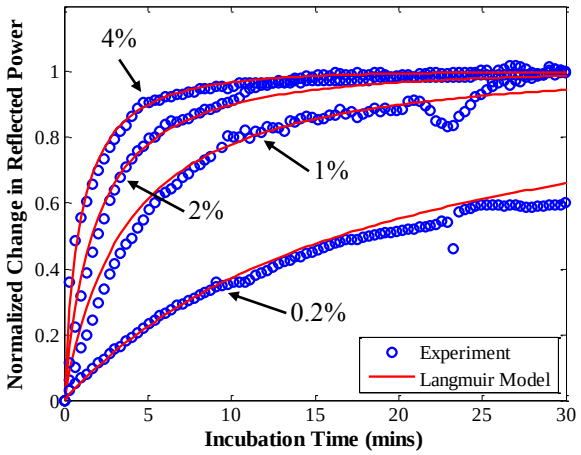


Figure 16. Experimentally obtained spectral shift of interference patterns as a result of 30 minutes APTES monolayer formation (blue circles) and model generated data-fitting of the same process (red lines).

It should be noted that although these parameters are unique for a given chemical reaction, solution chemistry, and temperature, the comparison with previously reported values for the silanization of APTES is difficult. It is well established that the silanization of APTES on silicon oxide (SiO_2) substrate is highly sensitive to the deposition conditions and outside environment, thus, the specific physico-chemical processes that occur during silanization vary significantly from experiments to experiment [28][29]. As shown in Fig. 16, the rate of deposition of APTES molecules onto the glass spherical tip based on the Langmuir kinetics was able to model the time profile of the spectral shift of the interference patterns during APTES monolayer formation from different bulk concentration.

The results thus far demonstrate that the fiber-coupled probe could be used in real-time binding assays to extract kinetic information of the given interactions. While the total spectral shift at equilibrium might have a relatively low sensitivity to the change in bulk concentration at higher concentration range due to the system approaching saturation, the initial slope of the spectral shift as a function of time could be used to resolve minute concentration differences. Furthermore, a more rapid assay could be achieved by utilizing only the initial rate from the real-time monitoring as oppose to employing the more conventional equilibrium level value in the point sample detection. Moreover, the spectral shift as a function of incubation time was taken from the reflected power at a single wavelength. The detection could theoretically be done with a single-wavelength light source without any needs for frequency sweeping, thus, lowering the total cost of the analysis procedure dramatically.

For the sample of known concentration, the kinetic constants Γ_{max}, k_a and k_d of the interactions can be determined. This process is an essential step in the development of new therapeutic agents as the specificity

and affinity of drugs to targeted species will greatly affect the efficacy of the treatment [10]. On the other hand, for the analysis of a certain interaction such as specific binding of antigen to its antibody pair, the concentration of the target molecules can be derived from the real-time binding data by fitting the results to known adsorption kinetics. In the case where both the transport phenomena and the reaction kinetics are well-established, such as the binding of a specific antigen to its antibody pair under a controlled flow in a microfluidic cell, the kinetic data obtained in real-time may offer a new platform of rapid CFCA [14][15]. Such analysis could make the fiber-coupled probe, given its low-cost, robust, and easy-to-fabricate nature, a promising alternatives to point-of-care diagnostics.

VI. Conclusion and Future Work

Demonstrating the mode coupling between an optical fiber and a microcavity sphere works to bring the high sensitivity of optical microcavity sensors off of the optical table and into a robust integrated microprobe [30]. The integrated microprobe design has the distinct advantage of use in vivo without the need of microfluidics or precise optical alignment of evanescent coupling, and can be implemented for sample volumes on the order of several hundred pico-liters to nano-liters, essentially a wavelength thick layer of sample surrounding the spherical tip.

The fiber-coupled microcavity probe operates on the bases of a self-reference interferometer with multiple reflections at multiple non-normal incident angles. The light source enters the probe tip via the optical fiber wedged into the finely tapered cavity inside glass spherical tip. The wave fronts scattered from the fiber tip are bent by the unique geometry of the tapered cavity, allowing the light to propagate within the spherical tip in a similar manner as a weak Q coupling of light into a spherical microcavity resonator. The multiple reflection angles allow the reflected interference signals from the probe to undergo both amplitude and phase modulation. This feature differentiates the microprobe from other optical fiber-based biosensors with similar needle-like device architecture, which only derive information from normal reflection at the flat fiber tip [31]. After multiple reflections within the glass cavity, the light eventually reenters the air cavity and couples back into the optical fiber, where it interferes with the back reflection from the flat end of the fiber. The probe model based on this principle of the effective Fabry-Perot cavity was able to describe the interference patterns of the reflected signal remarkably well. Even in the more complex cases of molecular interactions on the probe surface, modeling the layer of surface-bounded molecules as the third effective cavity with high refractive index could reproduce the seemingly counter-intuitive blue spectral shift observed during binding experiments.

The first generation pre-optimization microprobes demonstrated sensitivity in the range of 10^{-4} to 10^{-5} RIU and the 40 nm/RIU under bulk refractive index measurements. In addition, monitoring of the spectral shift during the salinization process confirms a high degree of near-field (evanescent) sensitivity to the deposition of molecules onto the probe surface as the formation of the 5 Å self-assembled APTES monolayer was detected in real-time. Performing the silanization procedure at different concentration of APTES in the solution demonstrated that kinetics of the interactions may be extracted from the real-time data. The time profile of the spectral shift was shown to reflect the binding rate of APTES molecules as the data fitted the Langmuir physico-chemical adsorption kinetics over the range of concentrations tested. In addition, utilizing the initial rate of spectral shift from the real-time monitoring could resolve much smaller concentration differences than the more conventional equilibrium data from point sample detection. Thus, the experimental results demonstrate the proof-of-concept that this fiber-coupled microcavity probe can be used for detection of specific bio-targets in the evanescence near-field zone of the glass cavity, and even has potential use in studying real-time surface interaction kinetics and perform CFCA for point-of-care diagnostics.

Although the specific wave characteristics of reflected light from the first generation prototypes varies from device to device due to the non-uniform probe geometry, the implementation of an automated fabrication platform with mechanical stages could control the size and curvature of the tapered air cavity and the glass spherical tip. Furthermore, though exhibiting sensitivity in bulk solution less than that achieved in other optical biosensors such as Surface Plasmon Resonance sensors, ring resonator sensors and interferometric sensors, which are able to detect down to 10^{-7} RIU, the data recorded thus far is from proof-of-concept hand-made microprobes that are yet to undergo optimization as far as their cavity size and geometric effects on mode coupling [12]. For example, by fabricating smaller diameter spherical tips, we will be able to increase the nm/RIU limit of detection due to a shorter effective cavity length. Similarly, implementing the controlled automated fabrication platform, we hope to optimize the geometry of the tapered air cavity and probe curvatures. This optimization could lead to enhanced mode coupling between the fiber and the spherical tip by bending the incoming wave fronts such that the reflections in the spherical tip occur at the optimal angles for maximum coupling Q-factor. In addition, the geometric variations of the tapered air cavity and probe curvature could allow for tunability between the competing effects of the reflection coefficient modulation and Goos-Hänchen phase shift for the creation of far field and near field sensors, respectively.

Acknowledgment

The authors wish to thank Anubhav Tripathi and Domenico Pacifici for their support and helpful discussion for this study. The authors are also very grateful for the guidance of Jin Ho Kim, Gustavo Fernandes, and Carlos Bledt, and the support of Center for Biomedical Engineering

at Brown University, ARL, AFOSR, CR Bard, and WCU program at Seoul National University.

REFERENCES

[1] Z. Ballard, N. Leartprapun, and J. Xu, "Fiber-coupled microcavity probe for in-vivo near-field sensing," *The Fourth International Conference on Sensor Device Technologies and Applications* (SENSORDEVICE 2013) IARIA, Aug. 2013, pp. 107-111, ISSN: 2308-3514, ISBN: 978-1-61208-297-4.

[2] M. A. Cooper, "Label-free screening of bio-molecular interactions,"*Anal. Bioanal. Chem.*, vol. 377, no. 5, pp. 834-842, 2003, doi: 10.1007/s00216-003-2111-y.

[3] S.O. Kasap, Optoelectronics and photonics, principles and practices, Prentice-Hall, pp. 1-42, 2001, ISBN: 0-201-61087-6.

[4] X. Zhang, C.R. Yonzon, M.A. Young, D.A. Stuart and R.P. Van Duyne, "Surface-enhanced Raman spectroscopy biosensors: excitation spectroscopy for optimisation of substrates fabricated by nanosphere lithography," *IEE Proc.-Nanobiotechnol.*, vol. 152, no. 6, pp. 195-206, 2005.

[5] B. Yu, A. Wang, and G. R. Pickrell, "Analysis of fiber Fabry–Pérot interferometric sensors using low-coherence light sources," *J. Lightwave Technol.*, vol. 24, no. 4, pp. 1758-1767, 2006, doi: 10.1109/JLT.2005.863336.

[6] H. N. Daghestani and B. W. Day, "Theory and applications of surface plasmon resonance, resonant mirror, resonant waveguide grating, and dual polarization interferometry biosensors," *Sensors*, vol. 10, pp. 9630-9646, 2010, doi: 10.3390/s101109630.

[7] F. Vollmer and L. Yang, "Label-free detection with high-Q microcavities: a review of biosensing mechanisms for integrated devices," *Nanophotonics*, vol. 1, pp. 267-291, 2012, doi: 10.1515/nanoph-2012-0021.

[8] T. Yoshie, L. Tang, and S. Y. Su, "Optical microcavity: sensing down to single molecules and atoms," *Sensors*, vol. 11, pp. 1972-1991, 2011, doi: 10.3390/s110201972.

[9] M. N. Velasco-Garcia, "Optical biosensors for probing at the cellular level: A review of recent progress and future prospects," *Seminars in cell & developmental biology*, vol. 20, no. 1. Academic Press, pp. 27-33, Feb. 2009, doi: 10.1016/j.semcdb.2009.01.013.

[10] Y. Fang, "Label-free optical biosensors in drug discovery," *Trends in Bio/Pharmaceutical Industry*, vol. 3, pp. 34-38, 2007.

[11] F. S. Ligler, "Perspective on optical biosensors and integrated sensor systems," *Anal. Chem.*, vol. 81, no. 2, pp. 519-526, 2009, doi: 10.1021/ac8016289.

[12] X. Fan, I. M. White, S. I. Shopova, H. Zhu, J. D. Suter, and Y. Sun, "Sensitive optical biosensors for unlabeled targets: A review," *Anal. Chim. ACTA*, vol. 620, no. 1, pp. 8-26, 2008, doi: 10.1016/j.aca.2008.05.022.

[13] C. Bee, Y. N. Abdiche, J. Pons, and A. Rajpal, "Determining the binding affinity of therapeutic monoclonal antibodies towards their native unpurified antigens in human serum," *PLoS ONE*, vol. 8, no. 11: e80501, pp. 1-13, 2013, doi: 10.1371/journal.pone.0080501.

[14] K. Sigmundsson, G. Másson, R. Rice, N. Beauchemin, and B. Öbrink, "Determination of active concentrations and association and dissociation rate constants of interacting biomolecules: an analytical solution to the theory for kinetic and mass transport limitations in biosensor technology and its experimental verification," *Biochemistry-US.*, vol. 41, no. 26, pp. 8263-8276, 2002, doi: 10.1021/bi020099h.

[15] S. Rodriguez-Mozaz, M. J. Lopez de Alda, M. Marco, and D. Barcelo, "Biosensors for environmental monitoring: A global perspective," *Talanta*, vol. 65, no. 2, pp. 291-297, 2005, doi: 10.1016/j.talanta.2004.07.006.

[16] O. R. Ranjbara, et al., "High pressure discrimination based on optical fiber microsphere cavity Fizeau interferometer," *Proc. of SPIE*, vol. *8421*, 2012, doi: 10.1117/12.966322.

[17] D. Q. Chowdhury, D. H. Leach, and R. K. Chang, "Effect of the Goos-Häinchen shift on the geometrical-optics model for spherical-cavity mode spacing," *J. Opt. Soc. Am. A*, vol. 11, no. 3, pp. 1110-1116, 1994, doi: 10.1364/JOSAA.11.001110.

[18] M. A. Cooper, "Optical biosensors in drug discovery," *Nat. Rev. Drug Discov.*, vol. 1, pp. 515-528, 2012, doi: 10.1038/nrd838.

[19] B. John, "Optical biosensors for unlabelled bio-molecules detection: future development trends," *Third National Conference on Modern Trends in Electronic Communication & Signal Processing*, 2013.

[20] A. V. Krasnoslobodtsev and S. N. Smirnov, "Effect of water on silanization of silica by trimethoxysilanes," *Langmuir*, vol. 18, no. 8, pp. 3181-3184, 2002, doi: 10.1021/la015628h.

[21] M. Zhu, M. Z. Lerum, and W. Chen, "How to prepare reproducible, homogeneous, and hydrolytically stable aminosilane-derived layers on silica,"*Langmuir*, vol. 28, no. 1, pp. 416-423, 2012, doi: 10.1021/la203638g.

[22] F. Vollmer and S. Arnold, "Whispering-gallery-mode biosensing: label-free detection down to single molecules," *Nat. Methods*, vol. 5, no. 7, pp. 591-596, 2008, doi: :10.1038/NMETH.1221.

[23] K. M. De Vos, I. Bartolozzi, P. Bienstman, R. Baets, and E. Schacht, "Optical biosensor based on silicon-on-insulator microring cavities for specific protein binding detection-art. no. 64470K," *P. Soc. Photo-Opt. Ins. IV*, vol. 6447, pp. 64470K1-64470K8, 2007, doi: 10.1117/12.698875.

[24] Y. Guo, et al., "Label-free biosensing using a photonic crystal structure in a total-internal-reflection geometry," *SPIE BiOS: Biomedical Optics*, vol. 7188, pp. 71880B–71880B12, 2009, doi: 10.1117/12.808369.

[25] M. Ritzefeld and N. Sewald, "Real-time analysis of specific protein-DNA interactions with Surface Plasmon Resonance," *J. Amino Acids*, vol. 2012, pp. 1-19, 2012, doi: 10.1155/2012/816032.

[26] E. S. Daya, A. D. Capilia, C. W. Borysenko, M. Zafaria and A. Whitty, "Determining the affinity and stoichiometry of interactions between unmodified proteins in solution using Biacore," *Anal. Biochem.*, vol. 440, no. 1, pp. 96-107, 2013, doi: 10.1016/j.ab.2013.05.012.

[27] A. Islam, M. R. Kahn and S. I. Mozumber, "Adsorption equilibrium and adsorption kinetics: a unified approach," *Chem. Enq. Technol.*, vol. 27, no. 10, pp. 1095-1098, 2004, doi: 10.1002/ceat.200402084.

[28] N. Aissaoui, L. Bergaoui,J. Landoulsi, J. F. Lambert, and S. Boujday, "Silane layers on silicon surfaces: mechanism of interaction, stability, and influence on protein adsorption," *Lamgmuir*, vol. 28, no. 1, pp. 656-665, 2012, doi: 10.1021/la2036778.

[29] E. T. Vandenberg, et al., "Structure of 3-aminopropyl triethoxy silane on silicon oxide," *J. Colloid Interf. Sci.*, vol. 147, no. 1, pp. 103-118, 1991, doi: 10.1016/0021-9797(91)90139-Y.

[30] S. Arnold, S. I. Shopova, and S. Holler, "Whispering gallery mode bio-sensor for label-free detection of single molecules: thermo-optic vs. reactive mechanism," *Opt. Express*, vol. 18, no. 1, pp. 281-287, 2010, doi: 10.1364/OE.18.000281.

[31] D. Dey, T. Godswami, "Optical Biosensors: A Revolution Towards Quantum Nanoscale Electronics Device Fabrication," *J. Biomed. Biotechnol.*, vol. 2011, pp. 1-7, 2011, doi: 10.1155/2011/348218.

CMOS Readout Circuit with Wide Dynamic Range for an UV-NIR Silicon Sensor

Emmanuel Gómez-Ramírez, A. Díaz-Méndez,
Mariano Aceves Mijares, José Miguel Rocha Pérez,
and Jorge Miguel Pedraza Chávez
INAOE
Puebla, Mexico
e-mail: {emmanuelgomez, ajdiaz, maceves, jmr,
jpch}@inaoep.mx

Carlos Domínguez Horna, Ángel Merlos
IMB-CNM (CSIC)
Barcelona, Spain
e-mail: carlos.dominguez@imb-cnm.csic.es

***Abstract*— Currently, a CMOS imager capable of detecting from Ultraviolet-to-Near Infrared (UV to NIR) light is desirable. A new silicon sensor that detects from UV to NIR to be used for CMOS imaging was developed. However, the range of photo current generated by this sensor is wide. Then, there is a need to develop CMOS circuits with a wide dynamic range to be used with this sensor, or any other with wide output signal. This paper describes a CMOS readout circuit for applications in UV-NIR imaging with sensors that generate current in many orders of magnitude. The developed UV-NIR sensor is compatible with a CMOS technology. A new topology of a wide dynamic range readout circuit using a multimode sensing technique is proposed. Also, the design, computer simulation, and experimental corroboration are shown. It is demonstrated that automatic switching between different modes is achieved, and then a dynamic range up to 160 dB can be obtained.**

Keywords-UV sensor; silicon sensor; smart-sensor; wide dynamic range; CMOS imagers; PWM sensors; continuously operating sensors; multimode sensing.

I. INTRODUCTION

Complementary Metal Oxide Semiconductor (CMOS) imagers capable of detecting from Ultraviolet-to-Near Infrared (UV to NIR) light is an area of research among the circuits and systems community. In our previous work [1], the design of an integrated circuit with a wide dynamic range was presented; this paper extends our results to show experimental feasibility.

Currently, CMOS technology allows the integration of complex electronic systems that include devices such as optical sensors. The integration of CMOS readout electronics and sensors allow the formation of CMOS imagers that offer several advantages compared to Charge-Coupled Device (CCD's) [2]. These include small power consumption, low voltage, low cost, etc., and have enabled the creation of imagers that represent single-chip solutions [2].

A CMOS imager capable of detecting radiation from UV to NIR is desirable for many applications in different areas, but these types of imagers have been difficult to implement due in part to the fact that silicon sensors have a reduced sensitivity in the UV light range when compared with the visible and NIR range.

To increase the response in the lower wavelength range it is necessary to use technologies that normally are not compatible with the CMOS technology, which generates difficulties for integration and extra cost.

A silicon detector sensible to UV light is reported in [3], which is compatible with CMOS technology [4]. This sensor increases the UV response, but it is necessary to have a readout system capable to respond in a wide range of currents, that is, the dynamic range has to be wide.

In order to have an integrated circuit that includes both the CMOS readout electronic functions and the mentioned sensor, two integrated circuits were designed. One of them has a CMOS 0.5 micrometer (μm) minimum feature with a new technique to improve the dynamic range and to face the problem of detecting currents from nanoamperes (nA) to microamperes (μA). The other one is done with 2.5 μm minimum feature CMOS technology. This technology is versatile enough to allow extra process steps to include the sensor fabrication.

Currently, both circuits are under fabrication. Moreover, it has been demonstrated that the sensor and the circuit can be built in the same process, 2.5 μm CMOS technology, with both conserving their characteristics [4]. On the order hand the design of the circuit with 0.5 μm technology was presented in [1].

In this paper, the design details and the experimental implementation of a CMOS pixel using the UV-NIR sensor and its acquisition circuitry with increased dynamic range to measure a wide range of currents are presented. The paper is divided into two sections; the first part corresponds to design details and the simulation of the integrated CMOS circuit with the 0.5 μm technology. The second part corresponds to the experimental section obtained using a discrete CMOS transistors implemented with Advanced Linear Devices (ALDs), which is a technique normally used to confirm the circuit design in a very economical way.

This paper is organized as follows: Section II presents the definition and basic assumptions of the multimode technique proposed and the design of the readout circuit. Sections III and IV present the results obtained in the simulation part and the experimental procedure of the proposed circuit, respectively. Section V summarizes the study.

II. PWM AND DIRECT MODE METHOD

In a CMOS imager, one of the most important figures of merit is the Dynamic Range (DR) [2] given by (1). The DR permits the differentiation between high and low excitation conditions in general, and, in the case of an image, between lightly and highly illuminated conditions.

It is clear that systems with higher DR would resolve extreme conditions without the loss of information.

$$DR = 20log(V_{max}/V_{min}) \text{ [dB]}. \quad (1)$$

Many efforts have been made to obtain vision systems with wide DR. Different techniques to increase the DR have been reported; some of them are reviewed in [5-6] and are listed here: logarithmic method, clipping sensors, multimode sensors, frequency-based sensors, time-to-saturation sensors, multiple sampling methods, and multiple integration time methods.

As mentioned above, one technique to increase the DR is the multimode sensor. Multiple sensing techniques can be used in only one system to enhance the DR, by taking advantage of the best resolution of each one.

In this work, a multimode technique is used to increase the DR, with the combination of integration and direct mode detection methods.

In the first case (i.e., integration-based sensor or integration mode), photogenerated charges are integrated to produce a linear response. Using this method, current in the order of femtoamperes (fA) can be detected [7], corresponding to very small illumination intensities.

In the direct mode operated sensor technique, the current generated by a Photodiode (PD) is directly transferred to the measurement system. In this case, the current detected is in the order of nA to miliamperes (mA), corresponding to very high illumination intensities. Then, a combination of both readout techniques allows detecting the current of a PD from fA to mA.

A block diagram of the multimode technique used is shown in Fig. 1**.** As can be seen, the photocurrent from the sensor could be integrated in the upper branch (steps 1 to 3) or it could be amplified directly to the lower branch (steps 1 and 4).

In the integration mode, the photocurrent, I_{ph}, obtained in the step 1 (with the switch activated in the up position) is integrated and converted to a voltage ramp, step 2. In step 3, when the voltage ramp reaches the comparator reference voltage a shot pulse is obtained and this pulse controls the switch. In the direct mode, step 4, the I_{ph} is directly fed to an amplifier. Both readout signals are passed into a counter and an analog-to-digital converter respectively, and finally, in step 5, a Digital Word (DW) proportional to the incident light in the sensor is obtained.

Both techniques are explained with more details in next sections.

A. Pulse Width Modulation Mode Readout

When a sensor works in the integration mode, the parasitic capacitor of a PD is charged to a reference potential; when light shines on it, this potential decreases almost in a linear fashion, due to the photocurrent discharge of the capacitor.

By measuring the voltage drop, the amount of light received can be obtained, using (2), where I_{ph} is the photocurrent, T_{int} is the integration time, and C_{PD} is the parasitic capacitance of the PD.

$$\Delta V = I_{ph} * T_{int}/C_{PD}. \quad (2)$$

As it can be seen, ΔV is directly depend of T_{int}, so, it is necessary, a fine control over T_{int} to obtain an suitable output signal. To solve this, in Pulse Width Modulation (PWM) [8] mode, the current generated by each PD defines the integration time, Tint, so that very small are integrated until an adequate output is obtained.

In favor of greater clarity, in Fig. 1, integration mode is represented by steps 1 and 2, and PWM mode by steps 1 to 3. So, hereafter, the first part of the system (from steps 1 to 3) is referred as PWM mode and the second part (steps 1 and 4) as direct mode.

The circuit shown in Fig. 2 is an example of how to implement the PWM mode. The current is integrated and then compared with the reference voltage.

The P-type Metal Oxide Semiconductor (PMOS) transistor called Mrst in Fig. 2, works as a reset switch; in the ON state the PD voltage (V_{pd}) is near V_{dd}. If the switch is turned OFF, the incident light produces V_{pd} to decrease linearly.

When V_{pd} reaches the reference voltage, the comparator generates an output voltage pulse. Measuring the width of this pulse, the amount of light shining on the PD can be estimated.

The problem with this technique is that have limited DR. At higher levels of photocurrent, the integration time would be too short and in some cases imperceptible, restraining this method to low currents only. This technique works very well at small light power levels but fails to work adequately with high power levels, as shown in Fig. 3. In other words, a low illumination level produces a small photocurrent, as the light intensity is augmented the photocurrent increases reducing the integration time. If the light is intense, the integration time will be very difficult to measure.

To solve the high current problem bright light would be detected by direct mode technique.

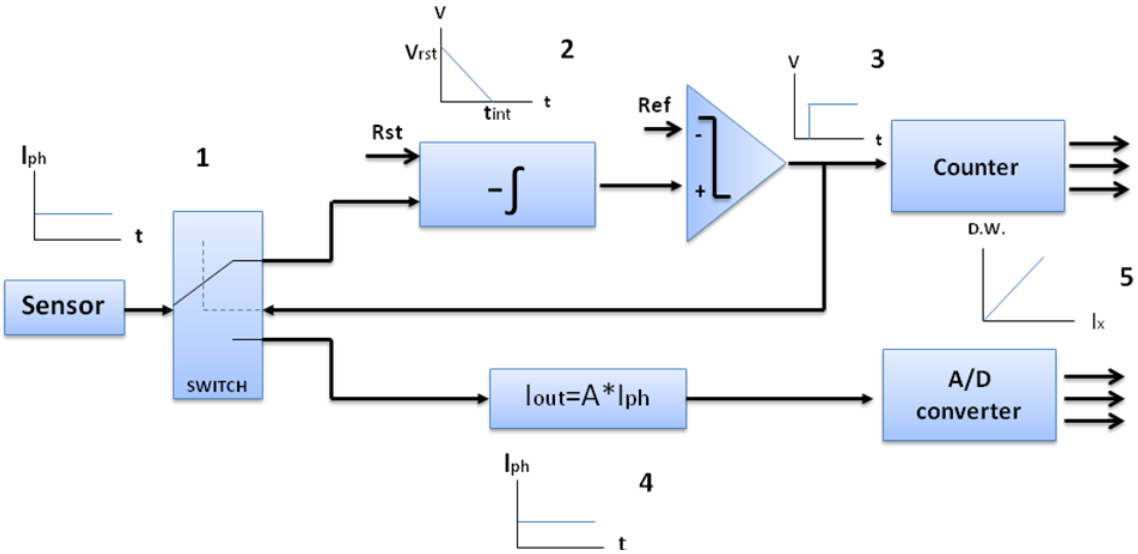


Figure 1. Multimode sensing block diagram.

B. *Direct Mode Readout*

In the direct output mode (or direct mode), the PD current is directly transferred to current mirrors with or without amplification. The disadvantage with this architecture is that suffers from low sensitivity at low levels of illumination; however, it works fine detecting high illumination levels.

Flipped-Voltage Follower (FVF) in current mode [9] is used to sense the PD current due to its very low input impedance and because it can drive large input current variations, as shown in Fig. 4. The FVF is shown in Fig. 4a; in this circuit, the input impedance Z_{in} is very low and given by

$$Z_{in} = 1/(g_{m1}g_{m2}r_{01}). \quad (3)$$

where g_{m1} and g_{m2} are the transistor transconductances and r_{01} is the transistor's output resistance in M1.

The input impedance is low due to the shunt feedback provided by M2. The output current is given by the expression $I_{out} = I_{ph} - I_{bias}$, where I_{bias} is the bias current provided by the current source in M1 and I_{ph} is the current from PD.

Normally, the current needs to be amplified so a current mirror is used and the gain is given by the ratio W/L of the transistors.

To make the mirrored current as accurate as possible it is necessary to use an operational amplifier, as shown in Fig. 4b, that keeps the bias voltage equal in the drains of M2 and M3.

Fig. 5 compares the fidelity of the current copy with and without the operational amplifier. As shown, the advantage of using the operational amplifier is clear.

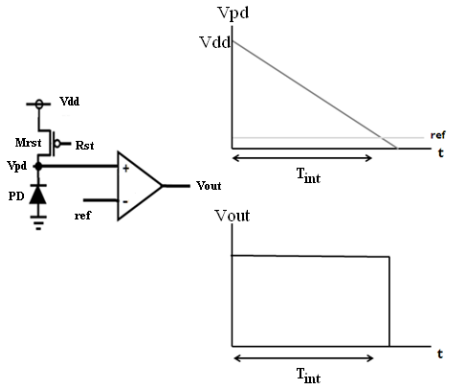


Figure 2. Circuit used to implement the PWM technique. The upper graph shows the V_{pd} voltage linear decay due to the incident light. The lower graph shows V_{out} as a function of time.

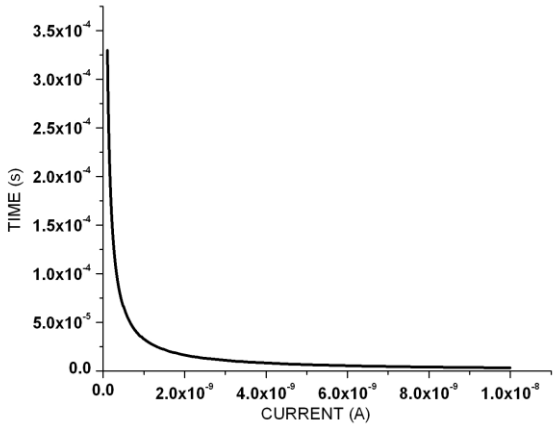


Figure 3. Photocurrent vs integration time; as the current increases the time tends toward zero.

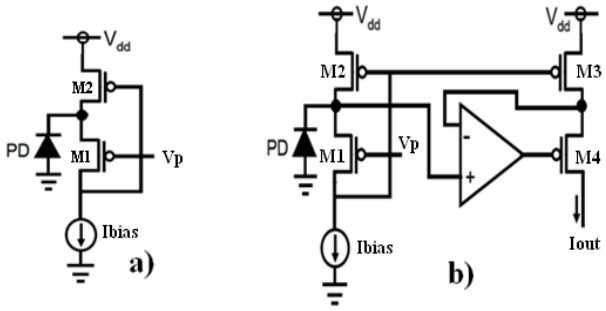


Figure 4. a) Flipped-Voltage Follower in current mode and b) with a current mirror.

C. PWM and Direct Mode Circuit Implementation

A schematic diagram of the proposed circuit is shown in Fig. 6, which combines both readout topologies: PWM and direct mode. When low illumination is shining on the PD, the PWM mode works, and when bright illumination is received, the direct mode topology comes into operation.

Internal switches that are controlled by the PWM output select one of the two topologies.

The PWM output voltage for a single sensor signal is shown in Fig. 7a; when the voltage ramp reaches the voltage reference a step is produced and the voltage gets a constant value. This voltage is then used to switch to the direct mode and the output current increases to a new value proportional to the current in the sensor, as shown in Fig. 7b.

As displayed in Figs. 2 and 6, an operational amplifier is needed in both topologies. So, sharing the operational amplifier in each pixel to run both techniques reduces the number of transistors and improves circuit performance.

III. CIRCUIT SIMULATION

Computer simulations of the circuit shown in Fig. 6 were performed, using HSPICE A-2008.03 [10]. The sensor was simulated using a simple photo-diode model, which consists of a voltage controlled current source in parallel with a capacitor and an exponential diode. The control voltage is proportional to the wavelength of the incident light in the sensor.

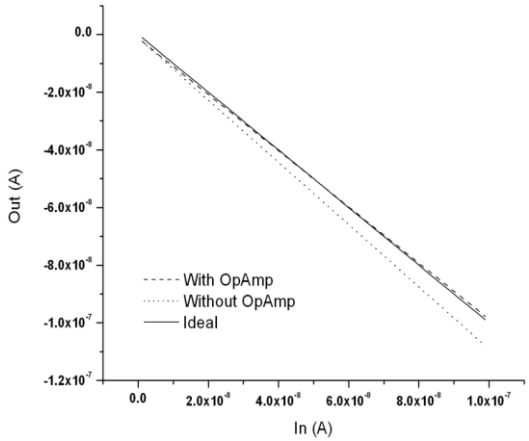


Figure 5. Input and output current comparison in the current mirror with and without the operational amplifier.

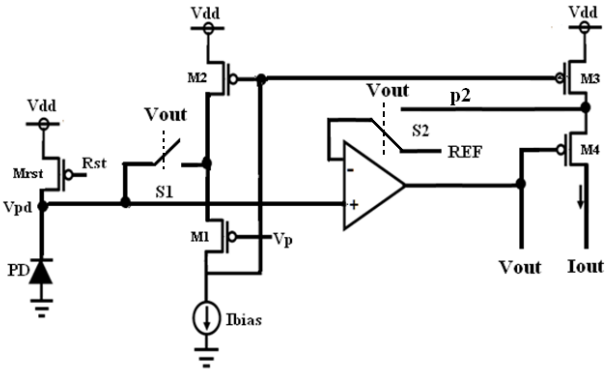


Figure 6. Dual mode circuit, when S1 and S2 are closed the direct mode is activated. When S1 is open and S2 is at the REF voltage, the PWM mode is activated.

Fig. 8 shows the results of simulations when dull incident light is cast on the sensor. As shown, the integration time starts when the reset command is triggered and V_{pd} is at V_{dd} (3.3V); then, different V_{pd} voltage ramps are produced by different light intensities. After some time, each ramp reaches the reference (in this case 1.5V). At that moment, the comparator output produces a voltage step.

This voltage step triggers the switches S1 and S2 to start the direct mode (switch S1 was off and S2 was in the REF position at the start), it is also used to bias the transistor, M4, allowing the I_{out} to increase to a value given by (I_{ph}-I_{bias}).

Fig. 8b shows the PWM output current; as can be seen, the current is zero until the voltage reference is reached then a high current is obtained. The elapsed period can be used to estimate the light intensity. In this case, the input current used was from 20 to 200 picoA (labels A and B correspond to the lower and the higher intensities, respectively).

Conversely, in Fig. 9, simulation results of the high current regime are shown.

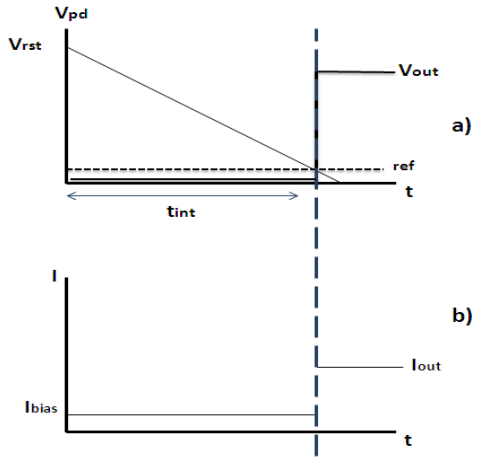


Figure 7. Output voltage and current of the PWM and direct mode sections. In a) the ramp of voltage is shown and after it reaches the reference a constant voltage step is produced. In b) as the PWM mode is active the I_{out} is low and when the direct mode is activated the I_{out} increases proportionally to the sensor's current.

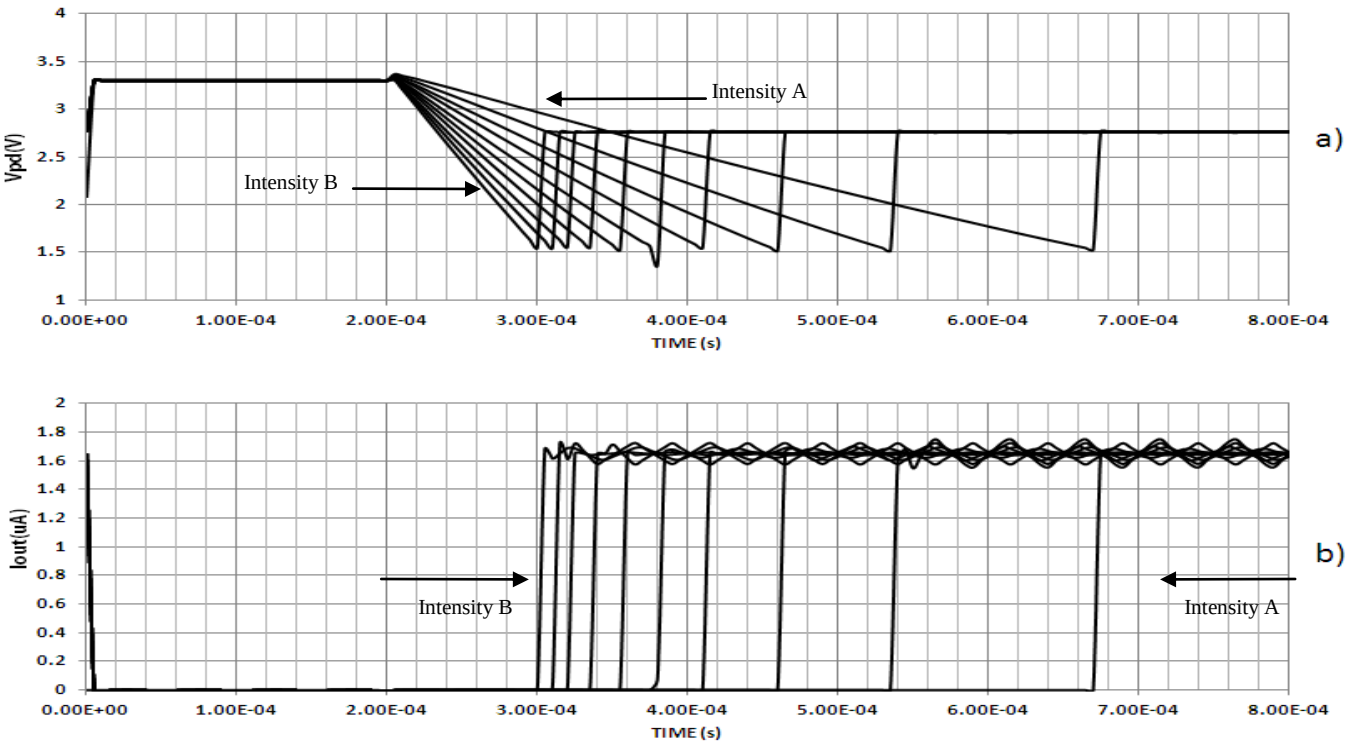


Figure 8. Simulation results of a low intensity light regimen, a) after 200us the V_{pd} linear decay starts and after reaching the reference a constant voltage is obtained; b) low intensities light sweep presents different integration time proportional to the light intensity.

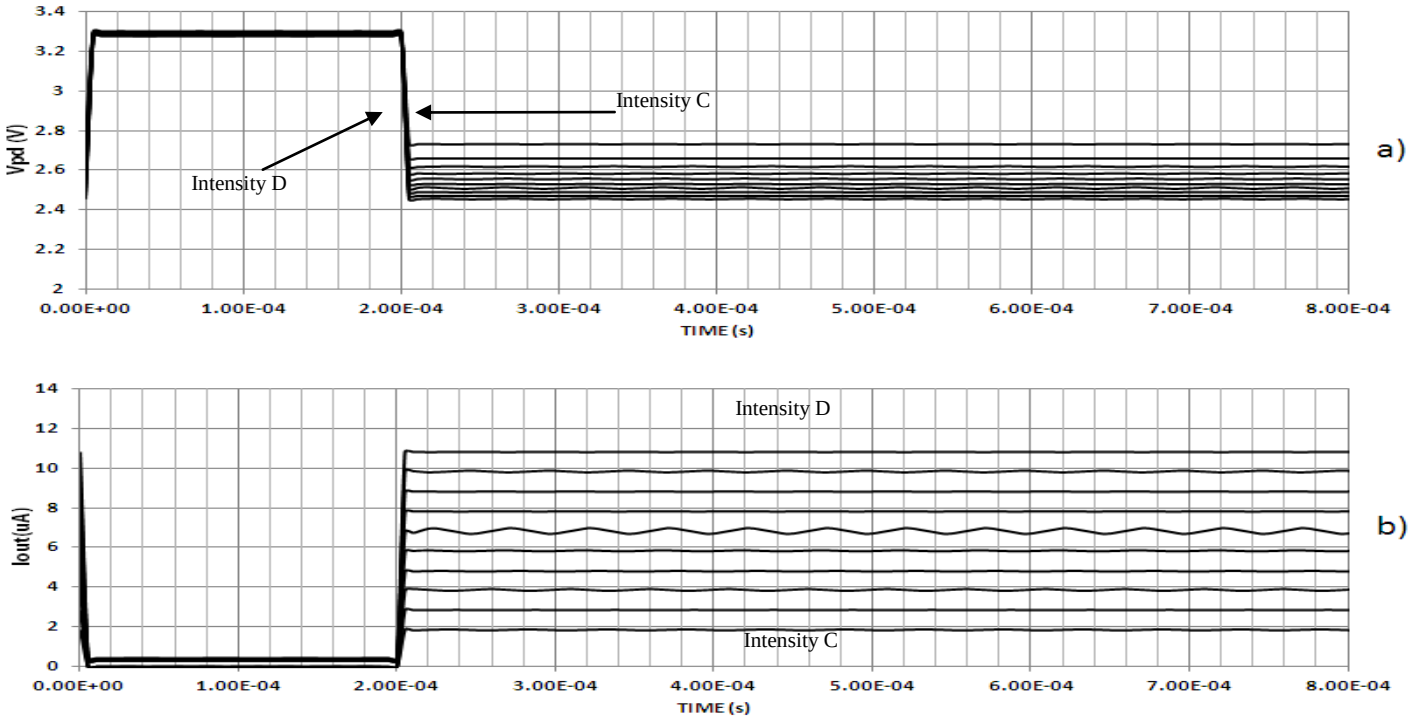


Figure 9. Simulation results of a high intensity light regimen, a) the V_{pd} linear decay is too fast and it is not differentiable due to the high current; then b) the current is measured in a straight forward manner.

In Fig. 9a, the PWM V_{out} is presented when the input currents varied from 50 to 500 nA. Labels C and D correspond to the lower and the higher intensities respectively. Comparatively in the results of Fig. 8a, after certain light intensity it is not possible to discriminate between the V_{out} ramps; intensities C and D cannot be differentiated. In this case, the output current is directly proportional to the current sensor and it is processed by the direct mode circuit. In Fig. 8b, the step of current is constant at a low value, approximately 1.6 μA, and starts at different times; on the other hand, in Fig. 9a, the current steps are variable in amplitude and practically start at the same time. In this way, switches S1 and S2 are activated to select PWM or the direct mode.

The output current, I_{out}, is the difference of I_{ph}-I_{bias}, when the output voltage switches to the direct mode and when I_{ph} is higher than I_{bias}, a detectable I_{out} current is obtained. Therefore, the output current is used to estimate the light intensity.

Both topologies work for different illumination intensities; this allows increasing the DR. PWM topology working at tenuous light and direct current mode topology working at brighter illumination.

To measure the robustness of the design against process variations, 4-corners simulation is used submitting the circuit to extreme conditions.

Two examples of 4-corners simulation with 100 and 600 pA input current (labeled Typ1 and Typ2, respectively) using PWM mode readout are shown in Fig. 10; as can be seen in the first case (Typ1), the difference between the 4 simulations could be depreciate.

In the second case (Typ2), the maximum error was less than 5%, however, the variation in the voltage step is approximately 10%, due to the finite resistance of the switches. To solve this, an exhaustive study will be done to reduce the error using other topologies for the switches.

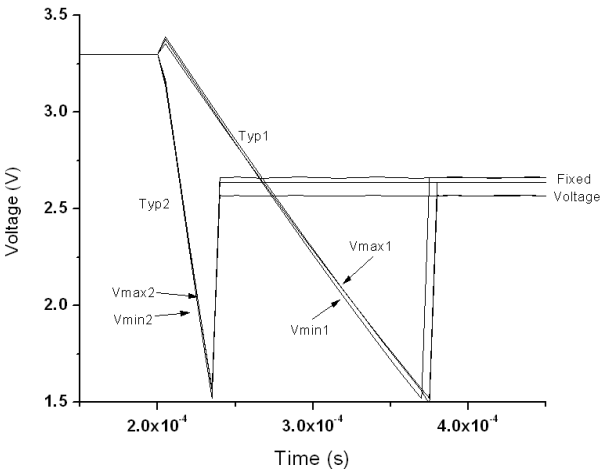


Figure 10. 4-corners simulation for PWM circuit with 100 and 600 picoA input currents, for the low current regimen no difference is observed. The high regime current shows variations of 5% in the ramp voltage and 10% in the voltage step.

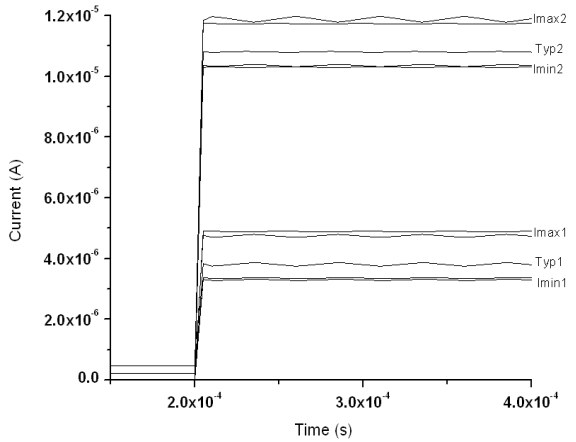


Figure 11. 4-corner simulation for direct mode circuit with 4 and 10 microA input currents; variations up to 10% are obtained.

Fig. 11 presents two examples of 4-corners simulation with 4 and 10 μA input currents (labeled Typ1 and Typ2, respectively); in this case, direct mode readout is used. In both cases, the worst case variation was 10% reflecting the 10% variations of the V_{out}.

Fig. 12 presents V_{out} for the PWM circuit and I_{out} for the direct mode circuit. As can be seen in the low current regimen, the PWM works well up to $1X10^{-8}$A; after this value, the output current starts to be measured directly. Consequently, the DR obtained in simulations is 160 dB, taking into account that the lower limit used to calculate it was the dark current of the photodiode.

The circuit was simulated using HSPICE, for a CMOS technology of 0.5 μm from MOSIS; all the simulations presented here were post-layout simulations. Fig. 13 shows the layout of one pixel.

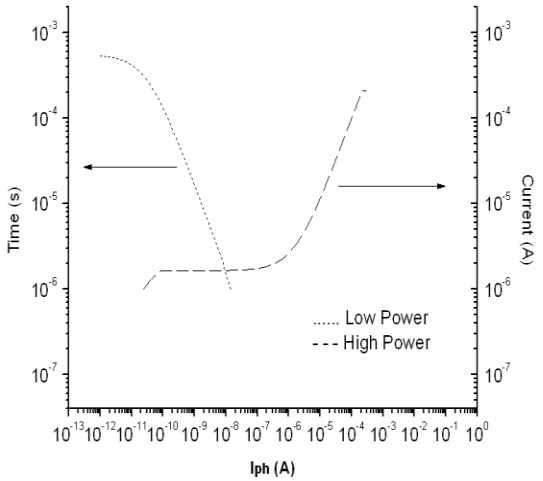


Figure 12. PWM and direct mode work ranges as function of photocurrent.

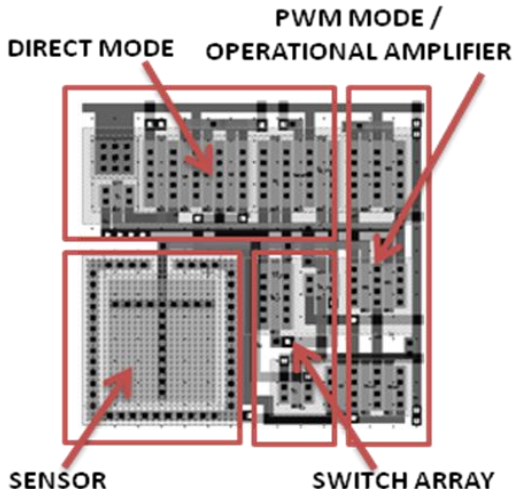


Figure 13. Layout of the PWM and direct mode circuit, using the 0.5 microns technology. The sensor area is 13x13 square microns and the total area is 50x50 square microns.

IV. EXPERIMENTAL CIRCUIT

In this section, experimental results are presented. First, the experimental procedure is explained. Second, results from characterization of UV sensor are presented. And finally, results from UV-ALD circuit with a variable source of light are presented.

A. Experimental Procedure

Currently, two integrated circuits are under fabrication: one is in a 2.5 microns technology and the other one is in 0.5 microns. The 2.5 microns technology is from CNM of Barcelona, Spain, and it allows modifying the process steps in order to integrate the sensor with the readout circuit. The compatibility of the sensor fabrication with a CMOS technology has been corroborated [4]. The 0.5 microns technology is from MOSIS. It is a standard technology and the same used in the simulation part. The MOS transistors used in the ALD arrays have characteristics similar to that of the MOSIS technology.

Experimentally, the circuit implementation was done using the quad N-channel matched pair MOSFET array ALD1106 [11] and the quad P-channel matched pair MOSFET array ALD1117 [12] with a discrete UV sensor.

The circuit in Fig. 6 was implemented with ALDs. The operational amplifier was accomplished using a Miller´s model amplifier. Instead of PMOS transistor Mrst, an external master reset was used. The dimensions of the ALDs transistors are fixed, so, it was necessary to use ten transistors in parallel to implement the FVF with a gain of ten. Transistors like transmission gates were used to implement the switches that allow the change between PWM to direct mode. The circuit was biased with a voltage of 5V and a current of 2.5 μA.

This discrete UV sensor was fabricated using a 2.5 microns technology. The sensor was not a standard Si PN junction; rather it incorporated special process steps during its fabrication [13]. Four sensors of area 0.0015 cm2 were built in the same chip and were packaged in a TO5. In this experiment, the four sensors were used simultaneously in parallel. Hereafter this arrangement is referred as "the sensor."

In order to characterize the sensor and the circuit from the 200 to 1000 nm wavelength range, the excitation light of a spectrofluorometer Horiba Jobin Yvon model FluourMax3 was used. The sensor was placed in a window normal to the light beam running from UV to NIR. To control the power intensity two configurations were used and referred to as: 5 slit and 15 slit. In order to do so the slit in the spectrofluorometer was in position 5 and 15.

Two measurements were done. First, the sensor current under different wavelengths was characterized. The spectrofluorometer was programmed to maintain each wavelength during 10 seconds and the current under each wavelength light was manually recorded using an electrometer Keitlhey model 6517A.

Next, the sensor was connected to the ALD circuit. The time and the current were measured from the outputs of the circuit. The output time was measured using an oscilloscope Agilent model 54622A and the current with the electrometer previously mentioned.

With this experimental setting the circuit implementation is expected to work as follows: when the master reset is turned on, the parasitic capacitor from the sensor is charged to Vdd, in this case 5V; when the excitation light from the spectrofluorometer illuminates the sensor and the master reset is turned off, a photocurrent from the sensor is generated. It is directly injected into the input of the FVF circuit implemented with ALD transistors, with the PWM topology enabled (S1 is turned off and S2 is in the REF position); this start the discharge of the capacitor. When the voltage of the sensor reaches the reference voltage (2.5V) a step is generated in the output of the operational amplifier working as a comparator, V_{out}. The elapsed time (or output time), since the reset is turned off until the voltage step is generated, is recorded using the oscilloscope and this time is proportional to the current.

The step in V_{out} is used to switch from PWM to direct mode topology automatically (S1 is turned on and S2 change to position p2), this is when the direct mode is activated. So, the photocurrent is directly injected to the FVF and is amplified by the current mirror. After, the photocurrent is measured directly by the electrometer.

When the power intensity used is low, 5 slit, a fine work for the PWM topology is expected. On the other hand, when the lamp has 15 slit the direct mode topology will have a better performance.

Both topologies work at separate times. The PWM topology works first and when the step in V_{out} appears, the PWM topology is automatically disabled and the direct mode topology starts to work.

B. Experimental Results and Discussion

1) UV-NIR Silicon Sensor

As mentioned above, a sensor capable of detecting from UV-NIR light is presented in [3]. This sensor was characterized under different types of illumination. Basic characteristics of the sensor were measured under these illuminations, and are showed in next figures: Fig. 14 shows the dark current, which is approximately 2pA; the current from the sensor with two power excitations is shown in Fig. 15; the typical responsivity is shown in Fig. 16; and the capacitance versus reverse bias voltage is presented in Fig. 17.

As seen in Figs. 14 and 15, depending on the power of the optical input the circuit has to be able to discriminate currents in the range of pA to μA.

2) Circuit

As is mentioned above, the UV-NIR sensor was connected to the arrangement and data obtained from the experiment are depicted in Table I, which shows: the current of the sensor (column "Current Sensor"); the output current of the current mirror (column "I_{out}-I_{bias}"); and the output time, from reset off to the step in V_{out} (column "Time Tout"); for 5 and 15 slit.

As it can be seen, for 5 slit the output time is clearly and easily discernible. However, for 15 slit, the current is higher and hence it is better to measure it directly.

For example, analyzing the case when the wavelength is 400nm, with 5 slit, the current from the sensor is 0.711 μA and the output current is $4x10^{-05}$ μA, hence there is no comparison between them. In this case, it is necessary to use the output time to have an adequate output, which is easily discernable (in this case 180 us). On the other hand, when the intensity increase to 15 slit, it is better to take the output current, since it is 4.10 μA, which agrees with the current from the sensor that is 4.290 μA and the output time is too small.

The different wavelengths used, have different intensities corresponding to the Xe lamp spectrum [14], but it is clear, from Table I, which variable (current or time) is better to use depending on the power of each wavelength.

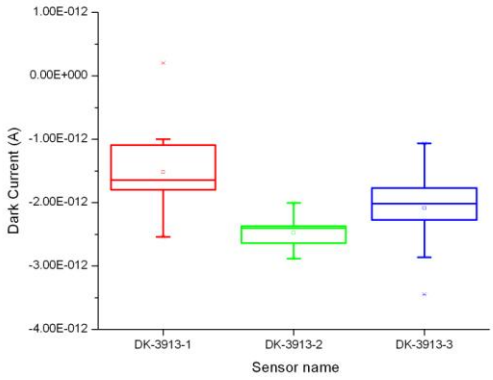


Figure 14. Dark current from different UV silicon photodiodes.

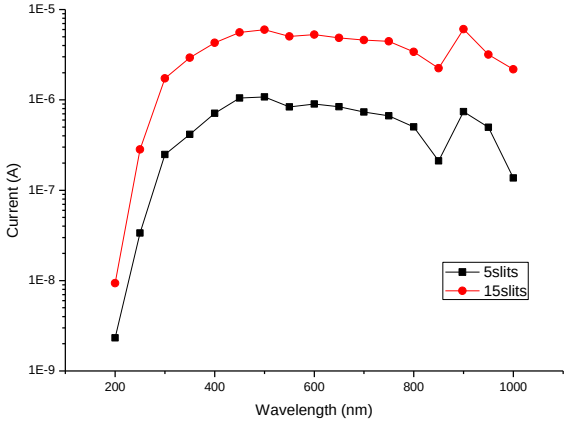


Figure 15. Typical UV sensor current as a function of wavelength for different optical power excitation.

TABLE I. DATA FROM SENSOR AND THE CIRCUIT IMPLEMENTATION FOR DIFFERENT LIGHT EXCITATIONS

Wavelength (nm)	5 slit			15 slit		
	Current Sensor (μA)	*I_{out} - I_{bias} (μA)*	*Time Tout (us)*	*Current Sensor (μA)*	*I_{out} - I_{bias} (μA)*	*Time Tout (us)*
200	0.002	0.000155	492	0.009	1.60	472
250	0.033	0.000155	416	0.283	1.20	288
300	0.248	0.000120	336	1.730	0.50	132
350	0.415	0.000085	228	2.930	2.40	68
400	0.711	0.000040	180	4.290	4.10	48
450	1.050	0.000010	152	5.590	5.70	36
500	1.080	0.000020	153	5.970	5.50	36
550	0.835	0.000040	156	5.040	4.80	36
600	0.900	0.000040	156	5.290	4.60	36
650	0.839	0.000030	156	4.860	4.80	36
700	0.735	0.000060	176	4.600	4.40	36
750	0.667	0.000070	192	4.450	3.60	44
800	0.503	0.000080	224	3.400	2.60	36
850	0.211	0.000120	316	2.240	1.20	92
900	0.740	0.000060	184	6.060	6.50	40
950	0.489	0.000090	232	3.170	2.20	72
1000	0.137	0.000095	368	2.180	0.80	104

To visualize better the different cases: when it is necessary the use one output (current) or the other one (time), figures 18-20 shows the discharge voltage of the capacitor and the output voltage for 5 and 15 slit, with different wavelengths.

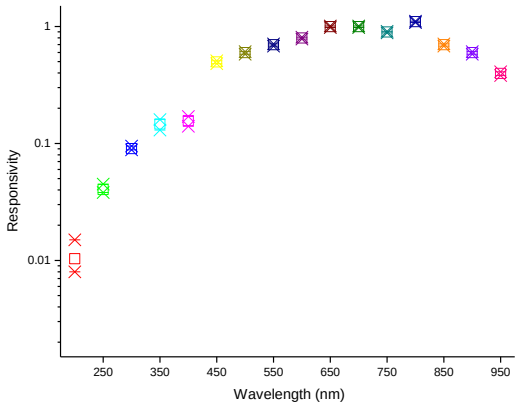


Figure 16. Typical responsivity of the sensor.

In Fig. 18, the fall off voltage of the parasitic capacitor of the UV sensor (when a beam of 200nm and 5 slit is applied) and the output signal from the comparator are shown.

In Figs. 19 and 20, the voltage in the PD and the output time are shown when the sensor is illuminated with different wavelengths but the same power intensity (5 slit). From this figure, it is clear that, for different light intensities, the current is different and that different time intervals are produced, which means the higher the current the shorter the output time.

The voltage in the PD and V_{out} as a function of time are shown in Figs. 21 and 22, respectively, but this time 15 slit was used. Comparing Figs. 19 and 20 with 21 and 22, it can be seen that the time out is shorter as the power is increased. This is a confirmation that as the power increases, the output time could be so short that it will be difficult to measure and differentiate a change in power or wavelength. In this case, for 15 slit it is hard to differentiate between 400, 600 and 800 nm, but for 5 slit it is difficult only to differentiate between 400 and 600 nm.

A comparison between the current from the sensor with the current from the output of the circuit is shown. In Fig. 23, the comparison is for 5 slit, while in Fig. 24 it is for 15 slit, and it is confirmed that only for the high power case most of the points correspond.

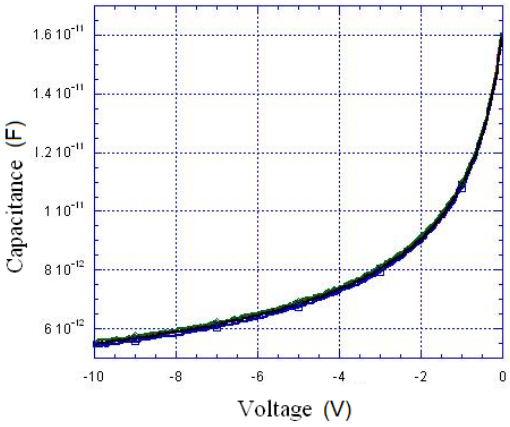


Figure 17. Typical capacitance VS reverse bias voltage.

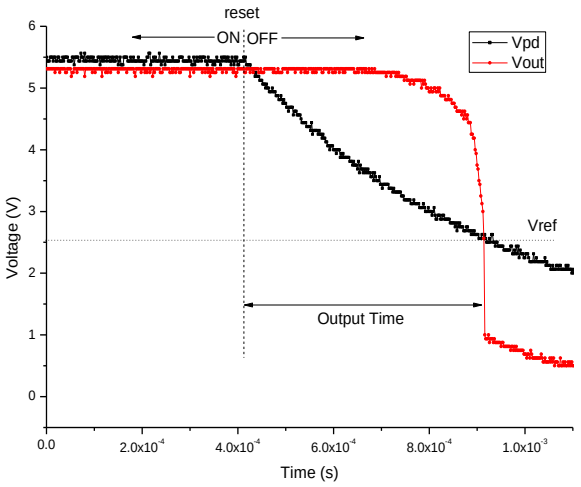


Figure 18. Output time and sensor decay voltage with a light excitation at the input of 200nm with 5slit

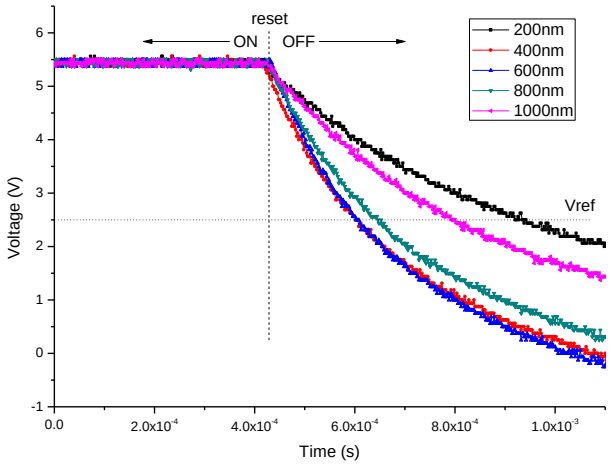


Figure 19. The discharge voltage of the capacitor as function of time is shown for different wavelengths. The input was illuminated with 5 slit.

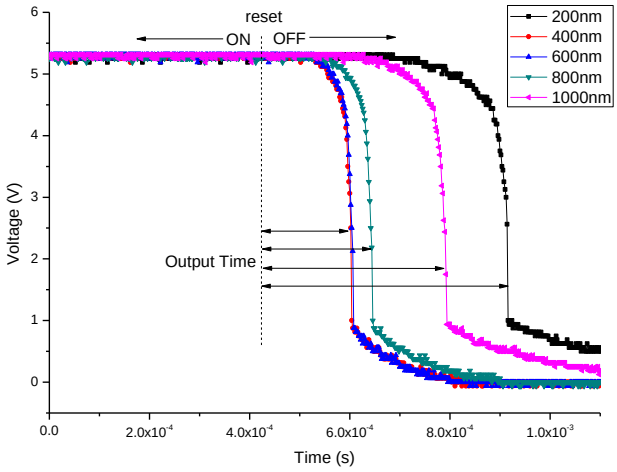


Figure 20. Output voltage for different wavelengths with 5 slit, the elapsed time for 200 nm is the largest time. The shortest elapsed time corresponds to the 400 nm wavelength, indicating that the blue color is the more intense in the Xenon lamp and produces the highest current in the sensor.

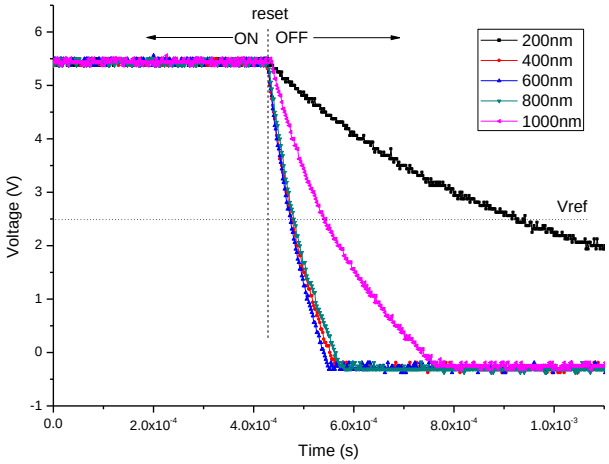


Figure 21. The discharge voltage of the capacitor as function of time is shown for different wavelengths. The input was illuminated with 15 slit.

It is clear that using the PWM and direct mode sequentially it is possible to sense low and high power signals. So, it has been proven experimentally that the circuit of Fig. 6 can be used to measure automatically very low and high currents with precision. The circuit has a total DR of 143 dB taking into account that the lower limit to calculate it was the intrinsic noise of the ALD circuits, and the upper limit was determined by the saturation current. Moreover, the DR can be incremented towards the upper limit, increasing the bias current, but this will cause an augment in the power consumption.

In Table II, a comparison between different topologies using the multimode techniques to increase DR is presented.

From this table it is inferred that only one, the Lineal-Logarithmic, reports a higher DR than the one presented here. However, the authors [17] used a smaller technology without reporting the power consumption.

Other articles present DR that exceeds the one reported here, but these works do not use the multimode technique nor the same technology [20]. Some reports do not mention the methods or techniques used to obtain a high DR [21].

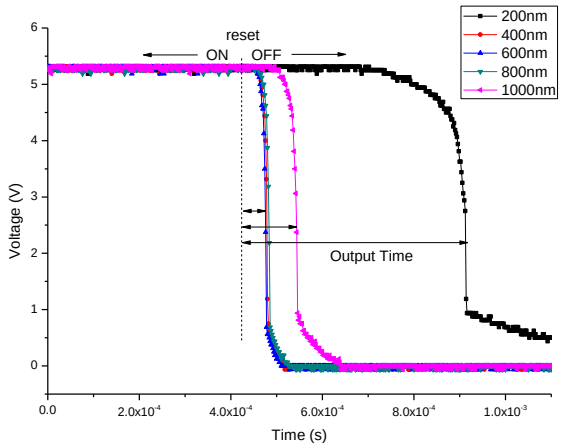


Figure 22. Output voltage for different wavelengths with 15 slit.

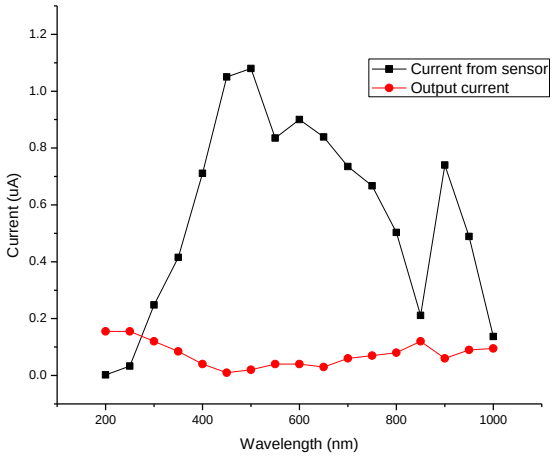


Figure 23. Comparison between the currents from the sensor and the circuit output currents with 5 slit.

TABLE II. MULTIMODE SENSING TECHNIQUES COMPARISON

	Specifications					
	Tech	***DR***	***Area***	***Power Consumption***	***Year***	***Ref***
Lineal – Logarithmic	0.35	124	7.5x7.5	---	2005	15
Lineal – Logarithmic	0.18	143	5.6x5.6	61mW y 84mW	2006	16
Lineal – Logarithmic	0.35	200	20x20	---	2006	17
Lineal – Logarithmic	0.35	112	9.4x9.4	---	2011	18
PWM – PFM	0.18	143	30x30	175mW	2011	19
PWM – Direct Mode	**0.5**	**160**	**50x50**	**36 µW**	---	**This work**
	ALDs	**143**	**discrete**	**70 µW**	---	

Another advantage to our approach is the improvement of the power consumption compared with those in Table II. This is because FVF working in current mode consumes negligible power.

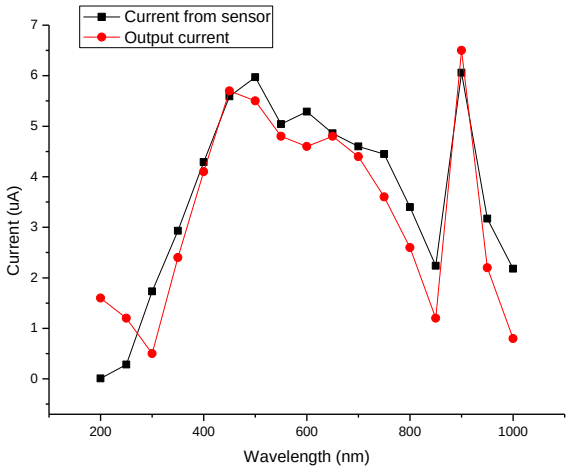


Figure 24. Comparison between the currents from the sensor and the circuit output currents with 15 slit.

V. CONCLUSIONS

A circuit that can be used in an integrated smart-pixel was designed, simulated and validated experimentally. It was corroborated that a wide dynamic range is achieved combining the pulse width modulation and direct current amplifications techniques in the same pixel; experimentally and by simulation, a DR of 143 dB and 160 dB were respectively obtained.

A reduced number of transistors were obtained sharing an operational amplifier by both techniques. The PWM and Direct mode are automatically selected depending on the range of current provided by the sensor. In the PWM mode the output time is used to estimate small photocurrents; in the direct mode high photocurrents are directly obtained. In comparison with similar circuits reported, the circuit proposed here improves the state of art.

In order to reduce area and power consumption, a single pixel circuit was designed using a 0.5 microns CMOS technology from MOSIS and occupying 50x50 square microns in total area.

The implementation with ALDs was tested in various wavelengths using a spectrofluorometer like the source of light.

Results show that the PWM topology works in cases when illumination is lower and the direct mode topology when is effective when it is higher.

Compared with other circuits reported in the literature, our approach has one of the highest DRs and smallest power consumptions, as demonstrated by the simulation.

ACKNOWLEDGMENT

The authors appreciate the English revision done by Rebekah Hosse Clark, also we appreciate the support of CONACyT.

REFERENCES

[1] E. Gómez-Ramírez, A. Díaz-Méndez, M. Aceves-Mijares, J. M. Rocha-Pérez, J. M. Pedraza-Chávez, and C. Domínguez-Horna, "Wide dynamic range readout for cmos pixel using pwm and direct mode sensing techniques," in Proceedings of the Fourth International Conference on Sensor Device Technologies and Applications IARIA, Aug. 2013, pp 64-70, ISBN: 978-1-61208-297-4

[2] J. Ohta, "Smart CMOS Image Sensors and Applications," CRC Press, 2008.

[3] D. Berman-Mendoza, M. Aceves-Mijares, L. R. Berriel-Valdos, J. Pedraza, and A. Vera-Marquina "Fabrication, characterization, and optimization of an ultraviolet silicon sensor," Optical Engineering, vol. 47, no. 10 , Oct. 2008, p. 104001, doi:10.1117/1.3000434

[4] M. Aceves-Mijares, E. Gómez-Ramírez, A. Díaz-Méndez, J. M. Rocha-Pérez, J. M. Pedraza-Chávez, J. Alarcón-Salazar, S. Román-López, C. Domínguez-Horna, A. Merlos, X. Formatjé, and A. Morales-Sánchez "Conservation of the optical properties of sro after cmos IC processing," in press.

[5] A. Spivak, A. Belenky, A. Fish, and O. Yadid-Pecht "Wide-dynamic-range cmos image sensors: comparative performance analysis," IEEE Transactions on Electron Devices, vol. 56, no. 11, Nov. 2009, pp. 2446-2461.

[6] D. Park, J. Rhee, and Y. Joo "A wide dynamic-range cmos image sensor using self-reset technique," IEEE Electron Device Letters, vol. 28, no. 10, Oct. 2007, pp. 890-892.

[7] B. Goldstein, D. Kim, A. Rottigni, J. Xu, T. K. Vanderlick, and E. Culurciello "Cmos low current measurement system for biomedical applications," IEEE International Symposium on Circuits and Systems (ISCAS), May 2011, pp. 1017-1020.

[8] A. Zarándy, "Focal-Plane Sensor-Processor Chips," Springer, 2011.

[9] R. González-Carvajal, J. Ramirez-Angulo, A. J. Lopez-Martin, A. Torralba, J.A. Gómez-Galan, A. Carlosena, and F. Muñoz-Chavero, "The flipped voltage follower: a useful cell for low voltage low power circuits design," IEEE Transactions On Circuits And Systems, vol. 52, no. 7, Jul. 2005, pp. 1276-1291.

[10] Synopsys, "HSPICE", URL: http://www.synopsys.com/, 2013.

[11] N-channel MOSFET, URL:http://html.alldatasheet.net/html-pdf/55017/ALD/ALD1106/46/1/ALD1106.html, 2014.

[12] P-channel MOSFET, URL:http://html.alldatasheet.es/html-pdf/55019/ALD/ALD1117/46/1/ALD1117.html, 2014.

[13] D. Berman-Mendoza, M. Aceves-Mijares, L. R. Berriel-Valdos, J. Carranza, J. Pedraza, C. Domínguez-Horna, and C. Falcony "Silicon-rich silicon oxide films boost UV sensitivity," Laser Focus World, vol. 41, no. 9, Sept. 2005, p.103.

[14] Fluoro max-3 operation manual, URL:http://www. Jobinyvon.com, 2013.

[15] K. Hara, H. Kubo, M. Kimura, F. Murao, and S. Komori "A linear-logarithmic cmos sensor with offset calibration using an injected charge signal," IEEE International Solid-State Circuits Conference, vol. 1, Feb. 2005, pp. 354-603.

[16] G. Storm, R. Henderson, J. E. D. Hurwitz, D. Renshaw, K. Findlater, and M. Purcell "Extended dynamic range from a combined linear-logarithmic cmos image sensor," IEEE Journal of Solid-State Circuits, vol. 41, no. 9, Sept. 2006, pp. 2095-2106.

[17] N. Akahane, R. Ryuzaki, S. Adachi, K. Mizobuchi, and S. Sugawa "A 200dB dynamic range iris-less cmos image," IEEE International Solid-State Circuits Conference, Feb. 2006, pp. 1161-1170.

[18] M. Vatteroni, P. Valdastri, A. Sartori, A. Menciassi, and P. Dario "Linear–logarithmic cmos pixel with tunable dynamic range," IEEE Transactions On Electron Devices, vol. 58, no. 4, Apr. 2011, pp. 1108-1115.

[19] C. Posch, D. Matolin, and R. Wohlgenannt "A qvga 143 dB dynamic range frame free pwm image sensor with lossless pixel," IEEE Journal Of Solid-State Circuits, vol. 46, no. 1, Jan. 2011, pp. 259-275.

[20] N. Ide, W. Lee, N. Akahane, and S. Sugawa "A wide DR and linear response cmos image sensor with three photocurrent integrations in photodiodes, lateral overflow capacitors, and column capacitors," IEEE Journal Of Solid-State Circuits, vol. 43, no. 7, Jul. 2008, pp. 1577-1587.

[21] Electronic Publication: Omron Corporation: "German venture company develops highly advanced wide dynamic range cmos image sensor," [Online]. URL: http:// industrial.omron.fr/fr/news/news/cmos-image-sensor, 2003.

Carrier Photogeneration in Metal-Semiconductor Structures Using Thin Films of Rutile-Phase TiO_2 Nanoparticles

Joel Molina, Carlos Zuniga, Edmundo Gutierrez,
Electronics Department,
National Institute of Astrophysics, Optics and Electronics
Santa Maria Tonantzintla, Puebla, Mexico
E-mails: jmolina@inaoep.mx, czuniga@inaoep.mx, edmundo@inaoep.mx

Eunice Mendoza, Jose Luis Sanchez, Erick Bandala
Energy and Environment Research Group,
Universidad de las Americas, Puebla
San Andres Cholula, Puebla, Mexico
E-mails: edith.mendozaco@udlap.mx, jluis.sanchez@udlap.mx, erick.bandala@udlap.mx

***Abstract* — In this work, rutile-phase TiO_2 nanoparticles (r-TiO_2) are embedded within a Spin-On Glass oxide matrix (using a simple, low thermal budget and economic deposition method) and the final TiO_2:SiO_2 suspension is used as photoactive material in Metal-Semiconductor structures. For this purpose, so-called "*horizontal*" and "*vertical*" structures are fabricated and characterized under I-V-Light conditions. The electronic, physical, chemical and photovoltaic characteristics of the final structures are obtained and correlated when irradiated with ultraviolet-visible (UV-Vis) light sources. I-V-Light characterization of these structures shows a reduction in the total resistance of thin aluminum stripes or reduction in the resistance state of TiO_2:SiO_2 composed dielectric when irradiated with UV light (compared to dark conditions). Because of the photogenerated carriers, these structures have *photoresistor* or *photocapacitor* features, which are quite useful for solar energy conversion and storage.**

Keywords-TiO_2 nanoparticles; photoresistor; photocapacitor; photogeneration; sol-gel processing; metal-semiconductor.

I. INTRODUCTION

Recently, carrier photogeneration during ultraviolet-visible (UV-Vis) irradiation upon rutile TiO_2 nanoparticles has been demonstrated by using so-called "horizontal" and "vertical" metal-semiconductor structures based on binary metal oxides [1]. There, direct exposure of TiO_2 nanoparticles to light irradiation enables a reduction in the total resistance of thin aluminum stripes or the sudden increase of the gate current of a Metal-Insulator-Metal (MIM) structure by about 4-7 orders of magnitude, thus showing the potential of TiO_2 nanoparticles for photovoltaic conversion and even solar energy storage.

It is widely known that TiO_2 (whether in *rutile*, *anatase* or *brookite* crystalline phases) posses enough photocatalytic properties than can be used for efficient conversion of solar energy into electric current if proper device architectures are provided. Photogeneration of electron–hole pairs in TiO_2 occurs naturally when this material is irradiated under high energy conditions like ultraviolet-visible light sources, and whose energy (in electron-Volts, eV) is well matched to the energy gap of TiO_2. After electron–hole photogeneration, an efficient mechanism for separation of these carriers is needed and the simplest way to achieve this is by developing a large enough electric field so that the negative and positive charges are attracted to the positive and negative polarities of the applied voltage, respectively. These very simple mechanisms of photogeneration and carrier separation by electric field are what most solar cells (mostly based in P–N junctions) use for conversion and handling of an electric current whose magnitude is in direct proportion to the energy and density of radiation being absorbed.

On the other hand, the use of TiO_2 nanoparticles for energy conversion is quite attractive given its high contact surface area as compared to a dense bulk film of the same material. This is important since, after irradiation with the proper light sources, a larger density of photogenerated carriers are expected in devices using photoactive nanoparticles. These photogenerated carriers could then be used for more efficient energy conversion ad even, simultaneous energy conversion and storage of the carriers in the same device. In this sense, and even though rutile-phase TiO_2 is considered a very inefficient material in terms of its photocatalytic activity (ability for carrier photogeneration) [2-3], use and development of this semiconductor material is quite important since the chemical synthesis of TiO_2 usually produces rutile phase TiO_2 quite easily, with relatively low concentration of impurities and also, economically. On the other hand, the synthesis of anatase-phase TiO_2 is more complicated, usually involving complex chemistry and/or doping with some metal or non-metal elements in order to increase its photocatalytic activity when exposed to UV or visible irradiation [4-7].

Additionally, low thermal budget processing of metal-semiconductor structures based on TiO_2 nanoparticles is useful in order to increase the lifetime of photogenerated carriers. This way, relatively higher quantum efficiency during photon-electron conversion is expected and therefore, higher densities of photogenerated currents upon light irradiation. In both our horizontal and vertical devices, a low thermal budget has been kept by using a maximum processing temperature of 250°C, which can be even lowered to 100°C in order to evaporate mostly water and some organic solvents off the TiO_2:SiO_2 suspension and for film densification. This low-thermal budget process makes these devices ideal for fabrication on large-area flexible substrates (like PET or any other polymer) with the potential to decrease their final costs for widespread use.

In this work, we embed rutile-phase TiO_2 nanoparticles (r-TiO_2) within an organic SiO_2 matrix and the final mixture of this dielectric structure is deposited on thin films of evaporated aluminum stripes. The final "horizontal" metal-semiconductor structure is then electrically characterized under dark and light conditions (I-V-light) so that the total resistance of a simple aluminum stripe is measured and compared before and after UV irradiation. Compared to dark conditions, excess carriers are photogenerated within the TiO_2 nanoparticles after light exposure and they are directly transferred to both ends of the aluminum stripe after applying a low potential difference. The highest density of photogenerated carriers is obtained when the TiO_2:SiO_2/Al is irradiated with UV-B light so that the total aluminum resistance is reduced by about 43%. Therefore, this initial device acts like a very simple "*photoresistor*". Additionally, we also fabricate so-called "vertical" metal-semiconductor-metal structures in order to obtain a solar energy conversion device with the intrinsic ability to self-store must of the converted energy in the form of a rechargeable capacitor. This device then acts like a very simple "*photocapacitor*" [8-9]. The *state-of-the-art* regarding these latest structures makes use of complex layered structures going from photo-rechargeable textiles for wearable power supplies [10], up to dye-sensitized solar cells (DSSC) connected in series with Li-ion batteries, metal oxides and/or TiO_2 nanotube arrays in order to increase energy conversion efficiency [11-13]. However, because of increasing fabrication complexity and use of a third additional electrode (in order to switch between the functions of energy conversion, storage and output), which consumes extra energy and increase cost of fabrication, a simpler two-electrode device is needed and that requirement is met by our proposed device structures.

This paper is arranged as follows: in Section I, we gave an introduction about the importance of testing simple "horizontal" and "vertical" metal-semiconductor structures, which make use of TiO_2 nanoparticles in order to promote energy conversion in "*photoresistor*" and "*photocapacitor*" devices. Section II presents the experimental conditions used for fabrication of these structures as well as details about the measurement setup that is used for their physical and photo-electrical characterization. Section III presents and discusses the main experimental results that are found for these structures, thus confirming the ability of r-TiO_2 to act as photoactive material for both conversion and storage of solar energy. Finally, the main conclusions drawn from all results are highlighted in Section IV, from where we state that it is possible to use the "vertical" structure as a *photocapacitor*, thus enabling direct storage of solar energy.

II. EXPERIMENTAL

A. Deposition of Thin Metal Films by Electron-Beam Evaporation under Ultra-High Vacuum Conditions

For the horizontal TiO_2:SiO_2/Al/Glass structures, and previous to depositing TiO_2 nanoparticles on Corning glass slides (2947, size of 75 mm × 25 mm), the initial substrates were degreased by ultrasonic cleaning in trichloroethylene and acetone (10 min and 10 min, respectively). After cleaning the Corning glass slides, 400 nm of aluminum was deposited on one surface of the substrates by e-beam evaporation under ultra-high vacuum conditions (UHV). A metal mask was used during this metallization so that relatively large stripes of aluminum (18mm× 3mm) were left on the glass slides and these substrates are now ready for deposition of TiO_2:SiO_2 suspensions. For the vertical Ti/TiO_2:SiO_2/Al/Glass structures, 400 nm of aluminum are initially deposited on one surface of the cleaned glass slides without using any metal mask. This initial aluminum layer works as the bottom electrode of the MIM structure. After deposition of the TiO_2:SiO_2 suspension, 100 Å of titanium is then deposited by e-beam evaporation (also in UHV conditions). This second metallization with titanium is now performed through a metal mask so that stripes (same size as before) are left on top of the final vertical structure. For all aluminum and titanium metallization, a same deposition rate of 1 Å/sec inside a vacuum of 10^{-7} Torr was used.

B. Preparation of Thin FilmsBased on TiO_2 nanoparticles Embedded within an Organic SiO_2 matrix

We have used a low-organic content or silicate-type spin-on glass (SOG)-based SiO_2 (700B from Filmtronics, Corp.) as a matrix for immobilization of r-TiO_2 (Dupont, R-706 with 93% purity and having an average diameter of 360 nm before embedding). Initially, specific amounts of commercial r-TiO_2 are suspended in deionized water by hydrolyzing this TiO_2:H_2O mixture in a hot water bath (*baine marie*, 45°C, 30 min) and then, adding SOG-based SiO_2 so that the final TiO_2:SiO_2:H_2O mixture is again subjected to a final hot water bath (*baine marie*, 80°C, 1 hr) in order to obtain an homogeneous suspension. The concentration ratios of TiO_2 (solute) to SiO_2:H_2O (solvent) are 200, 100, 50 and 10 mg/mL and these suspensions are labeled as A, B, C and D, respectively. The solute concentrations were measured with an analytical balance AG285 from Mettler-Toledo. The final TiO_2:SiO_2:H_2O suspensions were directly applied on the surface of glass slides (Corning glass 2947, size of 75 mm X 25mm), that were previously metalized with aluminum stripes. The suspensions were first spinned at 3000 rpm, 30 sec, and then 4000 rpm, 15 sec in order to obtain uniform layers of r-TiO_2 embedded in SiO_2. After spinning, all films (A-D) were baked for 2 hours using a hot plate at 250°C in N_2 flow (99.99% purity) in order to evaporate mostly water and some of the organic solvents present in the SOG-based SiO_2 matrix. For FTIR characterization, the same processing sequence was followed and the final solution was applied on prime-grade P-type silicon wafers (100) with resistivity of 5–10 Ω·cm in order to eliminate most of the organic and impurity elements present within the Corning glass slides. Given the ultra low thermal budget required for fabrication of these simple structures, their introduction into large area flexible substrates is expected, thus promoting wide spread use of optimized devices. The fabrication process flow for both structures is briefly summarized in Figure 1.

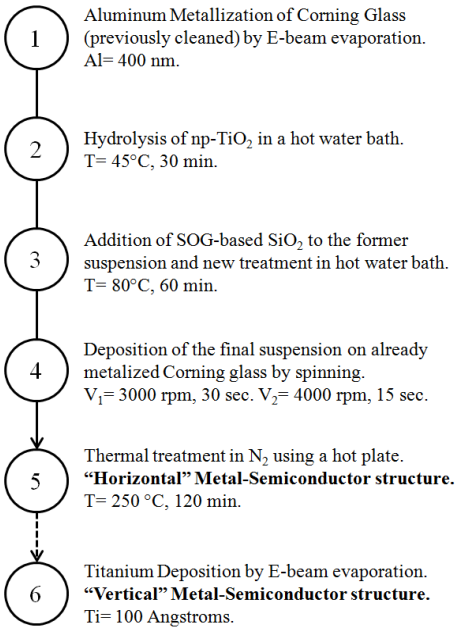


Figure 1. Process flow for fabrication of "horizontal" and "vertical" Metal-Semiconductor structures using r-TiO_2 as photoactive material.

It is important to notice that we did not synthesize the r-TiO_2 used for fabrication of all proposed structures. Instead, we decided to use readily available commercial TiO_2 nanoparticles (with rather low purity of 93% and large average diameter of 360 nm) in order to test the ability of this material for carrier photogeneration under several sources of light irradiation.

C. *Chemical and Physical Characterization of TiO_2 nanoparticles Embedded within SiO_2*

Dynamic-Light Scattering (DLS) measurements (Nanotrac Wave, from Microtrac) [14] were done in order to determine the final size distribution of TiO_2 nanoparticles after the embedding process. By using DLS, we are able to determine both the size and size distribution profile of TiO_2 nanoparticles in the final suspension (before deposition on metalized glass surfaces). In particular, the size distribution profile for r-TiO_2 is obtained with high accuracy by this system (close to 100% signal intensity), thus giving a direct estimation of the homogeneity of the r-TiO_2 in the final suspension. Also, thicknesses for all films were measured by profilometry (DEKTAK, V200-SI) after partially etching the TiO_2:SiO_2 film using a strong acid solution composed of diluted HF (HF:H_2O, with 1:2 ratio). The crystalline phases of the resulting TiO_2-based films were obtained after XRD measurements using an X-ray diffractometer (Empyrean, from PANalytical), with a scanning step of 0.02°, using Cu-K_α radiation with λ= 1.5406 Å as an X-ray source. The band gap energies Eg of the resulting films were calculated using optical transmittance data measured with an UV-Vis absorption spectrometer (LAMBDA 3B with double beam from Perkin Elmer, with Corning glass used as substrate) and the Tauc method [15]. Chemical compositional analyses for all films were obtained by FTIR spectrum measurements in absorbance mode with a Bruker Vector-22 system after 5 min of purge in N_2. The samples were measured against crystalline silicon substrate or SOG based silicon dioxide on glass (both were used as references).

D. *Electrical Characterization of Horizontal and Vertical Metal-Semiconductor Structures under I-V-Light*

As stated before, only the most concentrated suspension (A suspension) was used for fabrication of the horizontal and vertical structures and therefore, for I–V–Light characterization (I–V measurements under dark/illumination conditions). For characterizing these final structures we have used an HP4156B semiconductor parameter analyzer at 300 K. As light source, we have used natural light conditions of sunlight coming indirectly to the laboratory room, sunlight plus a white lamp put right above all the structures and an UV-B lamp (~ 300nm). For the horizontal structure, an I–V sweep was applied to the ends of the same aluminum stripe while limiting the current compliance to 100 mA. For the vertical structure, an I–V measurement in sampling mode was used where the gate current was constantly monitored with time while the gate voltage was kept constant at Vg = 10 V. In this latter case, the gate current was limited to 100 μA. For all structures, I-V-Light measurements were applied to at least 10 different metal-semiconductor devices in order to obtain typical experimental data.

III. RESULTS AND DISCUSSION

A. *Structure's Schematics and Energy Band Diagrams*

Figure 2 shows the 3-D and top view schematics for the horizontal TiO_2:SiO_2/Al/Glass structure that has been used as a photoresistor upon irradiation of different light sources.

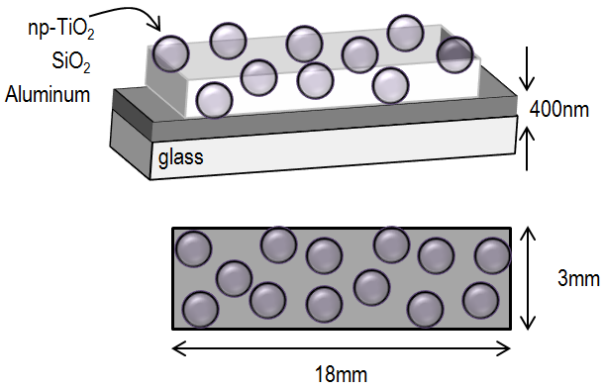


Figure 2. 3D and top views of r-TiO_2:SiO_2 deposited on Aluminum/Glass.

Because of the preparation method (previously discussed), uniform distribution of the r-TiO_2 within the oxide matrix is promoted so that these nanoparticles should have almost the same diameter size and separation in between as well. The aluminum stripes have an area of 18X3 mm^2 with thickness of 400 nm. For this structure, I-V-Light characterization takes place by measuring a current flow at

both ends of the same bottom aluminum electrode while both bottom and top electrodes are used in the vertical structure.

Figure 3 shows the idealized energy band diagrams for TiO_2 and TiO_2:SiO_2/Al systems during photogeneration of carriers after irradiation with energies greater than the band gap of TiO_2 hν≥Eg(TiO_2). The band gap Eg, in semiconductor theory, is the void energy region that separates the valence band from the conduction band. For TiO_2, the band gap can be overcome with energy from near UV photons. In the first band diagram, all physical mechanisms during light irradiation, (1) excitation, (2) relaxation and (3) diffusion are also shown while a small potential difference is developed in the second diagram so that carriers are injected in the metal. Excitation of a TiO_2 system with any light irradiation source having an energy hυ ≥ Eg (TiO_2), will generate an electron-hole pair density proportional to the irradiation density of the light source. These photogenerated electron and hole pairs are then enabled for electrical conduction within the conduction and valence bands, respectively. After excitation, relaxation mechanisms lower the potential energy of the excited carriers and then, diffusion by means of a concentration gradient, or even drift, by means of an applied electric field, is possible.

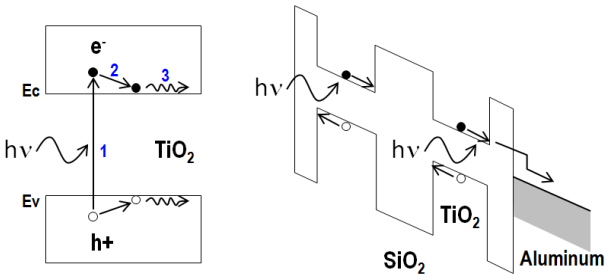


Figure 3. Idealized energy band diagrams for TiO_2, TiO_2:SiO_2/Al systems.

By applying a small potential difference to opposite ends of the same bottom aluminum electrode (see Figure 2 and right side of Figure 3), photogenerated carriers are then injected into this metal and immediately separated by the electric field, thus decreasing its original bulk resistance. In the vertical structure, a potential difference applied to both top/bottom electrodes would separate all photogenerated carriers thus increasing the level of the original gate current (decreasing the original resistance state of the combined TiO_2:SiO_2). In both cases, we do not take into account most of the physical mechanisms responsible for lowering the quantum efficiency of the structures: trapping, recombination, phonon interaction, annihilation, interface defects, etc. However, those mechanisms and defects are important in order to properly engineer the fabrication process for these structures and then, increase their quantum efficiency for less energetic sources like visible instead of UV irradiation (reducing Eg or increasing the lifetime of photogenerated carriers before recombination).

On the other hand, the photocatalytic properties of TiO_2 (quite useful for its bactericide properties) are derived from the formation of photogenerated charge carriers (hole and electron), which occurs upon the absorption of ultraviolet or visible light and that corresponds to the band gap of this material [16-20]. The photogenerated holes in the valence band diffuse to the TiO_2 surface and react with adsorbed water molecules, forming hydroxyl radicals ($^{\bullet}OH$). The photogenerated holes and the hydroxyl radicals oxidize nearby organic molecules on the TiO_2 surface. Meanwhile, electrons in the conduction band typically participate in reduction processes, which typically react with molecular oxygen in the air to produce superoxide radical anions ($O2^{\bullet-}$). In this sense, because of the high surface contact area presented by TiO_2 nanoparticles, this material will generate a large density of electron and hole pairs, which then can be used for both photovoltaic or photocatalytic applications having a high degree of effectiveness.

B. Dynamic Light Scattering (DLS) and Profilometry Results for Measuring Diameter of r-TiO_2 and Thickness of TiO_2 :SiO_2 Based Thin Films

As result of synthesis of the corresponding suspensions, Figure 4 shows the average TiO_2 particle diameters for two conditions: as-prepared (measured after at least 24 h of settling time of the synthesized suspensions) and right after sonication using an ultrasonic vibrator (Branson B1510, 2 min at 40 kHz and room temperature). The dotted arrow shows the nominal average diameter as stated by the manufacturer and that is located at about 360 nm. The as-prepared samples present a larger particle diameter because of their tendency to agglomerate or aggregate after dispersion and settling within a liquid suspension. During sonication, enhanced dispersion of TiO_2 agglomerates is obtained by overcoming their weaker attractive forces, the final result being smaller TiO_2 nanoparticle diameters. The average physical size for sonicated r-TiO_2 nanoparticles is around 300 nm. Figure 4 also shows the final TiO_2 film's thickness after deposition (by spinning on silicon) and thermal treatment of the prepared suspensions. Thicker TiO_2 films were obtained for more concentrated suspensions as expected.

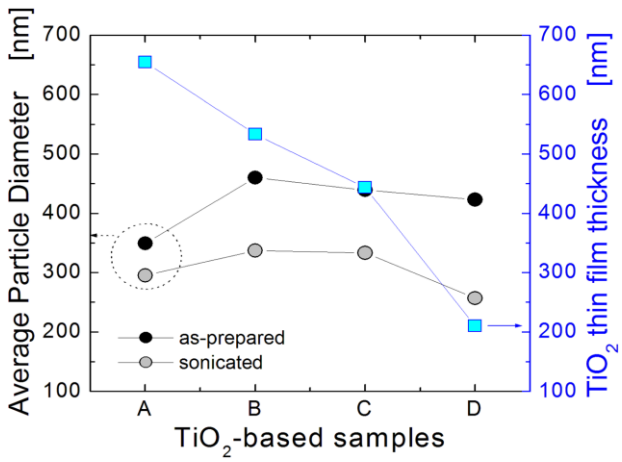


Figure 4. DLS and profilometry data showing the nanoparticle diameter size (before and after sonication) and TiO_2:SiO_2 thin film thickness, respectively.

As reference, film thickness of SOG based oxide alone (without dilution in H_2O) was about 400 nm, which compares well with data obtained for similar spinning conditions [21]. Since all A–D suspensions were prepared using different TiO_2:SiO_2:H_2O volume ratios (see the experimental procedure), different concentrations of TiO_2 within the same matrix would produce different thin film thicknesses. A similar trend of thickness reduction for SiO_2 when diluted using different percentages of H_2O was also found by Molina et al. [21] and final thickness of SiO_2 could be further reduced when the samples were annealed at higher temperatures. Given that for sample D there is a minimum amount of TiO_2 nanoparticles (only 10 mg·mL-1), the final average thickness for this film was kept below the average nanoparticles' diameter because of the reduction in most of the SiO_2 thickness by enhanced H_2O dilution and also, because larger amount of water per volume induces further dilution or dispersion of previously agglomerated TiO_2 nanoparticles.

Finally, it is important to notice that since these TiO_2:SiO_2 based films will be under direct irradiation of visible-ultraviolet light sources, measurements of the nanoparticles' sizes and their distribution at both the film's interfaces and in the bulk are necessary to mainly correlate surface distribution of nanoparticles to their carrier photogeneration. These measurements could be done by scanning electron microscopy (SEM) or atomic force microscopy (AFM) for the surface, while transmission electron microscopy (TEM) could be done to obtain the characteristics of the nanoparticles in the bulk of the embedding matrix. These last measurements are actually under progress, and the results will be published elsewhere.

C. X-Ray Diffraction (XRD)

After obtaining TiO_2-based thin films, Figure 5 shows the XRD diffraction patterns for all thin films including sample 0 (only SOG based SiO_2), which is only the amorphous matrix of SiO_2 deposited atop the glass slides and cured at the same temperature. It is clear that sample 0 does not present the characteristic sharp diffraction peaks of a crystalline material because the incident X-rays are scattered in many directions leading to a large bump distributed over a wide range of 2θ, just like the one shown here for reference purposes. Samples C–D present the lowest intensities for the diffraction peaks related to the crystalline phase of TiO_2, since their nanoparticle concentrations were quite small, (50 and 10 mg·mL^{-1}, respectively). On the other hand, samples A–B (with concentrations of 200 and 100 mg·mL^{-1}, respectively) clearly present the characteristic sharp diffraction peaks for rutile-phase TiO_2 including the broad amorphous phase from both the SiO_2 matrix and the glass slide (used for TiO_2 immobilization and mechanical support purposes, respectively). Given the relatively high concentration density of r-TiO_2 embedded within the SiO_2 matrix for the A-B samples, it is clear that sharper diffraction peaks will be obtained possibly because of nanoparticles' agglomeration. This effect could be triggered during spinning, which make use of high speed centrifugal forces during step 4 of the process flow; see Figure 1. In particular, sample A showed the clearest and highly intense diffraction peaks, which fit well with those of standard rutile TiO_2 (Powder Diffraction File No. 21-1276).

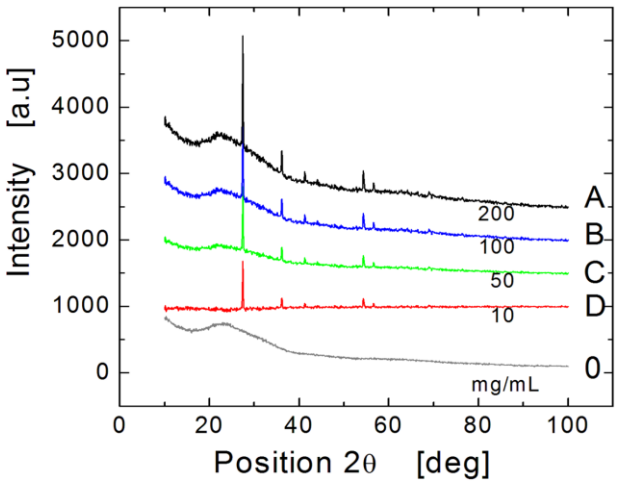


Figure 5. XRD data show existence of rutile phase for TiO_2 nanoparticles.

These results provide direct evidence of the existence of rutile phase in the thin TiO_2 films after using high TiO_2 nanoparticle concentrations (at least 50 mg·mL^{-1}). This is important since rutile-phase TiO_2 is considered a very inefficient material in terms of its photocatalytic activity and yet, we could obtain moderate photovoltaic activity by using the proposed horizontal and vertical metal-semiconductor structures here discussed.

D. Fourier-Transformed InfraRed (FTIR) Spectroscopy

The IR spectra for all samples are shown as absorption coefficient a after normalization of each sample to a single and averaged TiO_2 film thickness. It is important to notice that these samples are quite different for bulk and dense TiO_2 thin films. In these films, the TiO_2 nanoparticles tend to disperse within the SiO_2 matrix (generating empty spaces) thus avoiding normalization with respect to each physical thickness. Figure 6 shows typical chemical bond vibration energies in absorption mode, found in samples A-D for all the range of interest (wavenumbers from 4000–400 cm^{-1}).

The IR spectra for SOG-based SiO_2 (deposited on glass) is also included and whose absorption peaks for the Si-O bonds are detected at 1070, 943, 801, 570, and 443 cm^{-1} (peaks 1-5). The bands centered at 1070 and 801 cm^{-1} correspond to symmetric and asymmetric stretching vibrations of Si-O-Si bonds; the bands at 943 and 443 cm^{-1} correspond to bending vibrations of the Si-OH and Si-O-Si bonds, respectively, which confirm that the main composition of this material is SiO_2. The absorption band at 570 cm^{-1} is assigned to symmetric Si-O-Si vibrations of the SOG oxide. A decrease in the intensities of the absorption band related to Si-O-Si stretching vibrations was observed for the SOG and D-A spectra (in that sequence), which imply that IR energy is absorbed by the presence of other material.

On the other hand, a combination of both the glass slide and the SOG-based SiO_2 present high IR absorption characteristics (high Si–O–Si bond density), which screen-out the presence of any detectable Ti–O–Ti bonds.

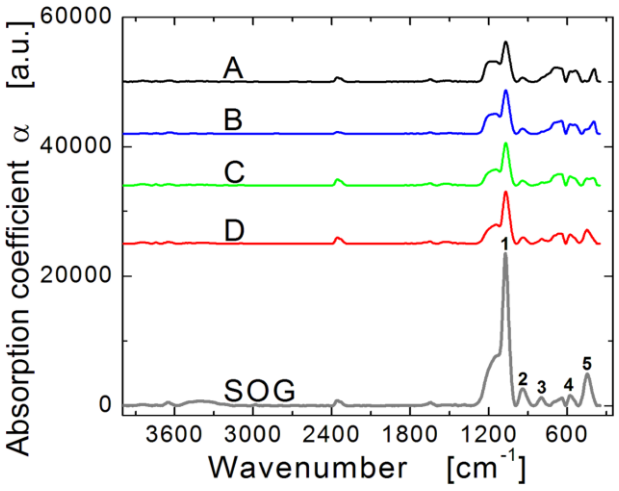


Figure 6. IR spectra (absorbance mode) of TiO_2:SiO_2/glass samples (A-D). Because of the high density of Si–O and Si–O–Si bonds, the presence of any detectable Ti–O–Ti bonds has been screened out in these measurements.

In order to analyze only the contribution of TiO_2 in the films, the IR spectra of A-D samples must be obtained using only the SOG-based oxide film as reference (SiO_2/glass) and take these measurements at wavenumber between 1600 and 400 cm^{-1} approximately. This way, we are able to eliminate the influence of the highly absorbent peaks related to Si–O and Si–O–Si bonds (those especially found at 1070 and 443 cm^{-1}), which could screen-out the presence of any detectable Ti–O–Ti bonds. Figure 7 shows the new IR spectra (using the SiO_2/Glass sample as reference) including some absorption bands at 765 (black arrow at shoulder section), 530 and 395 cm-1.

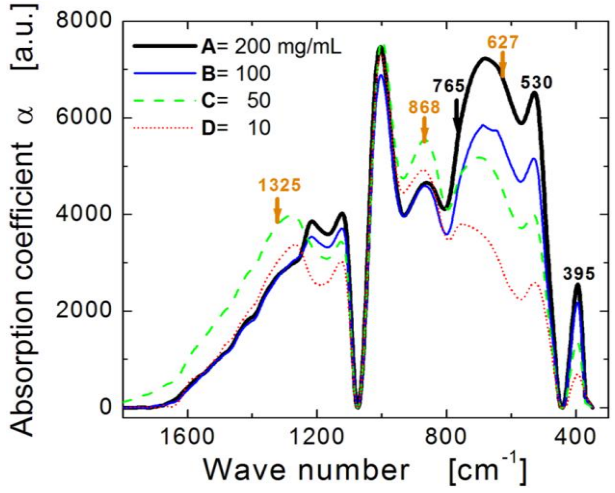


Figure 7. IR spectra (absorbance mode) of TiO_2:SiO_2/glass samples (A-D) and using SiO_2/Glass as reference in order to detect Ti–O–Ti bonds.

The band detected as a shoulder at 765 cm^{-1} is comparable with the IR spectrum of crystalline TiO_2 (having anatase or rutile crystalline structure) due to symmetric stretching vibrations of the Ti–O bonds. Most importantly, the absorption peaks for these Ti–O bonds increase with the content of TiO_2, thus confirming proper chemical bonding of this photoactive material for even the larger TiO_2 concentrations. Also, even though the main absorption bands for the Si–O–Si bonds mostly disappear when SOG-based oxide film is used as reference, some Si–O bonds appear in these samples surely because of vibrations of Si–O–Ti bonds, as observed in the band at 1005 cm^{-1}. On the other hand, the bands at 627 and 1325 cm^{-1} are thought to be related to vibrations of some Al–O bonds while the band at 868 cm^{-1} is related to a combination of both Al–O and Si–O bonding vibrations. Detecting contributions of both Al–O and Si–O bonds, make sense considering that commercial TiO_2 nanoparticles consist, according to the manufacturer, of 93% TiO_2, 2.5% Al_2O_3 and 3% SiO_2 (the remaining percentage consisting of other undetectable elements).

Figure 8 shows the IR absorption spectra of TiO_2:SiO_2 films from 4000 to 2600 cm^{-1}, where the presence of strong absorption bands in the region of 3200-3500 cm^{-1}, (corresponding to bending vibrations of adsorbed and possibly coordinately bounded OH molecules with Ti or Si) are noticed.

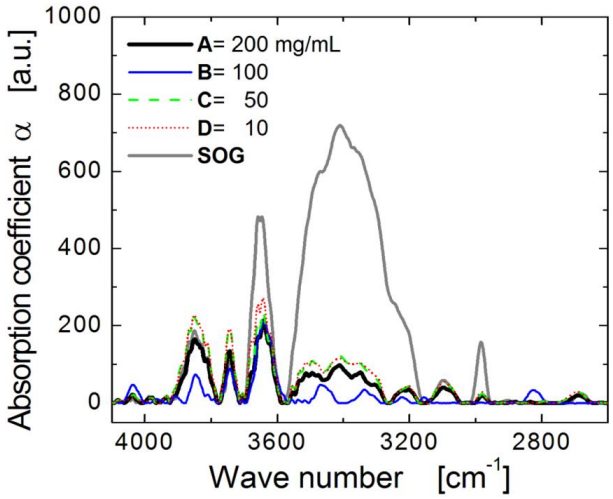


Figure 8. IR spectra (absorbance mode) of TiO_2:SiO_2/glass samples (A-D) for wavenumbers in the 4000 to 2600 cm^{-1} region. This measurement enables detection of any remaining organic impurities in the samples.

Also, the weak absorption peaks in the region of 2920-2930 cm^{-1} among others are related to the presence of organic residues that were not fully evaporated or decomposed during the low-temperature thermal treatments applied to the TiO_2 films. For comparison, the IR spectra for SOG-based oxide is also shown so the main absorption bands related to purely organic elements can be easily identified (having the strongest absorption peaks). Since the highest thermal treatment applied to all A-D samples was only 250°C, higher temperatures for this final curing process would evaporate or reduce these organic residuals more

efficiently. However, for practical applications, it is desired to reduce the total thermal budget for this material so that it could be possible to develop coatings of TiO_2 (with enough photovoltaic or light response) on plastic or other economic and readily available flexible substrate for large area applications.

E. UV-Vis Transmittance Spectroscopy

Optical band gap Eg, is an important physical parameter in semiconductor materials because it allows knowing the threshold energies to which a material in particular, like TiO_2, is "transparent" or able to absorb incident photons and therefore, create electron-hole pairs that could participate in photovoltaic processes. The UV-Vis spectra from 190 to 900 nm region for the different TiO_2 concentrations (A-D samples), including the spectrum for only the glass substrate, are all shown in Figure 9.

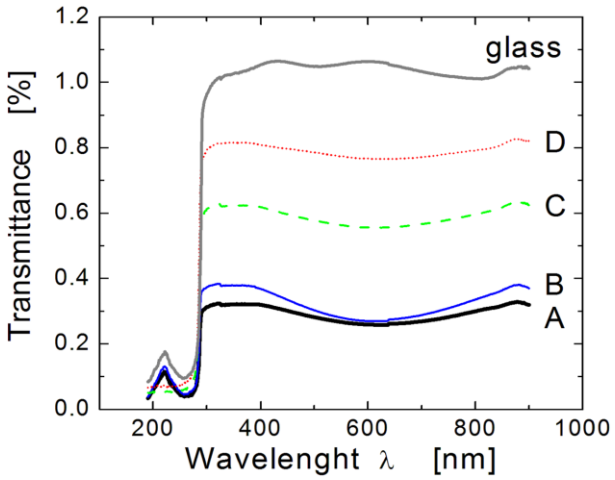


Figure 9. UV-Vis spectra obtained for all TiO_2:SiO_2 (A–D, Glass) samples showing that transmittance is in reverse proportion to TiO_2 concentration.

It can be seen that strong absorption occurs at wavelengths λ<290 nm (UV-B regime) for all samples and that transmittance is reduced in direct proportion to the r-TiO_2 concentration as expected. Even though the physical thicknesses for all samples are different (see Figure 4), minimum variations in their optical band gap are expected if we consider different densities for these films.

Additionally, since the number of interference fringes is not visible, this avoids using the Swanepoel model for optical band gap calculation Eg [22] and therefore, a fast estimation was done to obtain this important parameter. Figure 10 shows the transmittance spectra versus photon energy for the A-D samples. The inflexion point is the crossing for all samples after linearly extrapolating all slopes with the axe for photon energy. The band gap Eg is determined by dividing this inflexion point by $\sqrt{2}$ [23]. The inflexion points cut the photon energy axis at between 4.41–4.42 eV so that the correspondent optical band gap Eg for all A-D samples lies at 3.11-3.12 eV.

This band gap energy Eg corresponds well with the reported Eg for anatase or rutile TiO_2, between 3.0 and 3.2 eV, respectively [24-25].

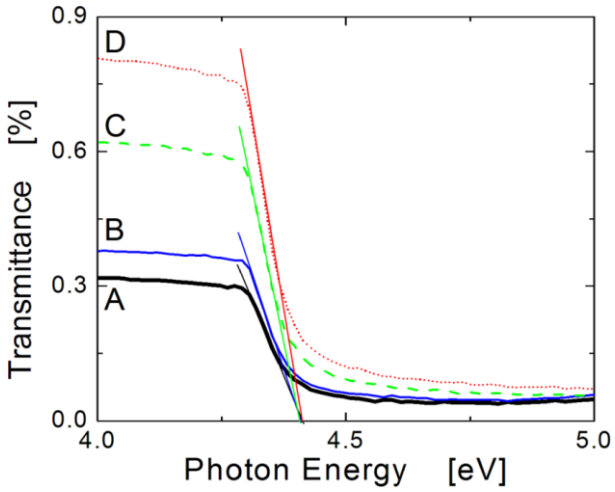


Figure 10. Extrapolation of UV-Vis transmittance data to photon energy for all A-D samples in order to obtain TiO_2 band gap energy (Eg).

On the other hand, for the vertical structure (Ti/TiO_2:SiO_2/Al/Glass device), we make use of an ultra-thin titanium stripe (thickness of 100 Å) in order to use it as a conductive top electrode while being able to transmit most of the UV-Vis irradiation through itself, thus allowing efficient photogeneration of carriers in the TiO_2. In order to obtain the optical properties of this electrode, the transmittance spectra (from 190 to 900 nm region) for different ultra-thin titanium films are shown in Figure 11.

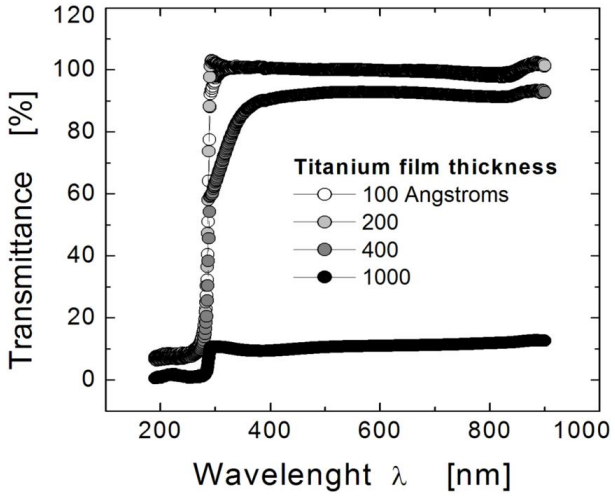


Figure 11. Transmittance (UV-Vis) spectra of ultra-thin titanium films. The titanium film having 100 Å in thickness was used as the top electrode in the vertical Ti/TiO_2:SiO_2/Al/Glass MIM structures.

We clearly notice that the thinner titanium films (100 and 200 Å in thickness) transmit virtually the complete (100%) electromagnetic spectra from 900 nm down to 290 nm, thus being transparent to the visible, the UV-A and UV-B regions

as well. For the UV-C region, all films readily absorb or reflect this energy so that the transmittance characteristics are lost. For a titanium film of 400 Å in thickness, the original transmittance falls to about 93% from 900 nm down to ~400 nm, thus this sample is transparent to only the visible region of the spectra. At wavelengths of 400 nm and downwards, this sample starts to absorb/reflect UV-A and UV-B thus it is not useful for proper absorption of those energies (by TiO_2) in the vertical structures. We also show the very low transmittance characteristics (around 10%) obtained from a thicker titanium film (1000 Å) as a reference.

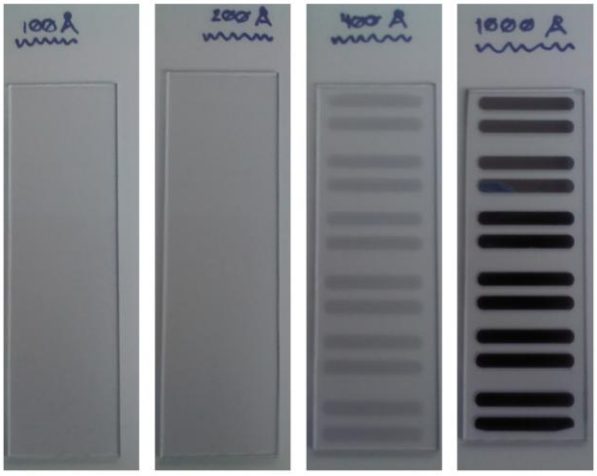


Figure 12. Photograph showing the transmittance quality of ultra-thin titanium films. The titanium films having 100 and 200 Å in thickness (in the form of stripes) are optically transparent while titanium films having 400 and 1000 Å in thickness are opaque since they absorb/reflect most UV.

Figure 12 above shows the optical transparency of thin titanium films deposited on Corning glass slides in the form of stripes (after E-beam evaporation under ultra-high vacuum conditions). We clearly notice that titanium films having 100 and 200 Angstroms in thickness are optically transparent while thicker films are opaque. In our vertical MIM structures, we have used an ultra-thin titanium film (having only 100 Å in thickness) so that we ensure good transmittance characteristics (especially in the UV regime) along with good conductive properties in order to have a transparent-conductive electrode.

F. I-V-Light Response of Horizontal TiO_2:SiO_2/Al/Glass

Figure 13 shows the I-V-Light characteristics of the structure shown previously in Figure 2 (A sample only). Dark, sunlight, sunlight+lamp and UV-B light (~300 nm) conditions were all applied on top of the structures so that surface r-TiO_2 were the first to absorb all possible irradiation coming from these sources. Compared to dark conditions, photogeneration of excess carriers (both electrons and holes) within the TiO_2 nanoparticles is greater after UV-B light exposure and these carriers are directly transferred to both ends of the Al-stripe after applying a low potential difference. During UV-B light irradiation, the total aluminum resistance is reduced by about 43%, which represent a moderate change in resistance given by rather low quantum efficiency presented by this structure.

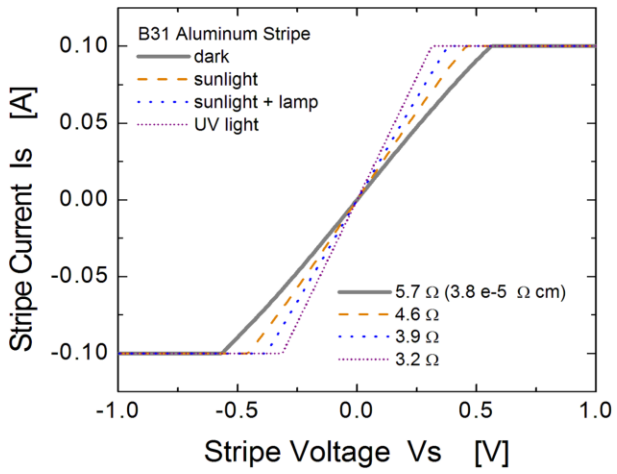


Figure 13. I–V–Light characteristics of an aluminum stripe (horizontal TiO_2:SiO_2/Al/Glass structure) before and after light irradiation. This device acts as a *photoresistor*.

G. I-V-Light Response of Vertical Ti/TiO_2:SiO_2/Al/Glass

Previously, I-V-Light characterization for horizontal structures produced a moderate photogeneration of carriers so that the total resistance of an aluminum strip was reduced. However, given that some of the photogenerated carriers will be trapped, recombined or "*lost*" within the SiO_2 matrix or at its interface with r-TiO_2 (any annihilation mechanism), the "*horizontal path*" followed by carriers in the initial structure would reduce their lifetime once they are photogenerated in the r-TiO_2. In order to increase photocarrier lifetime before recombination and thus, increase quantum efficiency during UV-B irradiation, vertical structures are proposed, see Figure 14.

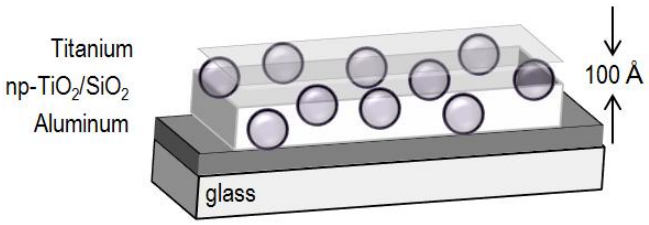


Figure 14. 3D view for a vertical Ti/TiO_2:SiO_2/Al/Glass structure where the titanium electrode is 100 Å in thickness, thus being optically transparent.

These vertical structures use titanium as a gate electrode (only 100 Å in thickness) so that a capacitor in the form of a Metal-Insulator-Metal structure is formed. Because of the ultra-thin titanium layer, this gate electrode is highly transparent to all UV-Vis irradiation so that when all carriers are being photogenerated, a vertical transition of these carriers between bottom/top electrodes (by an applied external electric field) would require a shorter distance thus

increasing their lifetime before recombination as compared to the horizontal structures.

Figure 15 shows the I–V–Light characteristics (under dark/illumination conditions) of three vertical Ti/TiO_2:SiO_2/ Al/Glass structures. Here, an I–V measurement in sampling mode (I-time) was used where the gate current Ig was constantly monitored with time and the gate voltage was kept constant (Vg = 10 V, applied at the semitransparent Ti electrode). We notice that the gate current increases about 4-7 orders of magnitude when the structures are exposed to sunlight. A high velocity photo-response to optical excitation is obtained given the shortest vertical transition between top and bottom electrodes.

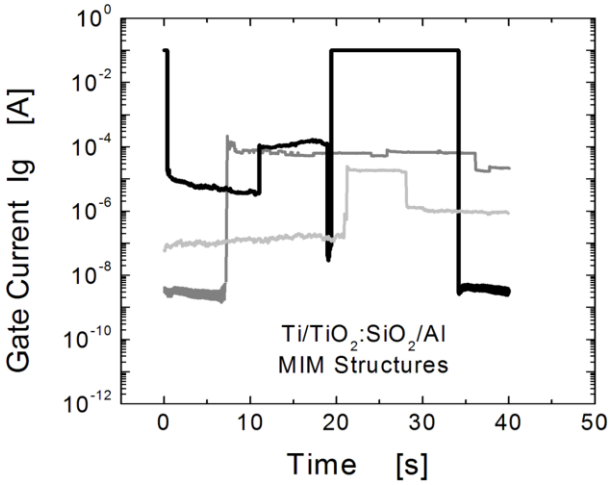


Figure 15. I–V–Light characteristics of three different MIM capacitors (Ti/TiO_2:SiO_2/Al/Glass structures) before and after sunlight irradiation. All samples present rapid dark-to-illuminated Ig transitions and vice versa. Any of these devices acts as a *photocapacitor*, thus enabling direct storage of solar energy after photogeneration of carriers.

These vertical structures are able to photogenerate and store carriers quite similar to the so-called *photocapacitor* [7-8], where all carriers could be efficiently stored within the dielectric itself right after photogeneration. Thus, a light-driven self-charging capacitor having a direct and efficient storage mechanism of solar energy has been obtained.

IV. CONCLUSIONS

Photocarrier generation during UV-B exposure of rutile-phase TiO_2 nanoparticles in horizontal and vertical metal-semiconductor structures, reduces the total resistance of an aluminum stripe by about 43% or instantly increase the gate current Ig for about 4-7 orders of magnitude, respectively. Both structures can be fabricated using simple and economic processing techniques with low thermal budget. The vertical structure works as a *photocapacitor*, enabling simultaneous conversion and storage of solar energy.

ACKNOWLEDGMENT

This work was fully supported by the National Council of Science and Technology (CONACyT-Mexico).

REFERENCES

[1] J. Molina, C. Zuniga, E. Gutierrez, E. Mendoza, J.L. Sanchez, and E.R. Bandala, "Carrier photogeneration during UV-Vis irradiation on horizontal and vertical metal-semiconductor structures based on rutile-phase TiO_2 nanoparticles," Proc. of the Fourth International Conference on Sensor Device Technologies and Applications, SENSORDEVICES 2013. vol. 1, pp. 41-45, 2013.

[2] Z. Ding, G.Q. Lu, and P.F Greenfield, "Role of the crystallite phase of TiO_2 in heterogeneous photocatalysis for phenol oxidation in water," J. Phys. Chem. B, vol. 104, April 2000, pp. 4815-4820.

[3] T.A. Kandiel, R. Dillert, A. Feldhoff, and D. Bahnemann, "Direct synthesis of photocatalytically active rutile TiO_2 nanorods partly decorated with anatase nanoparticles," J. Phys. Chem., vol. 114, February 2010. pp. 4909-4915.

[4] M.A. Behnajady, N. Modirshahla, M. Shokri, and B. Rad, "Enhancement of photocatalytic activity of TiO_2 nanoparticles by silver doping: photodeposition versus liquid impregnation methods," Global NEST Journal, vol. 10-1, 2008, pp. 1-7.

[5] D.H. Kim, D.K. Choi, S.J. Kim, and K.S. Lee, "The effect of phase type on photocatalytic activity in transition metal doped TiO_2 nanoparticles," Catalysis Communications, vol. 9, March 2008, pp. 654–657.

[6] C.D. Valentin, G. Pacchioni, and A. Selloni, "Origin of the different photoactivity of N-doped anatase and rutile TiO_2," Phys. ReV. B., vol. 70, August 2004, pp. 085116-1--085116-4, 2004.

[7] O. Diwald, L. Thompson, E.G. Goralski, S.D. Walck, and J.T. Yates, "The effect of nitrogen ion implantation on the photoactivity of TiO_2 rutile single crystals," J. Phys. Chem. B, vol. 108, January 2004, pp. 52-57.

[8] T. Miyasaka and T.N. Murakami, "The photocapacitor: an efficient self-charging capacitor for direct storage of solar energy," Appl. Phys. Lett., vol. 85, October 2004, pp. 3932-3934.

[9] C.W. Lo, C. Li, and H. Jiang, "A photoelectrophysical capacitor with direct solar energy harvesting and storage capability," Optical MEMS and Nanophotonics (OPT MEMS), 2010 International Conference on, August 2010, pp. 65-66.

[10] T. Song and B. Sun, "Towards photo-rechargeable textiles integrating power conversion and energy storage functions: can we kill two birds with one stone?," ChemSusChem, vol. 6, January 2013, pp. 408–410.

[11] X. Zhang, X. Huang, C. Li, and H. Jiang, "Dye-sensitized solar cell with energy storage function through PVDF/ZnO nanocomposite counter electrode," Adv. Mater., June 2013, pp. 1-4.

[12] W. Guo, X. Xue, S. Wang, C. Lin, and Z.L. Wang, "An integrated power pack of dye-sensitized solar cell and Li battery based on double-sided TiO_2 nanotube arrays," Nano Lett., vol. 12, no. 5, April 2012, pp. 2520–2523.

[13] M.S. Nuckowska, K. Grzejszczyk, P.J. Kulesza, L. Yang, N. Vlachopoulos, L. Häggman, E. Johansson, and A. Hagfeldt, "Integration of solid-state dye-sensitized solar cell with metal oxide charge storage material into photoelectrochemical capacitor," J. of Power Sources, vol. 234, no. 15, July 2013, pp. 91-99.

[14] "Microtrac: total solutions in particle characterization, NanotracWave," App Note, Microtrac. October 2012, pp. 1-4. http://www.microtrac.com/MTWP/wp-content/uploads/2012/10/Nanotrac-Wave-Temp-Brochure-Ver-9.pdf

[15] J. Tauc, "Optical properties and electronic structure of amorphous Ge and Si," MRS Bulletin, vol. 3, January 1968, pp. 37-46.

[16] A. Fujishima, T.N. Rao, and D.A. Tryk, "Titanium dioxide photocatalysis," J. Photochem. Photobiol. C: Photochem. Rev., vol. 1, March 2000, pp. 1–21.

[17] A. Fujishima, X. Zhang, and D.A. Tryk, "TiO_2 photocatalysis and related surface phenomena," Surf. Sci. Rep., vol. 63, December 2008, pp. 515–582.

[18] K. Nakata and A. Fujishima, "TiO_2 photocatalysis: design and applications," J. Photochem. Photobiol. C: Photochem. Rev., vol. 13, June 2012, pp. 169–189.

[19] M.R. Hoffmann, S.T. Martin, W. Choi, and D.W. Bahnemann, "Environmental applications of semiconductor photocatalysis," Chem. Rev., vol. 95, January 1995, pp. 69–96.

[20] D.S. Bhatkhande, V.G. Pangarkar, and A.A.C.M. Beenackers, "Photocatalytic degradation for environmental applications – a review," J. Chem. Technol. Biotechnol., vol. 77, January 2002, pp. 102–116.

[21] J. Molina , A.L. Munoz, A. Torres, M. Landa, P. Alarcon, and M. Escobar, "Enhancement of the electrical characteristics of MOS capacitors by reducing the organic content of H_2O-diluted spin-on-glass based oxides," Mater. Sci. Eng. B, vol. 176, no. 17, March 2011, pp. 1353-1358.

[22] R. Swanepoel, "Determination of the thickness and optical constants of amorphous silicon," J. Phys. E: Sci. Instrum., vol. 16, no. 12, December 1983, pp. 1214-1222.

[23] M. Sreemany and S. Sen, "A simple spectrophotometric method for determination of the optical constants and band gap energy of multiple layer TiO_2 thin films," Materials Chemistry and Physics, vol. 83, no. 1, January 2004, pp. 169–177.

[24] J. Dharma and A. Pisal, "Simple method of measurement the band gap energy value of TiO_2 in the powder form using UV/Vis/NIR spectrometer," App Note, Perkin-Elmer Inc. January 2009, pp. 1-4.

[25] S. Valencia, J.M. Marin, and G. Restrepo, "Study of the bandgap of synthesized titanium dioxide nanoparticules using the sol-gel method and a hydrothermal treatment," The Open Materials Science Journal, vol. 4, January 2010, pp. 9-14.

Feasibility of Geomagnetic Localization and Geomagnetic RatSLAM

Rafael Berkvens*, Dries Vandermeulen*, Charles Vercauteren*, Herbert Peremans†, and Maarten Weyn*

*CoSys-Lab, FTI, University of Antwerp, Paardenmarkt 92, B-2000 Antwerp, Belgium
†ENM, FTEW, University of Antwerp, Prinsstraat 13, B-2000 Antwerp, Belgium
rafael.berkvens@uantwerpen.be, dries.vandermeulen@student.artesis.be,
{charles.vercauteren, herbert.peremans, maarten.weyn}@uantwerpen.be

***Abstract*—The need for accurate indoor localization increases as we get used to accessible outdoor localization, and the number of applications depending on localization grows. Indoor localization is challenging because of frequent line of sight obstructions and dynamic changes in the environment. Magnetometers can be found in many modern electronic devices and provide a simple way to measure the geomagnetic field intensity. Due to distortions in this magnetic field, these measurements often provide enough information to enable identification of a location using pattern matching. We show the feasibility of using these magnetic field intensity measurements in localization and SLAM applications. Our SLAM system of choice is the biologically inspired RatSLAM, as it allows pattern matching as scene recognition. We demonstrate a number of experiments in various environments, including a suburban house and a university lab. We conclude that geomagnetic localization and SLAM are feasible in environments with many distortions in the magnetic field. Such locations are easier to identify than locations with little distortions, which will have the same pattern of magnetic field over larger areas.**

***Keywords*—Indoor localization; Indoor SLAM; Magnetic field intensity; Geomagnetic indoor localization; RatSLAM.**

I. Introduction

As we state in our AMBIENT 2013 paper [1], the outdoor global positioning system fails when used indoors. In addition, localization systems based on a single technology are prone to failure [2]. The last decade localization related research is focusing more and more on indoor localization, since most use cases concerning people or asset tracking also require an indoor location estimation.

Indoor localization can be performed by detecting the presence of radio frequency devices, of which Wi-Fi is probably the most common. Such technologies have been developed in an opportunistic sensor fusion system in [3]. These systems can be enhanced by additional localization measurements. The earth's magnetic field is even more ambient than Wi-Fi access points, and research shows that animals use this magnetic field for orientation [4, 5]. This leads to the idea that the earth's magnetic field can be used for indoor localization, a technique referred to as geomagnetic indoor localization.

In the field of geomagnetic indoor localization it is actually the distortions of the magnetic field that are used to find a location [6–11]. These distortions are usually created by concrete buildings, metal objects, electrical wires, etc. Our own research confirms these findings for different sensors and environments [1].

If a technology can be used for localization, it can often be used for simultaneous localization and mapping (SLAM). In localization, a map of the environment is available, with corresponding localization hints, such as access point locations or signal attenuation patterns [3]. In SLAM, this map is not available but is built simultaneously with the calculation of a path [12, 13]. Typical algorithms of SLAM are Extended Kalman Filter SLAM (EKF-SLAM), such as in [14]; GraphSLAM, such as in [15], which uses an information matrix; and FastSLAM [16], which uses a Monte Carlo particle filter.

A biologically inspired SLAM variant is RatSLAM [17], which is based on a rat's hippocampus. The hippocampus is the part of the brain where, among other things, the localization and mapping is done. This functionality is mimicked by the RatSLAM algorithm to create semimetric, topological maps of the environment. RatSLAM's original input is a simple web camera, which performed great even when mapping an entire suburb [18]. The camera input has also been replaced by a biomimetic sonar, an algorithm termed BatSLAM [19, 20]. Work has also been done to enable RatSLAM to use other sensors, like laser range finders, depth cameras, and simple sonars, in a sensor fusion system [21]. To summarize, the key difference between geomagnetic localization and geomagnetic RatSLAM is the need for an a priori known magnetic field intensity map for geomagnetic localization, which is not required for geomagnetic RatSLAM as such a map will be build implicitly by the system while exploring the environment.

This paper represents an extension of the work reported on in the paper [1] by applying the sensor model used for localization to the RatSLAM algorithm. This way, we can create maps suitable for geomagnetic indoor localization for a specific environment while simultaneously localizing on that map. Another advantage of such a system is that the magnetic maps used for indoor localization can at all times be kept up to date. Other geomagnetic SLAM approaches exist, one using a Monte Carlo particle filter [22] and another using a SLAM algorithm called FootSLAM [23, 24].

The structure of this paper is as follows. In Section II, we give some background on the earth's magnetic field and details on the RatSLAM algorithm, with a focus on the location recognition process. In Section III, we explain our pattern

matching measurement model. In Section IV, we provide detailed results for both localization and RatSLAM using the earth's magnetic field. In Section V, we come to our conclusion and discuss some of our future work.

II. Background

In this section, we discuss the main techniques that support our results, the sensing of the earth's magnetic field. Additionally, we describe how pattern matching localization was performed. Lastly, we explain the RatSLAM algorithm with a focus on the location recognition process.

A. Magnetic field sensing

In this section, we will discuss some issues to consider when measuring the magnetic field. Firstly, we will briefly discuss the magnetic field. Subsequently, we will explain how magnetometers measure this magnetic field and how they are influenced. Lastly, we will focus on the indoor magnetic field intensity, as this is our area of interest.

1) Magnetic B field: The earth's magnetic field is commonly called the magnetic B field. It originates from currents in the fluid outer core of the earth, which are created by both temperature, pressure, and composition of the fluid and the spin of the earth [10]. The magnetic B field is defined by its direction and intensity. The direction always points to the magnetic north; the intensity is measured in Tesla [T], and ranges between $22\,\mu$T and $67\,\mu$T according to [25].

The geomagnetic field vector, $\boldsymbol{B}_m$, has seven components, illustrated in Figure 1. The X intensity's axis points to the geographical north, which is at the north end of the axis around which the earth spins. The Y intensity's axis points to the corresponding geographical east. The Z intensity's axis points to the earth's nadir. Derived from X, Y, and Z are the total intensity F; the horizontal intensity H, which is the projection of F on the plane described by X and Y; the inclination I, which is the angle between F and the plane described by X and Y; and the declination D, which is the angle between X and H [25]. Note that H will point to the magnetic north of the earth, while X points to the geographical north of the earth.

At our location, Antwerp, Belgium, the declination H is $0^\circ\ 19'$ and inclination I is $66^\circ\ 25'$. The average total intensity F is $48.73\,\mu$T [1].

2) Magnetometers: The geomagnetic field vector $\boldsymbol{B}_m$ can be measured by magnetometers in the form of X', Y', and Z' intensities. These intensities are measured along the reference axes of the magnetometer and can only be translated to X, Y, and Z intensities by tilt compensation and turning the X' intensity's axis to the geographical north. Correspondingly, the F and H' intensities can be calculated as the euclidean norm:

$$F = \sqrt{X'^2 + Y'^2 + Z'^2} \quad (1)$$

$$H' = \sqrt{X'^2 + Y'^2} \quad (2)$$

where X', Y', Z', and H' indicate intensities measured relative to the orientation frame of the magnetometer. The

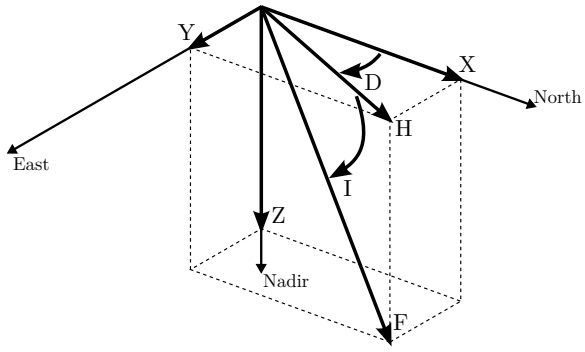


Figure 1. The components of the geomagnetic field vector $\boldsymbol{B}_m$, based on [25].

F intensity is the same for any orientation. Many recent, high level electronic devices, such as smartphones and tablets, feature magnetometers. This widespread availability of magnetometers makes them attractive to use for localization applications.

For the localization research, we use two different magnetometers on two different platforms. These are a Honeywell HMC5843 magnetometer, found in a Shimmer 9 DOF Kinematic sensor, and a AK9873 magnetometer, found in a Huawei Sonic U8650 smartphone. The behavior of both sensors was tested to determine if the localization performance was platform independent. For the RatSLAM research, a similar sensor is used. This is a AK9863 magnetometer, found in a LG Google Nexus 5 smartphone.

We will repeat here our extensive testing of the sensors used for the localization research [1]. The first test is conducted in an indoor bedroom apartment where both sensors are individually placed on a wooden desk, away from any possible interference factors like metal objects or electronic devices. The sensor sends data back via Bluetooth to a computer where all data was recorded. Both sensors are placed on the desk with their X' intensity's axis manually pointed towards geographical north.

Table I shows the average magnetic field intensity of the first test. Test results show that magnetic field intensity measurements are not the same for both sensor platforms. This can be expected, as both sensors have a unique electronic and metal composition, which might distort sensor readings. These distortions are called hard iron effects and are caused by the internal structure of the sensor. Compensation for these hard iron effects is needed. If no compensation for hard iron effects is performed and we use a different sensor for both offline training and online localization phase, we might have an inconsistency between the two data sets. Thus, compensating for hard iron effects is crucial for geomagnetic indoor localization.

Hard iron characteristics can be found by rotating the sensor around its x', y', and z' axis. These axes are defined relative to the sensor's reference frame, hence the apostrophe, and

Table I. AVERAGE MAGNETIC FIELD INTENSITY FOR BOTH SENSORS DURING STATIC TEST. VALUES ARE EXPRESSED IN µT.

Intensity	Shimmer	Smartphone
X'	-1.05	0.50
Y'	7.34	19.19
Z'	-57.61	-41.50
F	58.09	42.32

can be found in its documentation. If no hard iron effects are present, rotating a magnetometer 360 degrees and plotting the resulting data as y' axis versus x' axis, will result in a circle centered around the origin. Figures 2 and 3 show the resulting circles of rotating the Shimmer and the smartphone sensor in the $x'y'$ plane, before and after compensating for hard iron effects. Table II shows the compensation values for each axis of both sensors.

After compensating both sensors for these hard iron effects by subtracting the compensation values from the raw data, the first test is repeated. The results are shown in Table III. We can see that both sensors give very similar measurements at the same position.

Often, magnetometers are also calibrated to compensate for the presence of external metal or electronic distortions, called soft iron effects. For this research, this is an undesired calibration as the goal of geomagnetic localization is to measure and map these distortions.

If we do not look at the previous test data, we expect the smartphone to have a higher variance because of its more advanced electronic composition, which might influence the sensitive magnetometer. We note that the shimmer sensor has a slightly larger variance, which is unexpected. Additional tests are conducted with all receivers of the smartphone turned on, in an attempt to maximize the variance. Table IV shows the magnetic field intensity measurements of the smartphone with receivers disabled and enabled. Note that the Bluetooth

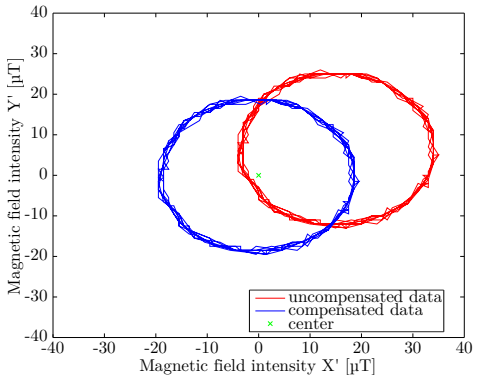


Figure 2. Shimmer hard iron compensation.

Table II. HARD IRON COMPENSATION FOR BOTH SENSORS. VALUES ARE EXPRESSED IN µT.

Correction	Shimmer	Smartphone
X'	15.00	3.67
Y'	7.25	0.16
Z'	-11.25	4.52

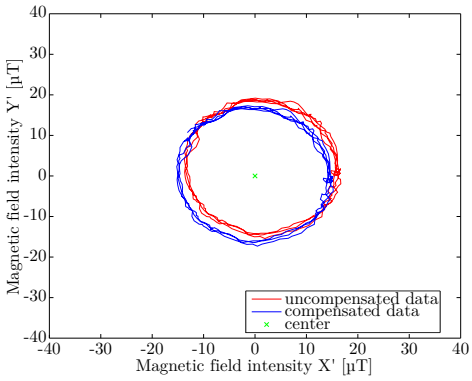


Figure 3. Smartphone hard iron compensation.

Table III. AVERAGE MAGNETIC FIELD INTENSITY AND CORRESPONDING STANDARD DEVIATION FOR BOTH SENSORS DURING STATIC TEST, AFTER HARD IRON COMPENSATION. VALUES ARE EXPRESSED IN µT.

Intensity	Shimmer		Smartphone	
	μ	σ	μ	σ
X'	-0.33	0.56	-0.80	0.48
Y'	17.12	0.52	18.01	0.51
Z'	-45.97	0.57	-44.46	0.51
F	49.06	0.57	47.98	0.52

receiver is enabled in both scenarios as it is used to send back the data to the computer. Although the variance in the data rises slightly when both receivers are activated, it does not significantly affect our measurements.

As the focus of the localization research is handheld smartphones, tests are conducted to see if human hand contact would significantly affect the measurements. During the offline calibration phase measurements can be taken either with or without contact by a human hand. Magnetic field intensity measurement are taken with and without contact by a human hand, without changing the position of the hand. The results of the 200 samples are presented in Table V. The test results show that there was no significant change between both scenarios.

Table IV. AVERAGE MAGNETIC FIELD INTENSITY AND CORRESPONDING STANDARD DEVIATION FOR THE SMARTPHONE MAGNETOMETER, WITH OR WITHOUT ADDITIONAL ELECTRONIC ACTIVITY. VALUES ARE EXPRESSED IN µT.

Intensity	Wi-Fi and GPS disabled		Wi-Fi and GPS enabled	
	μ	σ	μ	σ
X'	15.07	0.47	15.14	0.48
Y'	2.35	0.43	2.43	0.54
Z'	-32.94	0.49	-32.91	0.55
F	36.31	0.50	36.32	0.56

Table V. AVERAGE MAGNETIC FIELD INTENSITY AND CORRESPONDING STANDARD DEVIATION FOR THE SMARTPHONE MAGNETOMETER, WITH OR WITHOUT HUMAN HAND CONTACT. VALUES ARE EXPRESSED IN µT, DIFFERENCES IN %.

Intensity	No hand contact		Hand contact		Difference	
	μ	σ	μ	σ	$\Delta\mu$	$\Delta\sigma$
X'	14.33	0.55	14.48	0.48	98.96	114.58
Y'	1.21	0.50	0.93	0.55	130.11	90.91
Z'	-33.80	0.52	-33.32	0.52	101.44	100.00
F	36.74	0.54	36.35	0.51	101.07	105.88

3) Indoor magnetic field intensity: While indoor environments pose good candidates for geomagnetic localization, magnetic field intensity measurements must be stable over long periods of time. [10] conducted experiments where indoor magnetic field intensity was measured in different environments. The results show stable magnetic field intensity measurements over a 24 hour period. The experiments are repeated three months later, and no significant change was detected.

To achieve indoor localization, it is important that magnetic field intensities change considerably from position to position. If the magnetic field intensity measurements do not change considerably, the fingerprint might not contain enough information to overcome the cumulative error of the estimated position and indoor localization cannot be achieved [2].

A dynamic test is performed to see if magnetic field intensity measurements vary over the length of two hallways. The Shimmer sensor is placed on an office chair and is elevated to a height of 1.2 m. This height is similar to a person holding a smartphone. The elevation also made sure there is as little interference as possible from the chair itself.

The chair is moved at a constant velocity of 0.3 cm/s through the hallway. The speed is not always constant as human error is inevitable. The first hallway is expected to have changing measurement values because of the reinforced concrete floor and metal furniture in the rooms next to the hallway. The second hallway is expected to have less varying measurements because of the wooden floor and the absence of metal furniture.

Figures 4 and 5 show the measurements of the X', Y', and Z' intensities taken through respectively the first hallway and the second hallway. The test results show changing magnetic field intensity measurements for hallway A. These peaks and drops in magnetic field intensity allow us to identify certain areas inside the hallway and accordingly allow for localization. The measurements of hallway B tell a different story. Since there are no distinct fluctuations to identify certain areas, accurate localization seems improbable.

Additionally, indoor environments are places where objects are often moved or replaced. This will result in changes in the magnetic field intensity maps, decreasing localization

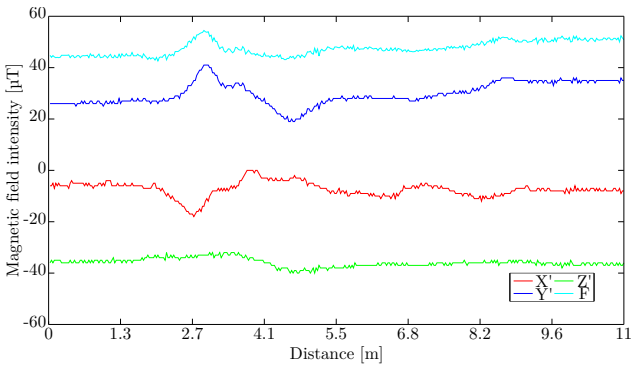


Figure 4. Magnetic field intensity dynamic test of hallway A.

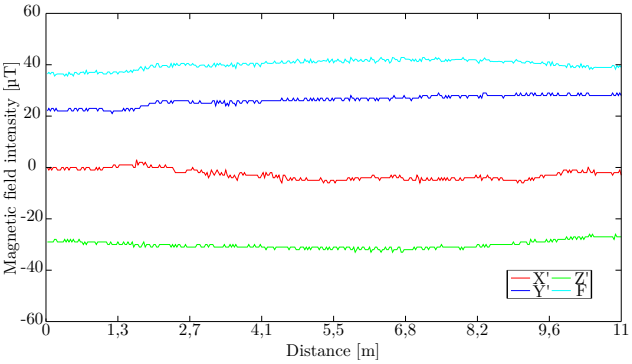


Figure 5. Magnetic field intensity dynamic test of hallway B.

performance, as discussed below.

Tests are conducted to investigate these interferences. Three objects are tested: a perforator, a mobile phone, and a hard drive. These objects are chosen because they can represent normal household objects which are often moved within an indoor environment. These differently sized objects are moved at a constant speed towards a Shimmer sensor to investigate the range and magnitude of the interferences. Figure 6 shows the results of the hard drive test. The hard drive is moved closer to the sensor at a constant speed, reaching the sensor after 50 s. Magnetic field intensity changes drastically as the hard drive moves closer to the sensor. As can be expected the change in magnetic field intensity was less significant for the smaller objects. Table VI shows the interference range of all objects.

Test results show that the size and magnetic composition of the object determines the range and magnitude of the interference. Small sized objects only caused interference starting from a range of about 15 cm, while larger objects

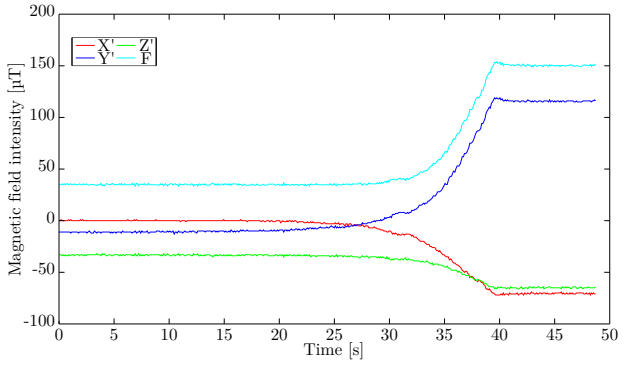


Figure 6. Metal and electronic object interference test of the hard drive.

Table VI. METAL AND ELECTRONIC OBJECT INTERFERENCE TEST RESULTS.

Object	Perforator	Phone	Hard drive
Average velocity [cm/s]	1.48	1.66	1.37
Start of interference [s]	29.00	24.00	23.00
Interference range [cm]	12.00	15.10	23.50

cause interference starting from about 25 cm. Small objects have a negligible influence for a room sized environment, yet the interference of larger objects cannot always be ignored.

B. Fingerprinting

In radio frequency (RF) based localization, fingerprinting is performed by measuring a pattern of RF signals and matching them to a database of such measurements. These measurements are called RF fingerprints, and consist of all pairs of received signal strength (RSS) value and media access control (MAC) address that can be seen at a certain location. This idea was originally published in [26].

Fingerprinting localization has an offline and an online phase. During the offline phase, fingerprints are recorded at reference locations and stored in a database. These fingerprints together form a radio map of the environment, which is used during the online phase. During this online phase, devices that need to be localized measure fingerprints at their location and compare these fingerprints with the radio map in the database. Due to measurement noise and fluctuations in RF signals, these fingerprints are usually not exactly the same as fingerprints in the the database, so a set of measurements is used to estimate a true location. This method is described in more detail in [3].

As shown by [8], this localization technique can be directly applied to magnetic field localization. Instead of RF fingerprints, magnetic field fingerprints are used, by measuring the X, Y, and Z intensities of the geomagnetic field vector $\boldsymbol{B}_m$, as explained above.

As described in [1], magnetic field intensity maps were created by measuring the magnetic field intensity at predefined locations. The sensor remained still during these measurements.

Three different locations are chosen for experimentation: the ground floor of a suburban house, with an area of 14×16 m; the second floor of a city centered apartment, with an area of 9×12 m; and the second floor lab at the university campus, with an area of 6×19 m. These locations are chosen because they represent distinct environments where indoor localization might be required. It is important that all locations have multiple rooms and are medium to large size, i.e., above 20 m^2. Figure 7 shows the recorded fingerprints of the suburban house. A slash is drawn through areas where no fingerprint measurement could be obtained because of built in cabinets, wardrobes or other furniture. For simplicity, the color map shows only the magnetic field F intensity measurements taken at one meter spacing. We do not explicitly research the maximum accuracy of geomagnetic localization for this feasibility research.

The fingerprint in Figure 7 shows that the magnetic field intensity characteristics change from position to position. There is a big metal stove located between the dining room and the kitchen. We measured a high magnetic field intensity at that location, which results in a light square. A test is done to determine if these characteristics are unique for an indoor environment. A fingerprint is created in a garden, with an area of 4×6 m, and in a small part of a street, with an area of

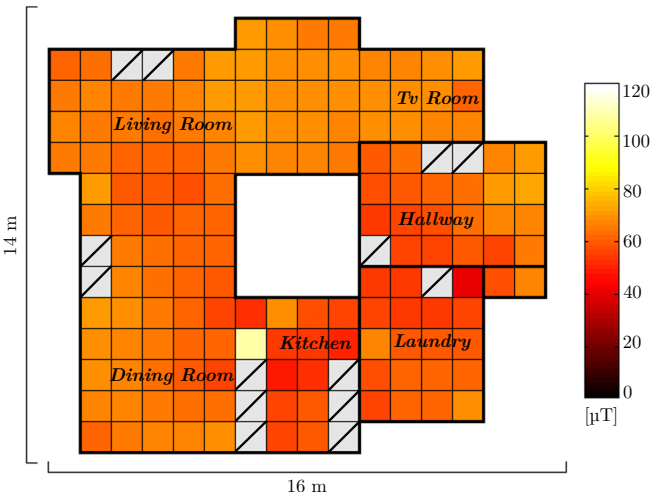


Figure 7. Magnetic field F intensity fingerprint map of the ground floor of the suburban house.

5×15 m. Figure 8 shows the fingerprint obtained at the street. The outdoor results are very different from the indoor results. The magnetic field intensities do not change significantly with position. Tables VII and VIII show the magnetic field intensity standard deviation of the recorded measurements for both the indoor and the outdoor fingerprints. The indoor environments clearly have more varying measurements than the outdoor environments.

Fingerprint maps are also created to confirm the findings on metal and electronic objects' interference mentioned above. A fingerprint is taken from a small bedroom with an area of 3.5×3.5 m. Magnetic field intensity measurements of the X', Y', and Z' intensity are taken at 0.5 m spacing. Figure 9 shows the interior setup of the room and the resulting magnetic field intensity fingerprint of the F intensity. As it can be seen form this fingerprint, the two speakers cause a clear magnetic field intensity interference pattern. The size of this distortion

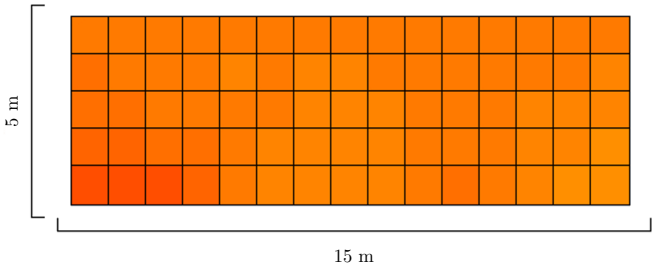


Figure 8. Magnetic field intensity fingerprint map of a part of a street. The intensity color scale is the same as in Figure 7.

Table VII. MAGNETIC FIELD INTENSITY STANDARD DEVIATION VALUES FOR INDOOR LOCATIONS. VALUES ARE EXPRESSED IN µT.

Intensity	House	Apartment	Lab
X'	5.70	5.49	6.99
Y'	4.63	5.52	4.84
Z'	5.11	4.65	8.08

Table VIII. MAGNETIC FIELD INTENSITY STANDARD DEVIATION VALUES FOR OUTDOOR LOCATIONS. VALUES ARE EXPRESSED IN μT.

Intensity	Garden	Street
X'	1.45	3.53
Y'	1.28	3.30
Z'	0.55	2.65

is rather large, as speakers are often constructed with strong magnets inside of them. After creating the first fingerprint map, one of the speakers was moved to a different location within the room. Subsequently, a new fingerprint map was created.

Figure 10 shows the new interior setup and the resulting new fingerprint. The interference pattern of the moved speaker is clearly visible in the new fingerprint. These test results give an example of how the repositioning and removal of objects inside a room can form an obstacle for indoor geomagnetic localization. When the interior setup of a room changes significantly, a new fingerprint should be taken. Of course, a SLAM algorithm could perform continuous mapping of the environment, while simultaneously performing localization.

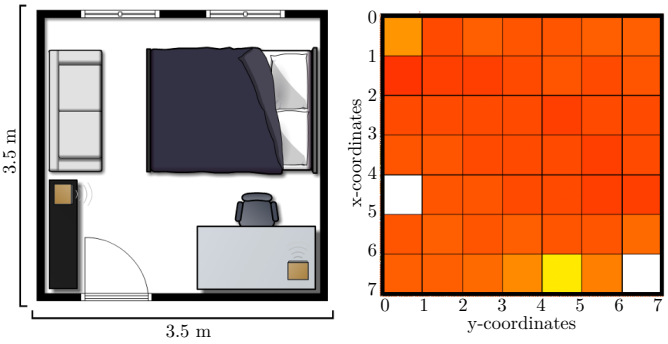


Figure 10. Magnetic field intensity fingerprint of bedroom, with the speaker on its new position. The intensity color scale is the same as in Figure 7.

C. RatSLAM

RatSLAM is a biologically inspired SLAM algorithm, modeled after spatial cognition in rats [27, 28]. It consists of three elements, called the local view network, the pose cell network, and the experience map, as shown in Figure 11. We will give a brief overview of the pose cell network and the experience map, and refer the reader to [28] for additional details. We do present a more in-depth discussion of the local view network as this is the only component that is modified for this research.

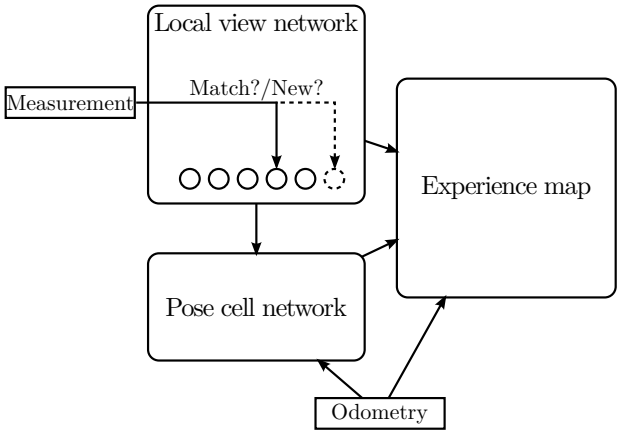


Figure 11. RatSLAM consists of three elements, called the local view network, the pose cell network, and the experience map.

1) Pose cell network: The pose cell network is a three dimensional continuous attractor network (CAN) [29, 30], representing pose consisting of position in the plane (x, y) and orientation (θ). The activity pattern in this network represents the local pose estimate, or estimates if the pose is ambiguous. It can be visualized as a cube in which activity packets are created, moved around and destroyed. The connectivity pattern between the nodes in a CAN is such that activity packets can be considered as discrete blobs of activity that keep their shape when moved around the network. The activity packets in the pose cell network are moved in accordance with the odometry information. The boundaries of the network are wrapped around, so that an activity packet that reaches the end of the network is wrapped back to the start. Hence, multiple positions and orientations in the real world are mapped upon the same nodes in the pose cell network. Extra activity is injected by the local view network, i.e., new activity packets are created when new sensor measurements correspond with memorized sensory signatures, as explained below.

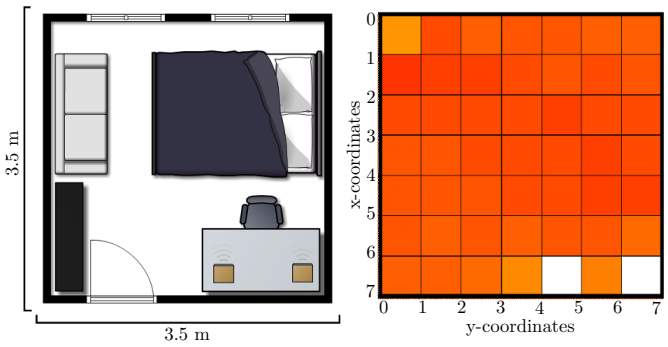


Figure 9. Magnetic field intensity fingerprint of bedroom, with the speaker on its original position. The intensity color scale is the same as in Figure 7.

2) Experience map: The experience map is a graph that represents a global topological map of the environment, based on information from the local view network, the pose cell network, and the odometry information. It creates new nodes based on the state of the local view network and connects them with new edges to nodes already present in the experience map by using the metric odometry information. This information is continuously updated on the basis of new sensor data, the state of the local view network and activity in the pose cell network. Hence, this topological map acquires semi-metric properties, i.e., progressively more accurate (x, y) coordinates are associated with the nodes that lie on paths that have been repeatedly traveled. A detailed overview of the functioning of the experience map can be found in [31].

3) Local view network: The local view network acts as a database of scenes that have been observed during the exploration of the environment. The measurement (see Figure 11) contains the sensor information about the current scene; typically, the measurement is a camera image. When a new measurement is taken, it is compared with previous measurements as stored in the local view templates associated with the local view cells. The activation of each local view cell depends on the quality of this match. If the smallest difference between the measurement and the local view cells is greater than a certain threshold, the measurement is said to correspond with an as yet unobserved scene. In that case, a new local view cell is created and the measurement is copied into the associated local view template. The local view cell is also linked to the position of the activity packet, i.e., the local pose estimate, in the pose cell network that is at that time the dominant packet. Alternatively, if the difference between the measurement and the local view cells is smaller than the threshold, a match is said to be found with the local view cell that had the smallest difference with the measurement. In this case, the local view network will inject activity into the pose cell network at the pose linked previously with this local view cell.

When traveling through previously mapped terrain, a sequence of familiar scenes will be observed by the sensor. However, if the state of the pose cell network differs too much from the true position this will result in the creation of a new activity packet. Indeed, activity being injected in the particular order at the particular places in the pose cell network corresponding with this sequence of familiar scenes will effectively increase the activity in the newly created activity packet. Subsequently, this activity packet will become the strongest and the specific mechanics of the pose cell network will suppress the old activity packet. This mechanism avoids cumulative build-up of odometric errors in the pose estimate when traveling through familiar terrain. Again, more details can be found in the literature [17, 18, 27, 28, 31].

The typical camera image that serves as measurement in the original RatSLAM implementation has been replaced by several other sensor modalities: a biomimetic sonar system modeled after the echolocation abilities of bats [19, 20], the fusion of a laser range finder, a simple sonar array, a depth camera and a normal camera [21]. In this research, we propose to replace the camera images with magnetic field fingerprints as used in the magnetic field localization system.

III. Method

In this section, we will discuss the methods used to perform both geomagnetic localization and geomagnetic RatSLAM. Slightly updated, the section on geomagnetic localization is also presented in [1].

A. Geomagnetic localization

The magnetic field intensity results discussed above suggest that magnetic field intensity measurements can be used to achieve indoor localization. It is important to note, however, that the quality of the localization often depends on the number of measured components that can be used as reference points [2]. Having many values to compare against can obviously increase the chance of identifying the actual position. The number of components that can be recorded by a magnetometer is rather small. Only the X', Y', and Z' intensities of the earth's magnetic field, in the reference frame of the magnetometer, can be measured. There are some practical consequences to be considered during the localization and fingerprinting phase when using these three intensities.

As stated before, a magnetometer will measure the magnetic field intensities relative to its own orientation. So, to use the three intensities requires that the orientation of the sensor is exactly the same during the fingerprinting and the localization phase. This is a requirement that cannot be met easily. A user will walk around in different directions and the orientation of the device will follow along with him. The way the user holds the device is also not always the same. Determining the orientation of the device will be key to using all three components. If no information is available about the orientation of the device in none of these two phases, we can only use the F intensity. This would reduce the number of components to be used for localization to only one.

To resolve this issue, a tilt compensated magnetometer can be used. Such a magnetometer uses accelerometers to detect the vertical orientation of the device by measuring the force of the earth's gravity. Using tilt compensation allows us to use two components, the Z intensity and the H' intensity [10]. To use all three components the horizontal orientation of the device needs to be known as well. To determine the horizontal orientation, the magnetometer can be used as a compass. A compass can determine the direction of the magnetic north, and can consequently determine the horizontal orientation of the device. To do this, the user has to manually point the device to a reference point on the map, e.g., geographic north. By defining a reference point the horizontal orientation can be determined.

This research shows, however, that indoor environments can cause interference in the magnetic field intensity measurements. These interferences are called soft-iron effects. Compensation has to be done to remove these interferences to get an accurate heading. It is important to note, that when soft-iron compensation is done, there needs to be a clear distinction between the compensated data and the raw data. Orientation requires soft-iron compensation while localization requires no soft-iron compensation.

All aforementioned information can be combined to define a measurement model for geomagnetic indoor localization. Defining a measurement model can provide a technology interface for sensor fusion systems [3]. Algorithm 1 describes the measurement model. The measurement model is used to find the probability of a position $\boldsymbol{x}_t = \{x_t, y_t\}$, where x and y are spatial coordinates, given a measurement $\boldsymbol{z}_t$, which can be any of:

$$\boldsymbol{z}_t = \{z_t^X, z_t^Y, z_t^Z\} \quad (3)$$

$$\boldsymbol{z}_t = \{z_t^{H'}, z_t^Z\} \quad (4)$$

$$\boldsymbol{z}_t = \{z_t^F\} \quad (5)$$

where X, Y, and Z indicate intensities of the magnetic B field. Equation (3) can be used when both tilt compensation and heading compensation are performed. Equation (4) can be used when only tilt compensation is performed. Equation (5) can be used when no compensation is performed. The algorithm uses a Gaussian kernel distribution $p(z_t^k|\boldsymbol{x}_t)$:

$$p(z_t^k|\boldsymbol{x}_t) = \exp(\frac{z_t^k - z_{dB}^k}{2 * \sigma^2}) \quad (6)$$

where $\boldsymbol{z}_{dB}$ is the fingerprint in the database corresponding with $\boldsymbol{x}_t$ and k is any of X, Y, Z, H', or F.

Algorithm 1 Geomagnetic measurement model $(\boldsymbol{x}_t, \boldsymbol{z}_t)$.

```
function CALCULATEWEIGHT(x_t, z_t)
    if z_t == {z_t^X, z_t^Y, z_t^Z} then
        w_X = p(z_t^X|x_t)
        w_Y = p(z_t^Y|x_t)
        w_Z = p(z_t^Z|x_t)
        return w_X · w_Y · w_Z
    else if z_t == {z_t^H', z_t^Z} then
        w_H' = p(z_t^H'|x_t)
        w_Z = p(z_t^Z|x_t)
        return w_H' · w_Z
    else if z_t == {z_t^F} then
        w_F = p(z_t^F|x_t)
        return w_F
    end if
end function
```

Although magnetic field intensity measurements remain stable over long periods of time, big, moving metal objects like an elevator cause variations in these measurements. These sources of errors can cause a mismatch in the magnetic field intensity measured at the same position. The accumulated error can be modeled as a Gaussian kernel distribution. The standard deviation of this distribution has to represent the maximum variation that can be expected. The standard deviation σ was set to $2\,\mu T$ as this was the maximum standard deviation reported at $2\,m$ from an elevator by [10].

B. Geomagnetic RatSLAM

We use our Pioneer 3DX mobile robot to collect measurements and to provide a reliable odometry source for our geomagnetic RatSLAM implementation. The Pioneer 3DX serves as robot platform, with a consumer grade laptop mounted on top to save the measurements. Elevated by a cardboard box, we place our sensor at a safe distance to avoid soft iron interference in the magnetic field caused by the metal parts of the robot. The setup is shown in Figure 12. The robot is also equipped with a laser range finder to serve as a comparison tool using an established laser range finder based SLAM method.

The local view network of the original RatSLAM algorithm uses camera images to recognize scenes. This functionality has to be replaced with an algorithm capable of recognizing magnetic field intensity measurements. Assuming that our magnetometer will always be in the same position relative to the robot, we can simplify the measurement model. Similar to how a 60° angle of view camera can only observe one direction at a time, we choose to match only to magnetic field intensity measurements that are oriented in the same way during initial measurement and during subsequent comparison. In other words, our algorithm will not attribute a high match quality to the measurements from one location when it is being traversed in a different orientation. This results in $\boldsymbol{z}_t$ to have only one option:

$$\boldsymbol{z}_t = \{z_t^{X'}, z_t^{Y'}, z_t^{Z'}\} \quad (7)$$

which is different from Equations (3), (4), and (5) by always using the magnetometer reference frame defined by X', Y', and Z', which is allowed since we only want matches when the magnetometer has the same orientation for the measurements being compared. In fact, we can assume $Z' = Z$, since Z' is always oriented to nadir. We do not explicitly model it that way, however, so that our approach is more general even for differently oriented magnetometers. The same Gaussian kernel distribution $p(z_t^k|\boldsymbol{x}_t)$ as in Equation (6) is used, with $\sigma = 0.67\,\mu T$. This difference in standard deviation is created to be more selective in matching local view cells.

We use the Robot Operating System (ROS, [32]) as a framework to read magnetometer messages from the smartphone and to operate the robot. The freely available OpenRatSLAM source code [31] is modified to create RatSLAM results.

IV. Result

Here, we will present the results we have obtained for both geomagnetic localization and geomagnetic RatSLAM. Firstly, the geomagnetic localization results are shown, originating with slight modification from the original paper [1]. Next, we present the new geomagnetic RatSLAM results.

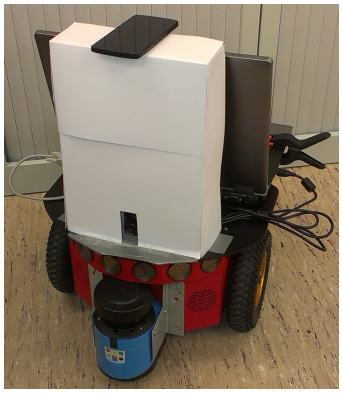


Figure 12. The Pioneer 3DX mobile robot platform.

A. Geomagnetic localization

The measurement model in Algorithm 1 is used to investigate the feasibility of geomagnetic indoor localization. To test the feasibility we use the suburban house, the apartment, and the university lab as experimental environments. Each individual fingerprint position and its accompanying magnetic field intensity measurement is used as a test position. Each test position is compared to all measurement positions in the fingerprint using the measurement model described in Algorithm 1. The measurement model will give a high weight to fingerprint positions that had magnetic field intensity measurements similar to the test position. The weight represents the likelihood of the sensor reading $\boldsymbol{z}_t$ given the position $\boldsymbol{x}_t$. The final estimated position $\boldsymbol{x}_t$ is calculated as the weighted average of all fingerprint positions, using Equation (8). Positions with a high probability will contribute more to the final estimated position [3].

$$\boldsymbol{x}_t = \frac{\sum_{i=1}^{N} w_t^{[i]} \cdot x_t^{[i]}}{\sum_{i=1}^{N} w_t^{[i]}} \tag{8}$$

where N is the number of positions.

The coordinates of the final estimated position are compared to the real coordinates of the test position and the error is stored. The process will be repeated for all measurement positions within the fingerprint. The maximum, minimum and average errors for every location are determined. The amount of estimated positions that are within 1 m and the amount of estimated positions that are in the same room is also determined. Table IX shows the results that are obtained from the three fingerprints that are recorded.

It is clear from the results that using three components gives the best localization results and results deteriorate when fewer components are used. The maximum and minimum errors stay relatively the same for all amounts of components. All localization results are combined to form a cumulative density function in Figure 13.

This test is repeated, with this difference that the room of each test position is known. Table X shows the results of this test. Only measurement positions in the same room as the test positions are compared to the test position. Test results improve significantly, so that even using only one component, localization close to 1 m can be achieved. These results indicate that geomagnetic localization might be more suited for localization within a room. Figure 14 show the cumulative density function when the room is known.

Table IX. GEOMAGNETIC LOCALIZATION FEASIBILITY RESULTS FOR DIFFERENT ENVIRONMENTS, USING ONE, TWO, OR THREE COMPONENTS IN THE MEASUREMENTS MODEL.

Properties	Suburban house			Apartment			Lab		
	1	2	3	1	2	3	1	2	3
mean [m]	4.8	4.3	3.1	3.7	3.3	2.5	4.5	3.4	2.5
min [m]	0.1	0.0	0.0	0.1	0.1	0.0	0.2	0.0	0.0
max [m]	9.3	9.4	8.8	7.2	7.4	7.0	10.5	11.3	11.8
$<$ 1 m [%]	4.0	9.0	17.0	7.0	13.0	23.0	9.0	20.0	32.0
room [%]	10.0	18.0	44.0	21.0	31.0	49.0	73.0	74.0	82.0

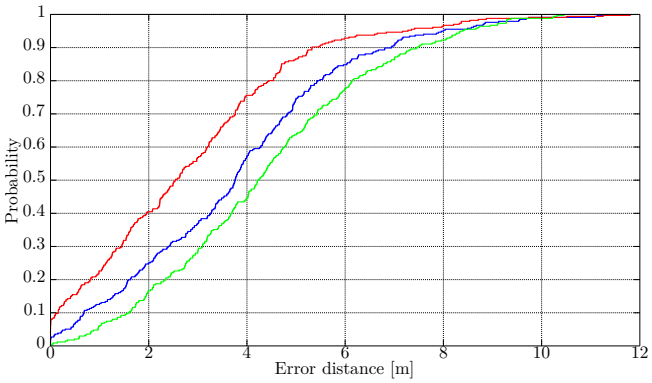


Figure 13. Cumulative density function of the error of localization. The green line is using one component; the blue line is using two components; the red line is using three components.

Table X. GEOMAGNETIC LOCALIZATION FEASIBILITY RESULTS WHEN THE ROOM IS KNOWN FOR DIFFERENT ENVIRONMENTS, USING ONE, TWO, OR THREE COMPONENTS IN THE MEASUREMENTS MODEL.

Properties	Suburban house			Apartment			Lab		
	1	2	3	1	2	3	1	2	3
mean [m]	1.4	1.0	0.8	1.4	1.0	0.6	2.2	1.4	0.9
min [m]	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
max [m]	3.2	2.8	2.3	5.2	3.9	2.1	5.4	5.2	3.7
$<$ 1 m [%]	30.0	47.0	67.0	35.0	50.0	74.0	21.0	42.0	61.0

Although previous results give a good indication of how feasible geomagnetic indoor localization can be, they are largely theoretical. To verify these findings, a more practical test is performed. A route is recorded through the suburban house. On this route, magnetic field intensity measurements are taken at roughly the same positions as where fingerprint measurements are taken. The position can not be exactly the same as human error is inevitable. Figure 15 shows the recorded magnetic field intensity for the route and the fingerprint.

The results show that the recorded measurements are not exactly the same, however, the average correlation coefficient between the route and the fingerprint X, Y, and Z

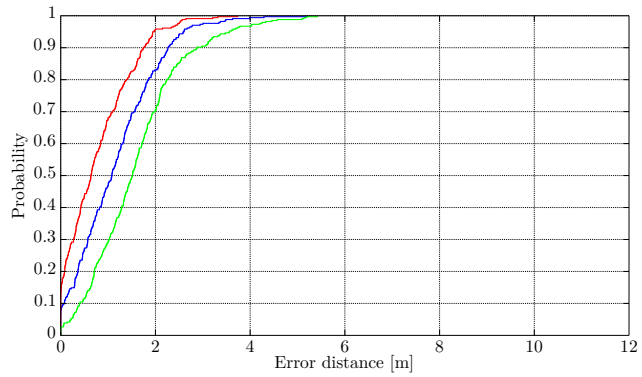


Figure 14. Cumulative density function of the error of localization when the room is known. The green line is using one component; the blue line is using two components; the red line is using three components.

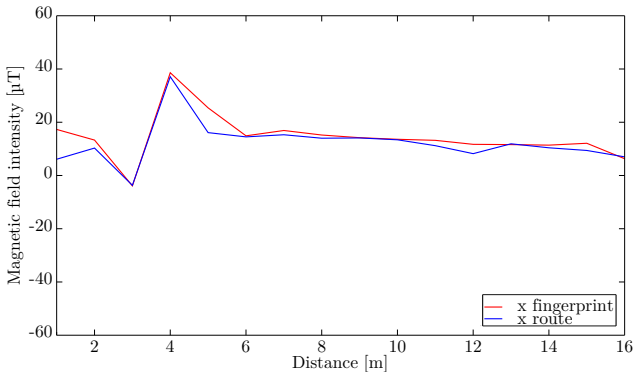


Figure 15. Magnetic field intensity X measurements from fingerprint map and during route.

measurements is 0.93, which means that both recordings are very similar. The recorded route is estimated within the environment using Algorithm 1. Figure 16 shows the original route in blue, and the estimated route in green. First the route is estimated when nothing about the room is known, later the route is estimated when the room of the measurement is known. Table XI shows the localization results of both scenarios.

This practical test confirms the original findings. Localization is very dependent on the amount of components that can be used, and results are superior when only room sized localization is required.

B. Geomagnetic RatSLAM

Our geomagnetic RatSLAM results are collected in the same university lab as the geomagnetic localization results. We do not provide the algorithm with any information about the specific room in which measurements are taken. A point to point quantitative ground truth is not available for the results, but we show the path produced with only odometry and the path produced by Grid Mapping as qualitative comparisons. The odometry only path is expected to perform much worse, as it has no sensor information to recognize familiar scenes. The Grid Mapping algorithm is freely available in the ROS framework and described in [33]. This path is expected to perform better than our own estimation, since it utilizes the laser range scanner high precision data.

Four separate datasets are constructed. Three of them are simple runs up and down the lab, starting and ending in different rooms, lasting about 15 min each. A fourth dataset drives up and down to different rooms in different order, lasting about 30 min. The first three are used to find the correct RatSLAM parameters, by training on two of them and checking on the third, switching datasets for every parameter setup. This approach does not guarantee that optimal parameters are found, but decreases the chance of overfitting the parameters. To additionally prevent overfitting, the fourth dataset is used as final check. Figure 17 shows a schematic overview of these runs, drawn against the output of the Grid Mapping algorithm applied to the fourth dataset.

The traveled path is the general output of the RatSLAM algorithm. This will be discussed further on for the fourth, challenging dataset, however, our focus was on the local view network. These results are generally discussed using local view cell matching diagrams, which are diagrams on which the local view cell identification number, or template ID, is drawn as a function of time. Horizontally, the first time a template ID is encountered is when the local view cell is created. Subsequent occurrences of the same template ID indicate when the local

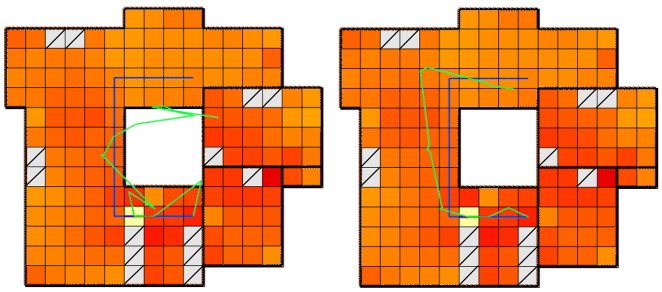


Figure 16. Suburban house route results for three components. Left hand side, when the room was not known. Right hand side, when the room was known. The blue line is the recored route and the green line is the estimated route. The intensity color scale is the same as in Figure 7.

Table XI. ROUTE ESTIMATION RESULTS.

Properties	Global			Room known		
	1	2	3	1	2	3
mean [m]	2.1	2.1	1.4	1.3	1.1	0.9
min [m]	0.2	0.1	0.0	0.0	0.0	0.0
max [m]	4.22	4.2	4.5	3.1	2.6	2.0
< 1 m [%]	13.0	13.0	38.0	43.0	43.0	62.0
room [%]	25.0	31.0	56.0	N/A	N/A	N/A

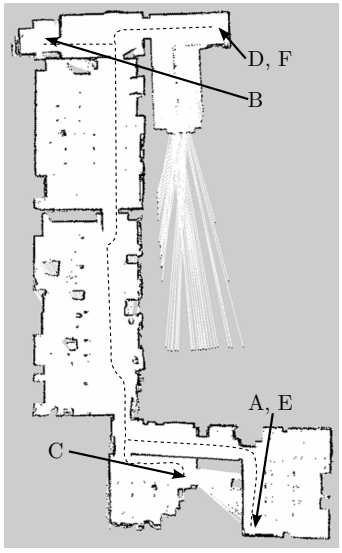


Figure 17. The second floor university lab laser map. The first three datasets consist of three runs going up and down the lab. The first dataset went from A to B and back; the second dataset went from C to D and back; and the third dataset went from E to F and back. The fourth dataset went to all rooms in different order. The general trajectory is shown as a dashed line.

view network decided a measurement to originate from a location encountered before. Similarly, the experience node identification number, or experience ID, is drawn, to indicate when exactly the experience map was informed to create a new location or to link to an existing location.

Figure 18 shows such matching diagrams for the first dataset, using a camera the first time and using magnetometer the second time as input. We observe a similar response in both sensors, where locations encountered while going up the run are found to be different from locations encountered while going down the run. As explained before, this is expected behavior. Three parallel, diagonal lines can be seen in the figure, once for view template matches and once for experience node matches. The first, left hand side of these lines indicates new local view cells or experience nodes being created. The second and third of these lines start when the robot reaches location A again, as indicated in Figure 17, on the 300th and 600th second mark. These lines indicate matches to the originally created view templates or experience nodes. False positive matches are labeled FP on the figure. They can be found between the lines created by correct matches to the originally created view templates.

We also note many more false positive local view cell matches when using the magnetometer. This is explained by the RatSLAM parameters, where we have chosen a local view cell matching threshold for the magnetometer that favors matches, including some false positive matches, above one that finds few matches but avoids finding false positive matches. This parameter setting improves experience node matching, facilitating effective loop closure. A camera does not need this coarse matching threshold, because it has a sample rate six times higher than our magnetometer. Matches will be reported many times more often than with the magnetometer.

The other two initial datasets showed similar results, so we can test the obtained parameters on our more challenging fourth dataset. An overview of our parameters can be found in Table XII. A new local view cell will be created when the weight calculated by our measurement model is lower than the match threshold. The recency threshold prevents the local view network from matching new measurements with recently created local view cells. In other words, with a magnetometer frequency of about 5 Hz, the last eight seconds of measurements are ignored when creating a match. The dimension of the pose cell network is increased to further cope with the false positive local view cell matches. The pose cell injection energy is how much energy is injected into the pose cell network on each local view match. Other parameters are left on their default values. A detailed discussion of these parameters when using the OpenRatSLAM system can be found in [31].

Figure 19 shows the map created using only odometry, using our geomagnetic RatSLAM, and using Grid Mapping. The raw integrated odometry in Figure 19a shows some structure of the environment when observed carefully, however, it can in no way be used for either localization or navigation. The trajectory created using Grid Mapping in Figure 19c shows clearly what path has actually been followed by our robot. Do note that this trajectory was created using a high precision laser range finder, in contrast to our simple smartphone magnetometer. The trajectory created using the magnetometer is shown in Figure 19b. It can be divided into two parts, the coarse lower part and the precise upper part. The lower part of the run has fewer experience node matches, i.e., loop closure, so that different traversals of the same location are not matched to each other. The upper part of the run has many more experience node matches, so that different traversals of the same location are matched to each other.

Table XII. PARAMETERS USED FOR THE GEOMAGNETIC RATSLAM.

Parameter	Value
Match threshold	0.5
Recency threshold	40
Pose cell xy dimension	37
Pose cell injection energy	0.03

This difference is supported by the template match diagram in Figure 18, where less experience node matches are seen in the region when the robot is near location A (as indicated on Figure 17), which is the 0th, 300th, 600th, and 900th second region. The regions in Figure 17 indicated by A, C, and E are located in a much older section of the building, with wooden floors and thin walls, without any offices. This causes the magnetic field intensity to be only slightly distorted in these regions. Consequently, geomagnetic localization in these regions is difficult.

The upper part of Figure 19b indicates a very precise operation of geomagnetic RatSLAM. This is a more modern region of the building, indicating that geomagnetic RatSLAM is feasible to use as SLAM mechanism in average indoor environments. The created experience map can subsequently be used for both localization and navigation tasks.

V. CONCLUSION AND FUTURE WORK

In this research, we show that the geomagnetic B field is feasible to use in both localization and SLAM applications. We first show an extensive review of platform and sensor independence when measuring the magnetic field intensity. Subsequently, we show localization feasibility with tests in various environments. Lastly, we demonstrate our newly developed geomagnetic input for the local view network of the RatSLAM system.

Our results indicate that geomagnetic localization and geomagnetic RatSLAM is strongly dependent on the environment. Environments with much magnetic field intensity distortion will allow more accurate localization and SLAM. Such environments are rather commonplace for indoor applications: most modern domestic or professional environments hold enough metal and electric devices to distort the geomagnetic field. Environments with little magnetic field intensity distortion will not provide enough information for accurate localization and SLAM.

We also note that geomagnetic localization performs better when used for localization within limited areas, such as rooms.

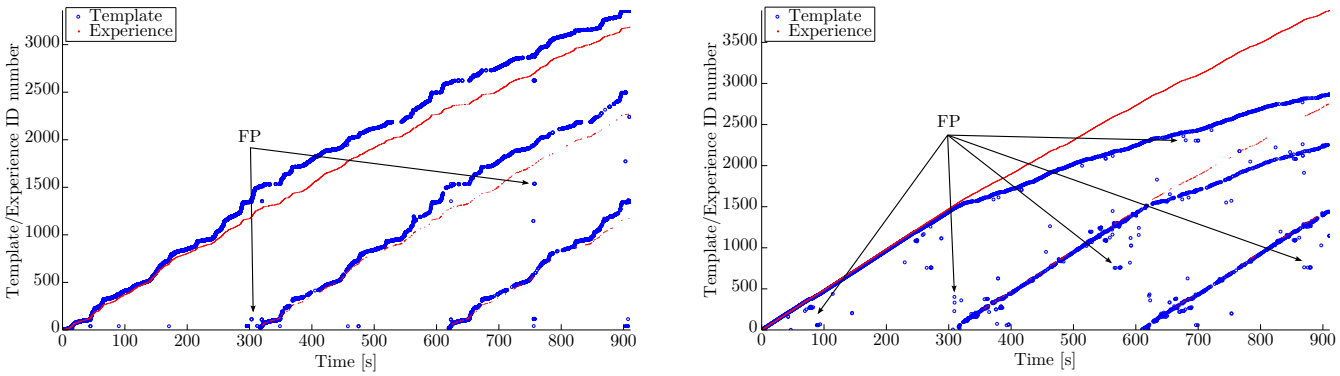


Figure 18. View template and experience matches for the first dataset: (a) shows the view template matches and experience node matches created using camera; and (b) shows the view template matches and experience node matches created using magnetometer.

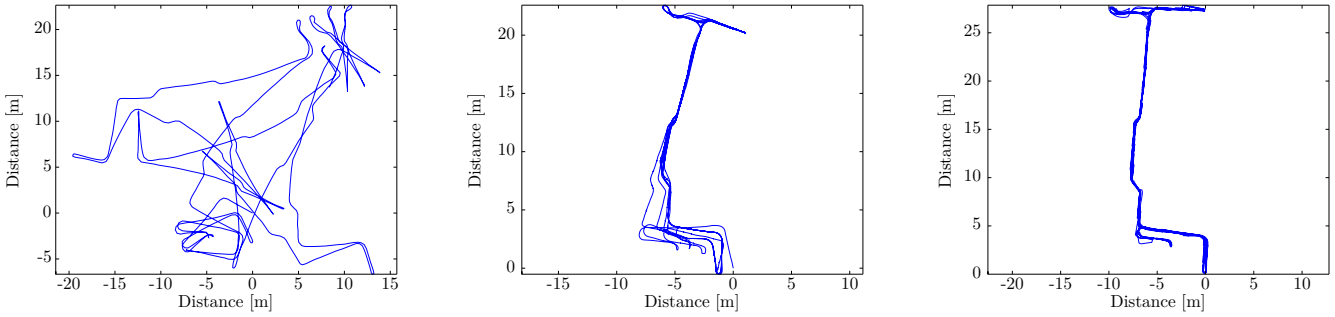


Figure 19. Traveled path for the fourth dataset: (a) shows the raw integrated odometery; (b) shows the experience map created by geomagnetic RatSLAM; and (c) shows the traveled path obtained from using the laser range finder based Grid Mapping algorithm available in ROS.

This suggests a complementarity with Wi-Fi as a localization system, which provides a rather coarse spatial localization and is used usually to locate up to room level [3]. In further research we will focus on fusing the virtues of these systems into the RatSLAM local view network. We will also look into fusing other electromagnetic sensors in the same system.

References

[1] D. Vandermeulen, C. Vercauteren, and M. Weyn, "Indoor localization using a magnetic flux density map of a building," in *The Third International Conference on Ambient Computing, Applications, Services and Technologies*. Porto: IARIA, 2013, pp. 42–49.

[2] R. Mautz, "Indoor Positioning Technologies," Ph.D. dissertation, Institute of Geodesy and Photogrammetry, 2012.

[3] M. Weyn, "Opportunistic Seamless Localization," Ph.D. dissertation, University of Antwerp, 2011.

[4] L. C. Boles and K. J. Lohmann, "True navigation and magnetic maps in spiny lobsters," *Nature*, vol. 421, no. 6918, pp. 60–3, Jan. 2003.

[5] H. Mouritsen, G. Feenders, M. Liedvogel, and W. Kropp, "Migratory birds use head scans to detect the direction of the earth's magnetic field," *Current biology*, vol. 14, no. 21, pp. 1946–9, Nov. 2004.

[6] S. Suksakulchai, S. Thongchai, D. Wilkes, and K. Kawamura, "Mobile robot localization using an electronic compass for corridor environment," in *SMC 2000 Conference Proceedings. 2000 IEEE International Conference on Systems, Man and Cybernetics. 'Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions' (Cat. No.00CH37166)*, vol. 5. Nashville, TN, USA: IEEE, 2000, pp. 3354–3359.

[7] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1028–1035, Oct. 2009.

[8] W. F. Storms, "Magnetic field aided indoor navigation," DTIC Document, Tech. Rep., 2009.

[9] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Proceedings of the 9th international conference on Mobile systems, applications, and services - MobiSys '11*. New York, New York, USA: ACM Press, 2011, p. 141.

[10] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*. Sydney, Australia: IEEE, Nov. 2012, pp. 1–9.

[11] M. Frassl, M. Angermann, M. Lichtenstern, P. Robertson, B. J. Julian, and M. Doniec, "Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, Nov. 2013, pp. 913–920.

[12] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[13] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[14] R. Madhavan and H. F. Durrant-Whyte, "Natural landmark-based autonomous vehicle navigation," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 79–95, 2004.

[15] M. Eich, R. Hartanto, S. Kasperski, S. Natarajan, and J. Wollenberg, "Towards Coordinated Multirobot Missions for Lunar Sample Collection

in an Unknown Environment," *Journal of Field Robotics*, vol. 31, no. 1, pp. 35–74, Jan. 2014.

[16] M. Montemerlo and S. Thrun, *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*, B. Siciliano, O. Khatib, and F. Groen, Eds. Springer Berlin Heidelberg New York, 2007, vol. 27.

[17] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired SLAM system," *The International Journal of Robotics Research*, vol. 29, no. 9, pp. 1131–1153, 2010.

[18] ——, "Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038–1053, Oct. 2008.

[19] J. Steckel, D. Vanderelst, and H. Peremans, "BatSLAM: combining biomimetic sonar with a hippocampal model," in *Proceedings of the Robotica Conference*, Guimaraes, Portugal, 2012, pp. 81–86.

[20] J. Steckel and H. Peremans, "BatSLAM: Simultaneous localization and mapping using biomimetic sonar." *PloS ONE*, vol. 8, no. 1, p. e54076, Jan. 2013.

[21] M. Milford and A. Jacobson, "Brain-inspired Sensor Fusion for Navigating Robots," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. Karlsruhe, Germany: IEEE, 2013, pp. 2891–2898.

[22] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning, "Simultaneous localization and mapping using ambient magnetic field," in *Multisensor Fusion and Integration for Intelligent Syst. (MFI), 2010 IEEE Conf. on*. IEEE, Sep. 2010, pp. 14–19.

[23] M. Angermann and P. Robertson, "FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors—Hitchhiking on Human Perception and Cognition," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1840–1848, May 2012.

[24] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. G. Puyol, M. Khider, M. Lichtenstern, and L. Bruno, "Simultaneous Localization and Mapping for Pedestrians using Distortions of the Local Magnetic Field Intensity in Large Indoor Environments," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*. Montbéliard-Belfort, France: IEEE, 2013, p. 10.

[25] S. Maus, S. Macmillan, S. McLean, B. Hamilton, M. Nair, A. Thomson, and C. Rollins, "The US/UK world magnetic model for 2010-2015," British Geological Survey, Edinburgh, UK, Tech. Rep., 2010.

[26] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.*, vol. 2. Tel Aviv, Israel: IEEE, 2000, pp. 775–784 vol.2.

[27] M. Milford, *Robot navigation from nature: Simultaneous localisation, mapping, and path planning based on hippocampal models*, B. Siciliano, O. Khatib, and F. Groen, Eds. Springer, 2008, vol. 41.

[28] G. Wyeth and M. Milford, "Spatial cognition for robots," *IEEE Robotics and Automation Magazine*, vol. 16, no. 3, pp. 24–32, Sep. 2009.

[29] S. M. Stringer, T. P. Trappenberg, E. T. Rolls, and I. E. T. de Araujo, "Self-organizing continuous attractor networks and path integration: one-dimensional models of head direction cells." *Network: Computation in Neural Syst.*, vol. 13, no. 2, pp. 217–42, May 2002.

[30] S. M. Stringer, E. T. Rolls, T. P. Trappenberg, and I. E. T. de Araujo, "Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells." *Network: Computation in Neural Syst.*, vol. 13, no. 4, pp. 429–46, Nov. 2002.

[31] D. Ball, S. Heath, J. Wiles, G. Wyeth, P. Corke, and M. Milford, "OpenRatSLAM: an open source brain-based SLAM system," *Autonomous Robots*, pp. 1–28, Feb. 2013.

[32] M. Quigley, K. Conley, B. Gerkey, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[33] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

Fiber Optic Capillary Sensor with Smart Optode for Rapid Testing of the Quality of Diesel and Biodiesel Fuel

Michal Borecki, Piotr Doroz, Przemysław Prus, Paweł Pszczółkowski, Jan Szmidt
Warsaw University of Technology
Institute of Microelectronics and Optoelectronics
Warsaw, Poland
E-mail: borecki@imio.pw.edu.pl

Jarosław Frydrych
Automotive Industry Institute
Warsaw, Poland.
e-mail: j.frydrych@pimot.eu

Michael L. Korwin-Pawlowski
Département d'informatique et d'ingénierie
Université du Québec en Outaouais
Gatineau, Québec, Canada
e-mail: michael.korwin-pawlowski@uqo.ca

Andrzej Kociubiński, Mariusz Duk
Lublin University of Technology
Department of Electronics
Lublin, Poland.
e-mail: akociub@semiconductor.pl

***Abstract*—There are many fuel quality standards introduced by national organizations and fuel producers. Usual techniques for measuring the quality of fuel, as for example cetane index, fraction composition and flash point, require relatively complex and expensive laboratory equipment. Therefore, testing of fuel is not rapid and can be costly. On the fuel user side, fast and low cost sensing of useful state of biodiesel fuel is important. One of the devices that address this task is the fiber optic capillary sensor in which forced local conversion of diesel fuel into vapor is implemented. The present paper concentrates on the critical elements the construction of the sensor as well as on the interpretation of experimental results. We have investigated the construction of the micro heater and the technology of smart capillary optrode preparation. We propose a capillary optrode construction and technology that reduces unwanted light coupling, as well as a new micro heater construction that uses a silicon carbide heating element. Our experimental assumption is that diesel fuel quality can be correlated with the type and concentration of its bio-components. We examined fuels that are mixtures prepared from components that are in line with European Union standards. The components used are petrodiesel fuel and bio-esters as well as edible rapeseed oil. For the mentioned fuels, we showed that the results of experiments are easy to interpret and that the useful state of diesel and biodiesel fuels can be determined from the time of local heating that is required for vapor phase creation and the local time of vapor bubble formation.**

Keywords-biodiesel fuel; fuel quality; useful state of fuel; fiber optic capillaries; fiber optic sensors; capillary sensor; smart capillary optrode

I. INTRODUCTION

This paper's focus is on selected aspects of construction of sensor that uses capillary optrode and enables rapid testing of quality of diesel and biodiesel fuels, the principle and preliminary results of which were presented in [1].

A. Diesel and biodiesel fuel production examination and usage

Rudolf Diesel constructed the first internal combustion engine using pure plant oil as carburant in 1912. The fuel was 100% peanuts oil; therefore, today this fuel classification is PPO – pure plant oil. The next time engines operated on PPO were built in 1985 [2].

Classical fuels are made from distillated products of crude oil. After distillation the oil composition depends on the process and on the crude oil parameters. The parameters of oil distillation process are not merely the boiling temperature but include such distillation characteristics as: initial distillation temperature (about 170ºC), end distillation temperature (about 370ºC) and the fractional contents temperatures. Those characteristics determine also the basic fuel parameters.

On the practical side, diesel engine performance, fuel consumption, and emitted pollutants result from the combustion process. The environment of combustion, the injected fuel's form and the fuel quality all play a primary role in the diesel combustion process. One of the most important diesel fuel quality parameters is ignition quality. The ignition quality depends on the molecular composition of the fuel. The ignition quality in turn is linked with ignition delay time, which is the time between the start of injection and the start of combustion. Measurements of ignition quality of fuel (CN) have to be carried out in the Cooperative Fuel Research (CFR -5) engine, under carefully controlled test conditions. The smaller is the delay of testing, the CN value is greater. The CN scale is based on the characteristics of know chemical single components liquid hydrocarbons. Therefore, the CFR-5 engine is also called cetane engine. The basic disadvantage of such approximation to fuel quality measurement is the high cost of the measurement device and the complexity of the procedure. The alternative approximation is the use of ignition quality tester (IQT™)

and the ASTM D6890/EN 15195(IP 498) test methods. The mentioned devices can be seen in [3, 4].

Nowadays, producers define the useful state of diesel fuel by several parameters: cetane number (min 51.0), density (860 to 890 kg/m^3), and distillation temperatures (for example T_{90} maximal value is 360°C), kinematic viscosity at 40°C (3.5 to 5.0 mm^2/s), etc. Other diesel fuel parameters characterize its operability: carbon residue, water and sediment, cloud point, conductivity at 20°C, oxidation stability, acidity, copper corrosion, flashpoint, lubricity, appearance, and color [5]. For the ordinary fuel user such collection of parameters is often too complex for practical use because their testing requires special laboratory equipment. Therefore, fuel examination is not rapid and can be costly.

The introduction of biodiesel fuel increases the number of parameters connected with the bio-component content [6]. In this situation the user requires the simplest possible answer to a question: Is that fuel useful for my engine?

B. *Economical reason of sensor of diesel and biodiesel fuel examination*

Sensing of useful state of biodiesel fuel is exceptionally important for car fleet owners and farmers.

Car fleet owners are interested because of legal regulations and of the risks of buying poor quality fuel [7]. In certain countries, including Poland, units of public administration, production and customers the law prescribes public auction for transactions of larger quantities of fuel. In such auction the ordered fuel is certified on the day of delivery as meeting national standards, like the Polish norm PN-EN 590. The purchased quantity of the fuel is often split between different tanks belonging to the buyers. During the fuel transfer into the buyers' tanks the buyers' representative may be present. Even though the ordered fuels may have to a certificate of quality, the buyers often reserve the rights to check the quality of the fuel. Quite often the delivered fuel quality tests are at the option of the buyers, and are often waived because of the associated costs, with sometimes conflicts arising when the poor quality is discovered at a later stage.

Farmers are often very interested in examination of diesel fuel quality because they can produce bio-fuel components for their own use. The characteristics of those home-produced components are not optima [8]. One can see clear differences between the freshly pressed technical rapeseed oil and the edible rapeseed oil, which is chemically clarified and stabilized with antioxidants, such as vitamin E. For practical fuel application the technical rapeseed oils have too small cetane numbers and too high viscosities. It seems that one of the reasons of low biodiesel fuel mixtures usage by farmers is the absence of a low cost device to evaluate its useful state [9]. In certain regions, there are in use mixtures of rapeseed oil with diesel fuel, sometimes they are modified with n-butanol or ethanol [10]. However, the viscosities of oils decrease with the increase of temperature. For these reasons, in tropical countries the potential of using biodiesel fuels is larger. The uses of mixtures of soapnut oil with petrodiesel fuel are discussed in [11]. It turns out that, despite significant differences in fuel viscosity and flash point, the observed engine parameters with the prepared mixtures were very similar [12].

In a European study, it was observed that using the biodiesel fuel of the first generation at low environment temperatures can lead to the degeneration of engine parameters [13]. Therefore, production standards for biodiesel fuel were introduced: density at 15°C (ISO3675) and temperature of fluidity for the transitional periods of season and winter (DIN EN 116). The disadvantages of biodiesel fuel can be overcome by fuel processing [14] or by using biopetrodiesel fuel mixtures [15]. There are also publications describing low energy processes for upgrading the technical parameters of biofuels. For example: transestrification of soybean oil with the use microwave and nanopowders enables creation of biodiesel fuel, which meets the requirements of the EN-14214 norm [16]. A new generation of 100% biodiesel fuel can be made with isomerization [17, 18]. Neste Oil proposes NExBTL bio component that can be used as 100% biodiesel fuel that means the diesel fuel without petrodiesel components, and is known as NesteGreen 100. The hydrogenated vegetable oils (HVO) are also used as biodiesel components. Their advantage is agreement with diesel particulate filter (DPF) of engines.

C. *Critical points of diesel fuel conversion into energy*

The starting point of this development of a new sensing method of the useful state of diesel and biodiesel fuels was to consider the two critical points of fuel conversion into energy. The first is the injector of atomized fuel into the combustion chamber by forcibly pumping it through a small nozzle. The second critical point is the exhaust of gases filtered with the diesel particulate filter. Periodically, the DPF has to be taken up to high temperatures to burn off the matter it has collected, which is realized by contact of DPF with a part of fuel vapor [19].

Typically, fuel is injected into the cylinders just after the vapor fires and the exhaust valve opens. At injection point, the fuel vaporizes and a part of vapor moves down the exhaust to the DPF and cleans it in a precisely controlled injection scheme [20]. Because biodiesel fuel has a higher distillation temperature than petrodiesel fuel, it does not vaporize as fast [21, 22]. Some of the biodiesel fuel can end up adhering to the injector, the cylinder wall or runs past the rings, diluting the engine oil and diluting DPF deposits instead of cleaning it.

Therefore, the examination of vapor creation parameters of biodiesel fuels is critical to evaluate its useful state regardless of the composition of fuel. The methods of spray forming observation in diesel engine have been used [23], but are not good for integration into a sensor device.

In this work, we present new developments and new applications of on capillary photonic sensors working on the principle of monitoring optical intensity changes in dynamically forced measurement cycles, first postulated in [24]. We present the idea of the sensor, the construction of the head, the experimental results of testing biodiesel fuels for their quality for use, and conclusions.

The sensors use fiber optic capillaries in which the phase of the filling liquid changes locally to gas when forced by local heating, while the propagation of light in the capillary is monitored. Therefore, the sensors examine simultaneously many parameters of the liquids. To evaluate the performance of the sensors, we used oils with known quality.

II. IDEA OF SENSOR HEAD

The sensor head idea is inspired by one of most critical diesel engine element that is fuel injector and nozzle. The actual dimensions of the typical nozzle showed in Fig. 1 are ID/Id = 4, L/Id = 5 and Id = 0.2mm. The fuel injector nozzle diameters depending on construction can vary from 50 to 200µm [25].

The influence of relation between nozzle dimensions, pressure ratio on the type of diesel fuel flown is under investigations [26]. The character of the fuel flow through the nozzle depends strongly on the pressure difference. It can be classified as cavitating or non-cavitating type, while in the non-cavitating flow case we distinguish between the turbulent and the laminar flow. The caviting flows happen when the difference in pressures is high enough. The input pressure (P_i) can vary from 15MPa to 110MPa. The output pressure (P_n) is of the order of 6MPa.

Typical temperatures inside the fuel injection nozzle are from 235 to 275°C, the maximum does not exceed 300°C [25]. The flame temperatures in the cylinders (T_n) are about 1500°C and the wall temperatures are under 350°C. Therefore, we cannot replicate the flame temperatures and pressures in small portable sensor devices.

On the other hand, the volume of a single injection of the fuel is at the range of 3÷50mm^3. The fuel injected into cylinder forms a spray and enters the cylinder in about few milliseconds [28]. Then it vaporizes and flames.

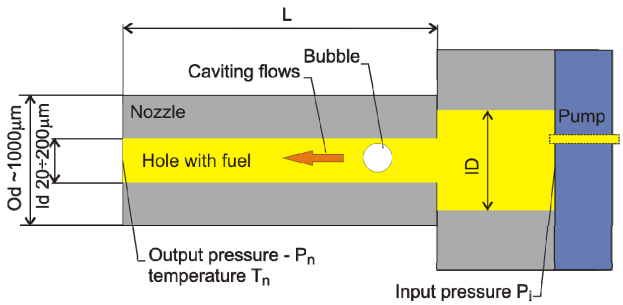


Figure 1. Schematic construction of the nozzle.

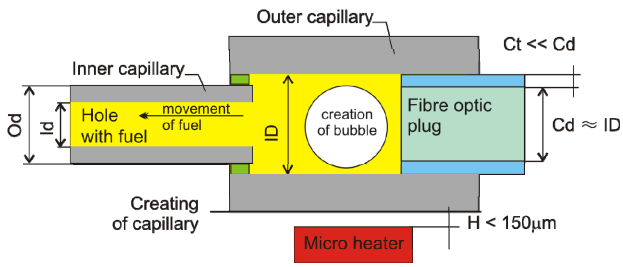


Figure 2. Schematic construction of sensor head.

These processes are correlated with the quality of ignition, which forms one of the major factors of the fuel quality. The second factor when considering biodiesel fuel is connected with fuel viscosity. Therefore, we intend to examine in our sensor the fuel vaporization and forced flow in conditions that are as close to reality as possible. We have to create a setup allowing the examination of partial evaporation of fuel, which take place in the nozzle and can move the fuel into the orifice of few hundreds micrometers diameter. Such a nozzle can be modeled with two glass capillaries that would allow observation of the direct optical fuel phases and their movement. The capillary with the smaller outer diameter can be positioned inside the bigger capillary using glue forming a single-use replaceable optrode [1].

The inner temperature that is needed to create the bubble of vapor can be achieved with a local heater positioned near the capillary. With one end of capillary closed, the local heater can act as a fuel pump by producing a vapor pressure (see Fig. 2). The set of commercially available capillaries produced by VitroCom enables to prepare the optrode that is characterized by dimensions close to those of practical nozzles. For example; as the inner capillary we can use CV3040Q capillary that Id = 300µm as the outer capillary we can use CV7087Q capillary, which ID = 700µm. Both capillaries can work in temperatures up to the quartz glass annealing point, which is 1070ºC. The outer capillary (CV7087Q) length should be greater than 7.8mm to move 3mm^2 of fuel.

The creation and movement of the bubble in the liquid depends on the liquid thermo-dynamical parameters as well as its viscosity and its vapor phase parameters, and also on the container's geometry and the outer thermo-dynamical conditions [29]. The faster is the bubble creation from liquid phase, the more probable is the turbulent flow of fuel in the nozzle. Therefore, we have to distinguish two stages of the bubble creation: the time of liquid fuel heating and the time of phase change from liquid to vapor that forms the bubble filling the full cross section of the capillary.

III. HEAD CONSTRUCTION

The sensor's head consists of two functional blocks: the base and the optrode [30]. The base is used to integrate the micro heater, the optical path of source and receiver as well as for positioning the optrode. The optrode is the replaceable part of the head that imitates the geometry and the main physical characteristics of a fuel nozzle and enables monitoring of creation of the vapor bubble.

A. Micro heater

The micro heater has to supply sufficient heat for the biodiesel fuel to reach over 200°C inside the capillary. We examined experimentally and by numerical modeling the map of temperatures in the model of nozzle. We used the Coventor software, a R300 NEC thermo-vision camera, and the InfReC analyzer software. The results of micro heater simulation are presented in Fig. 3.

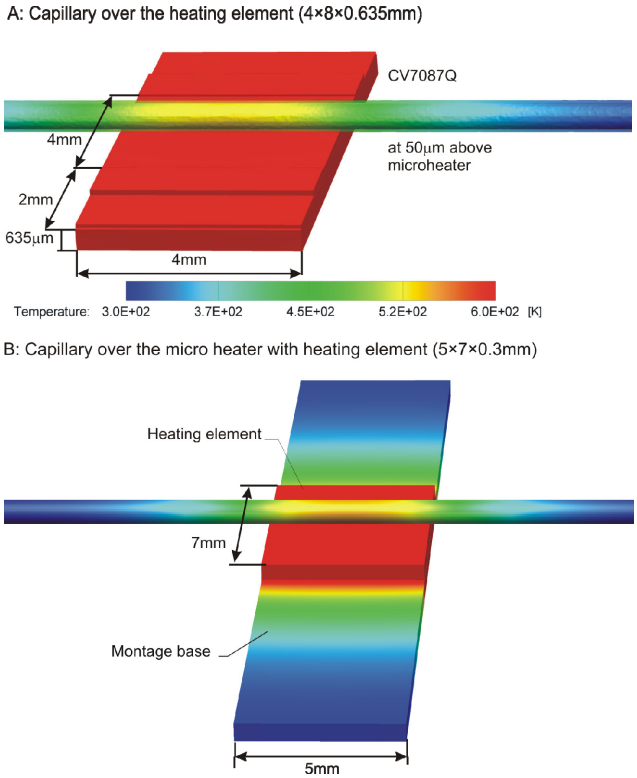


Figure 3. Temperature map in [°K] at 30s of heating for a glass capillary CV7087Q filled with diesel fuel.

In both cases, the capillary optrode and its position over heating element is the same. The situation in Fig. 3a concerns thick film type planar heating element with total dimension 4mm×8mm×0.635mm. The element is equipped with a resistance heating element and with connection pads. The resistance element area covers 4×4mm and is positioned at 50μm under the capillary. For dissipating 5W in 30 seconds the temperature of the surface of the heater reached 327°C while the temperature inside the capillary reached 247°C. The full microheater assembly (Fig. 3b) including the mounting base requires dissipating of 7W in 30 seconds to achieve the same temperature inside the capillary.

Sequential simulations showed that for the 290ºC temperature inside the filled up capillary and for the assumed distance between the capillary and micro heater surface, the micro heater with dimensions of 5mm×5mm, the upper surface temperature has to be at least 350°C. This temperature is more than can withstand the planar resistors, e.g., Vishay High Power Thin Film Wraparound Chip Resistor in type 2512 package [31]. Wire heaters can easily work at such temperatures, but since such constructions do not provide a constant and repeatable distance between the microheater and the capillary, they cannot replace the planar structures. The most favorable shape of planar microheaters is rectangular with side length from 2mm to 5mm, and the recommended power of heating is 7÷10W. The power density can reach 1.6W/mm^2, which is also too high for commercial hybrid resistors. For the heater current supply the recommended value of resistance is between 10÷50Ω. Even current dividers from Vishay like the Current Sensing Bondable Chip Resistors type S.C. are not optimal for such application. More over, the head construction requires that the microheater has to be positioned on a rectangle shaped substrate with the length of 3cm and width of 5mm. On this substrate the electrical contacts for the heating element and wire connections have to be provided, as well as isolating pads for mounting the head. The isolating regions are necessary because the head base is made of metal. The micro heater base is outlined in Fig. 4.

We have built different versions of planar micro heaters. In all construction we used alundum ceramic bases with the thickness of 635μm. For preparing the heating elements we used thick film and thin film hybrid technologies as well as monolithic silicon carbide semiconductor technologies.

The thick film technology enabled us to make a fully integrated micro heater in the form of one piece. In our investigation this element worked repeatable in 20 seconds cycle without long term resistance changes when end temperatures did not exceed 200°C. The parameters of the micro heater were stable for temperature shocks from 30°C to 200°C – the reversible resistance changes were low, within 1.5Ω at 30Ω of nominal resistance. When the temperature exceeded 200ºC we observed cracking of the resistive layer followed by the splitting of ceramic base into two parts. Therefore, we examined the two other technologies.

We used thin film metal deposition technology to prepare standalone resistance elements. The metal was deposited on a silicon substrate. There were three areas in the element: one square resistive heating area and two areas for electrical connections. The connections between the heating element and the microheater base were made by wire bonding or with high temperature conductive glue use. To bond it, we at first positioned the element on the base with dielectric glue. At that, we observed two unwanted phenomena, both due to the high temperature of the connected elements. For wire bonding technology we observed at 150°C that standard dielectric glue gave smoke, but we did not detect any rapid changes of the resistance of the micro heater. With conductive glue technology use at 300°C we observed that the high temperature connecting glue, which was specified as working up to 400°C also gave smoke while the connections resistance increased from 1 Ω to 10Ω. This increase had an unstable character, but the glue still properly positioned heating element on the base.

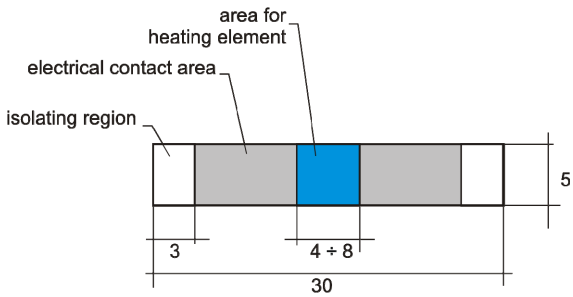


Figure 4. Micro heater base outline.

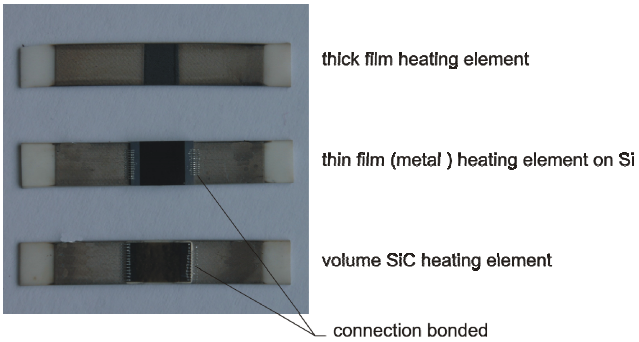


Figure 5. Tested micro heaters.

Moreover, the thin film heating element increased its resistance during heating cycles in such way that the initial resistance of following heating cycle was bigger than in previous one. We think that special passivation of thin metal layer will solve this problem.

At last we evaluated monolithic silicon carbide heating elements. The element enabled heating capillary to the required temperature in 30 second cycles. The increase of time of heating from 25 to 30 second with decreased power but maintained final temperature resulted in partial degradation of the glue used for bonding. We used high temperature conductive glue; therefore, we observed the mentioned degradation as small permanent changes in the microheater's resistance. Hence, we limited heating time to 25 seconds. The next heating cycle with maximum power was safe, when the heater was allowed to cool down to room temperature. In normal condition it required about 2 minutes for cooling. The tested micro heaters are shown in Fig. 5.

B. Path of optical signal

The creation of the bubble can be observed from outside or inside of capillary with the use of optical fibers [32]. The bubble position can vary in the area of local heating due to variation of fuel composition and real geometrical dimensions of capillaries within with their specified tolerances. Moreover, the outside observations of effects happening inside the capillary are sensitive to outer capillary cleanness [33, 34]. Therefore, the observation from outside is not optimal for measuring the bubble creation time. Observation of the bubble creation with two fibers positioned inside the capillary is not convenient for a replaceable optrode setup, and also complicates the fuel flow. To overcome those problems, we used a modified capillary optrode with a phosphor layer to convert radiation (see Fig. 6).

In the presented optrode, the phosphor converts the light from 460nm wavelength of the high power light emitting diodes, to 562nm. Only part of the light radiated in the full angle extent propagates in the inner capillary to the area of examination. The fragment of optrode where phosphor is deposited is presented in Fig. 7. When we illuminated the phosphor with the radiation of 460nm at the cross section of outer capillary of optrode we registered the presence of radiation at 460nm and 562nm wavelengths (see Fig. 8).

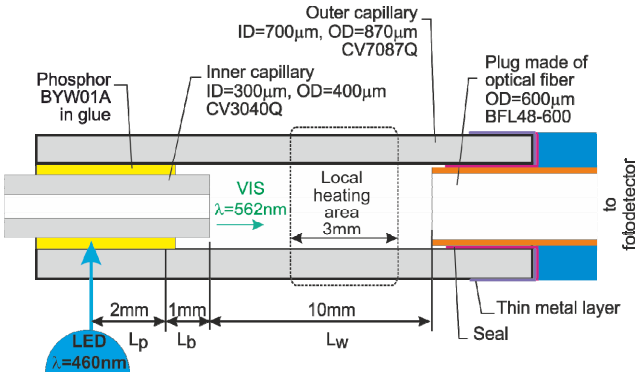


Figure 6. Optrode that uses phosphor to convert outer radiation into light inside capillary.

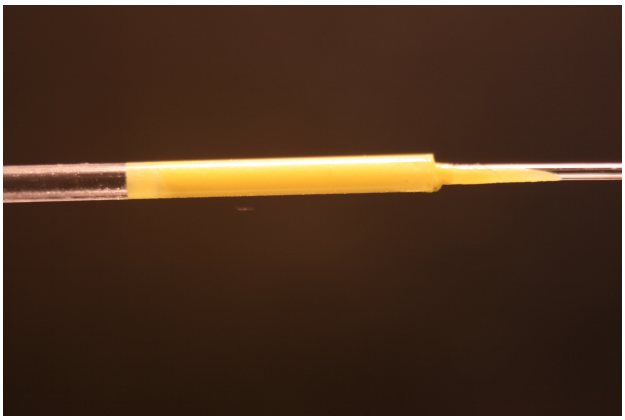


Figure 7. Optrode part where UV phosphor is deposited.

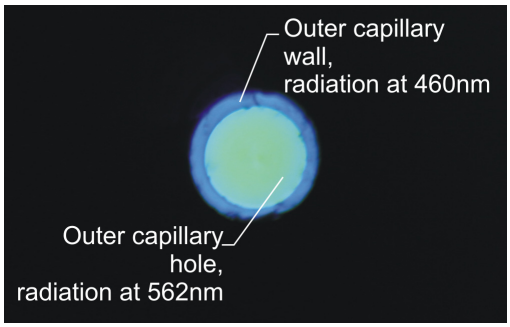


Figure 8. Cross section of outer oprtode when phosphor is radiated with radiation at 460nm wavelength, the output power is at the rank of 300nW.

To eliminate the coupling to the receiver of the unwanted radiation propagating into the wall of outer capillary, we used two means of protections. First, the receiver fiber plug was inserted into optrode. Second, the optrode walls at the plug side were covered with a thin metal layer (see Fig. 9) [35]. The optical quality of thin film protection is presented in Fig. 10, where the optical beam coupled into the outer capillary has 2mW of power at 675nm. No optical signal that would output from the capillary walls is in evidence, contrary to the situation presented in Fig. 8.

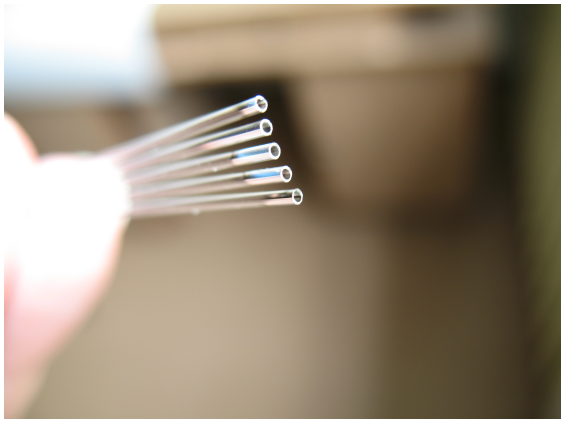


Figure 9. The end of optrodes with deposited thin film metal layers.

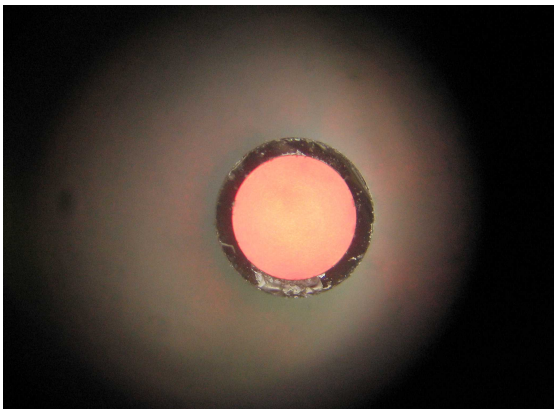


Figure 10. The end walls of outer capillary protected with metal layer, when 2mW optical beam was coupled into the capillary.

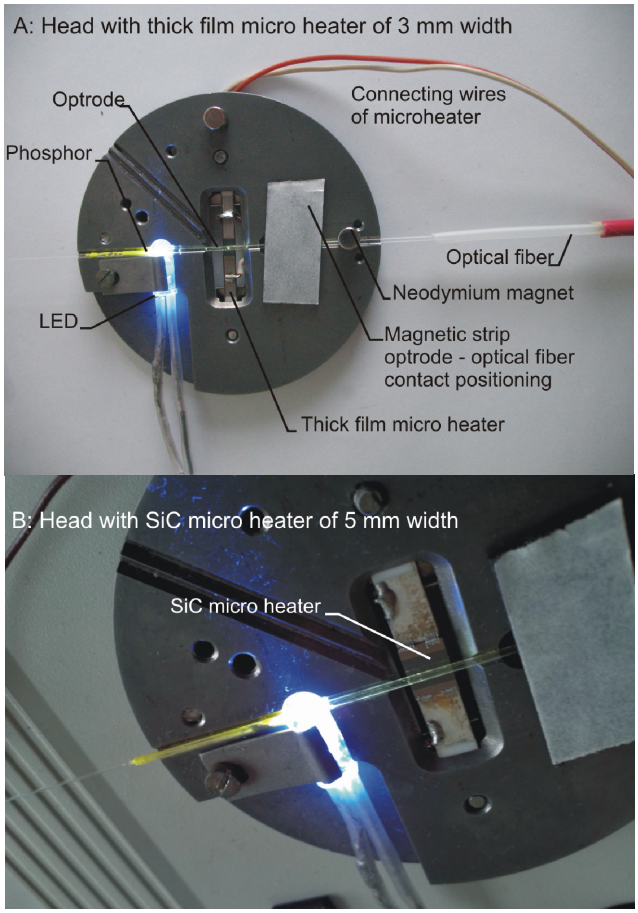


Figure 11. The head construction.

The efficiency of light conversion in the developed optrode is low, but acceptable. We optimized the optrode elements position: Lp, Lb and Lw (see Fig. 6), as well as the method and parameters of phosphor deposition. After optimizing the construction, we got from a L7113QBC-G LED operating at 20mW, at the end of the plug made from optical fiber, 111nW for an empty capillary and 0.3µW for a capillary filled with biodiesel fuel. The uncertainty of low signal level in our construction was 10nW. The optrode was held in position with elastic magnetic strips, while the optical fiber was secured with miniature neodymium magnets. The construction of the head is presented in Fig. 11. The two types of micro heaters were installed. In Fig. 11A, the holders of micro heater are better visible than in Fig. 11B. The electrical connections from micro heater to power supply are made at some distance from the heating element to secure their proper working temperature.

Though we predicted that thick film micro heater may be not proper for biodiesel fuel examinations, we checked such a possibility by making a series of fuel examination experiments. The results showed that a critical degradation of such micro heater happens after over a dozen measuring cycles. Therefore, for further experiments we used only micro heaters with SiC heating element.

C. *Optoelectronic signal processing*

As the light source driver we built an electronic device that enabled current modulation from DC to 50 kHz at selected frequencies, which was equipped with configurable current limiters to prevent accidental LED burning. To improve the rejection of ambient light influence on the experimental results in our experiment we used electrically modulated light with 1kHz frequency.

The optoelectronic detection unit of our own construction had an SMA fiber input and consisted of an integrated photo-amplifier and a band-pass filter with amplification and RMS detection. We used the S8745-01, AD8253, UAF42, AD536 and AD8250 components. In the realized construction we were able to measure signals in the range from 10nW up to 500nW with 2nW accuracy when the signal duration was 0.01s. The optoelectronic unit was connected to a personal computer through an analog input IOtech personal Daq 3000 16bit/1MHz USB data acquisition system. We fed the heater from a laboratory power supply Hameg HM8143 controlled by the analog output from Daq. The view of sensor hardware set-up is presented in Fig. 12. We also used a Daq 3000 system to monitor the temperatures of the measuring head base and of the surrounding ambient with two LM35DT circuits connected by low pass filters.

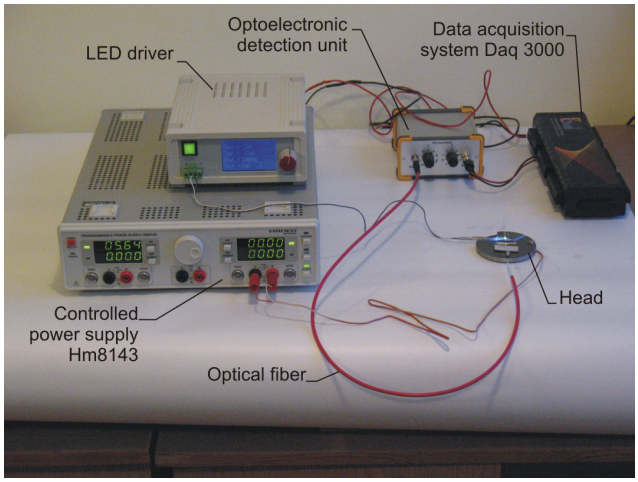


Figure 12. View of experimental set-up, [1].

To operate the system, we designed a script in DASYLab with a 0.01s sampling rate. The script automates the measurements and automatically switched off the micro heater when the light signal dropped under a specified value corresponding to the point of vapor bubble creation or when maximum time of local heating is exceed. The length of signal registration was 60s.

IV. EXPERIMENTAL RESULTS

In this section are presented the experimental procedure and the results of examination of different diesel and biodiesel fuels.

A. Experiment procedure

At the start of the experiment the optrode was closed with a fiber optic plug without use of sealant and then is placed in the head. The LED was switched on at the power of 20mW. The initial output signal was measured and when it was greater than 110nW we assumed that the optrode was qualified to use (see Fig. 13). Next, the optrode was filled with fuel, after which its end was closed with fiber optic plug secured with sealant. When there were bubbles of gas observed at the initial state of experiment, the optrode and capillary had to be withdrawn [29]. When the capillaries were filled uniformly by the liquid, the optrode was placed into the head and the initial levels of transmitted signal were examined and used as reference levels. In normal situation the signal should be greater than 300nW. We normalized such initial signal level to 4 a.u. (in Fig. 13).

As the examined fuel in the useful state was semitransparent, we expected initially high signal levels, and then low signal levels when the bubble would appear. The bubble directed the signal from the liquid to the capillary walls [32]. When the transmitted signal decreased rapidly it gave the impulse to switch off the microheater. We terminated the heating when the signal dropped under 2.5 a.u.

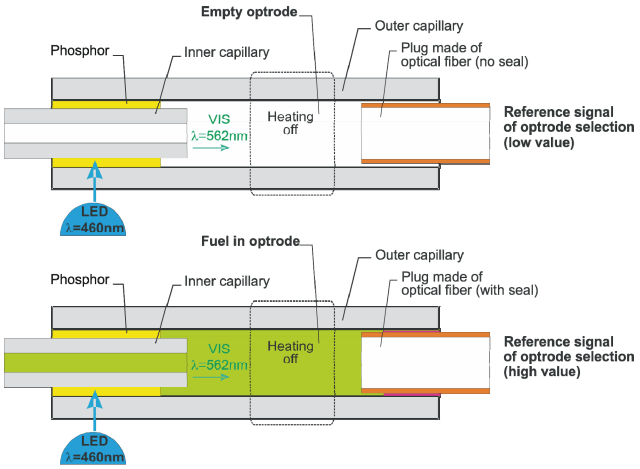


Figure 13. Initial situations of experimental procedure.

Depending on the thermo-dynamical conditions; the vapor gas phase moved the fuel to the open end with a laminar or a turbid flow. The turbid flow could be detected optically after the experiment as a presence of series of small bubbles in the optrode. Depending on the flow type and fuel decomposition after heating we could observe two situations presented in Fig. 14.

Both situations presented in Fig. 14 result in the reduction of the output optical signal and are difficult to be distinguished in the measurement, but are easy to be distinguished visually. The small bubble appears at the stable position over center of micro heater, where the temperature achieves its maximum. We thought that a small bubble appeared and lasted in the fuel due to structural changes induced by heating to some components that were added after distillation, or to some decomposition of bio-components, since the bubble was particularly present after examination of edible oils [36].

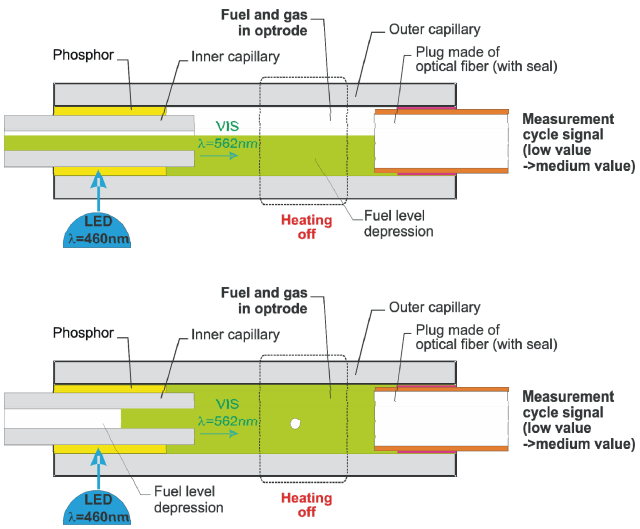


Figure 14. Two possible situations after local heating procedure.

B. Diesel and biodiesel fuels used for examination

We examined six potential fuels in which set five fuels were prepared from the same pure and fresh components at different ratios. For these fuels we use petrodiesel, fatty acids methyl esters (FAME) and additives according to EU standard. As sixth potential fuel we suppose edible rapeseed oil (ERO). The edible rapeseed oil has to be distinguished from technical rapeseed oil (TRO). Technical rapeseed oil is the direct product of rapeseed corn pressing, while edible oil is the product of technical oil purification and addition of antioxidants.

We defined the fuel quality as: premium, good, acceptable for selected engines and acceptable for special engines. As the premium quality fuel we evaluated the clear and fresh petrodiesel component that was mixed with selected additives according to EU standards. The good quality fuel was premium quality fuel mixture with 10 of FAME. The acceptable for selected (tolerant) engines fuels consisted of 30 and 60% of FAME. We assumed that the FAME with additives fuel known as 100% biodiesel fuel could be used in special engines.

Selected parameters of prepared fuels are grouped in Table I. The examination of the fuels in the laboratory prior to the experiment showed that fuels P100, B10, B30 and B60 were meeting the norms. The FAME fuel starts to distillate at an unacceptable high temperature. B100 - modified FAME may start distillation at T0 = 280ºC while T10≈T50≈T90≈340ºC, therefore, its usage in classic diesel engines is questionable [13]. The used ERO is characterized by a very quick start of distillation at 170-200ºC, then no observable change to 295ºC, where it starts distillate as fuel up to 363ºC. The B100, TRO and ERO did not meet the distillation standards of fuels.

TABLE I. SELECTED PARAMETERS OF PREPARED FUELS

Parameter	Fuel acronym				
	P100	B10	B30	B60	B100
Assumed fuel quality	P	G	Ab	Ab	Bb
Base oil [%]	100	90	70	40	0
FAME [%]	0	10	30	60	100
Density at 15ºC [kg/m^3]	832.6	837.4	847.0	862.3	883.2
Temp of flame [°C]	74	75.5	79.5	90	163
Kinematic viscosity at 40ºC [mm^2/s]	3.367	3.432	3.595	3.934	4.509
CI	54.9	57.7	57.5	56.8	*
CN	59.6	57.3	54.9	54.0	51.2
T_0 [ºC]	188.6	195.6	196.7	200.2	369*#
T_{10} [ºC]	225.7	230.4	242.1	278.1	*
T_{90} [ºC]	345.5	343.6	344.1	345.3	*

Abbreviations used: FAME – Fatty acids methyl esters (bio-component); CI – cetane index, CN – cetane number, T0 temperature of distillation start, T_X – temperature of x% volume of distillation, * - our lab equipment do to allow of such examination, value given in [14]. Assumed fuel quality set: P – premium, G – good fuel, Ab – acceptable bio fuel for selected engines, Bb – acceptable biofuel for special engines.

C. Examination of biodiesel fuels using the developed sensor

Our aim was to distinguish fuels by their quality. For this purpose we set in the experiment the power of micro heater and the maximum time of local heating. The 5W of power in thick film heating element and corresponding 7W of power dissipated in SiC micro heater in up to 25 seconds enabled to produce vapor phase for fuels that meets the norm as well as ERO.

We examined at least 3 times samples of each fuel, and on the following figures we present representative measurement cycle signal. As micro heater we used the SiC component.

We made the first experiments with P100 fuel at two powers of heating 5W and 7W (Fig. 15 and Fig. 16). The experiments showed that increasing the power from 5W to 7W reduced the average time of heating τ from 14.5 seconds to 9 seconds, and it reduced the average time of vapor phase creation Δτ from 0.2 seconds for 5W, to 0.1 seconds for 7W. The reductions in those times were in agreement with the theory of thermo dynamics.

The next experiments were made with 7W heating power and their results are presented in Fig. 17 to Fig. 20.

From our experimental results we saw that P100, B10 and B30 fuels did not differ significantly. The B60 fuel formed in our heating condition a vapor phase, but the mixture was characterized by very high dispersion of time of heating - τ.

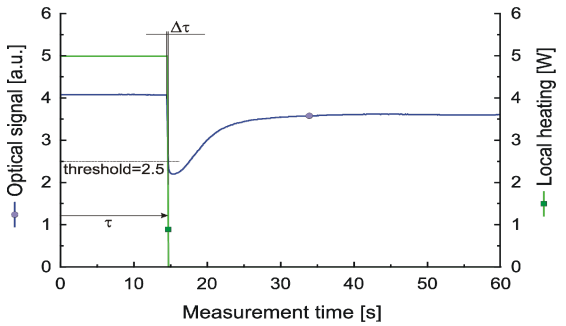


Figure 15. Measurement procedure representative signal of P100 heated with 5W.

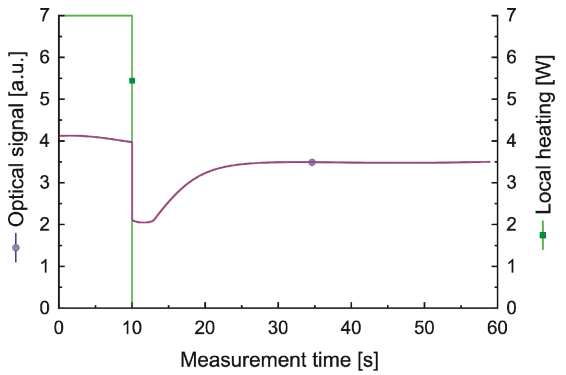


Figure 16. Measurement procedure signals of P100 heated with 7W.

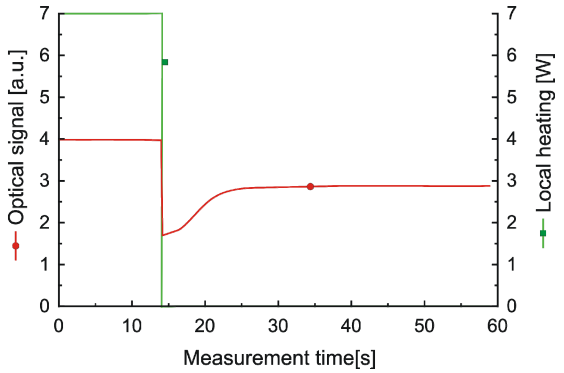


Figure 17. Measurement procedure signals of B10 heated with 7W.

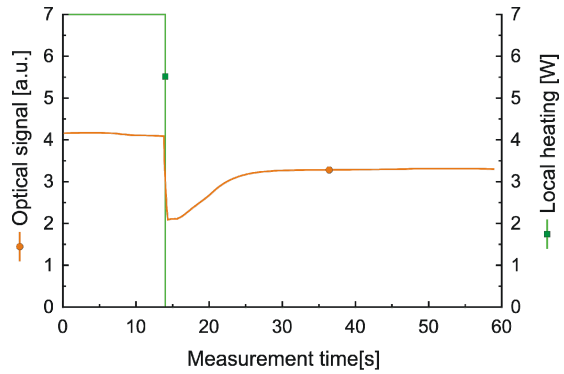


Figure 18. Measurement procedure signals of B30 heated with 7W.

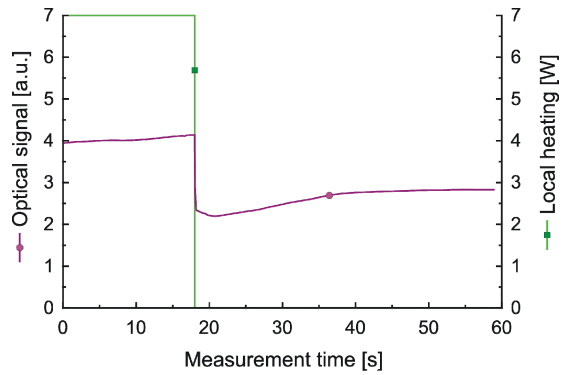


Figure 19. Measurement procedure signals of B60 heated with 7W.

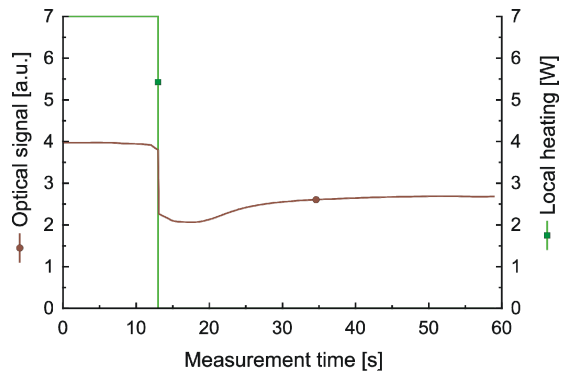


Figure 20. Measurement procedure signals of ERO heated with 7W.

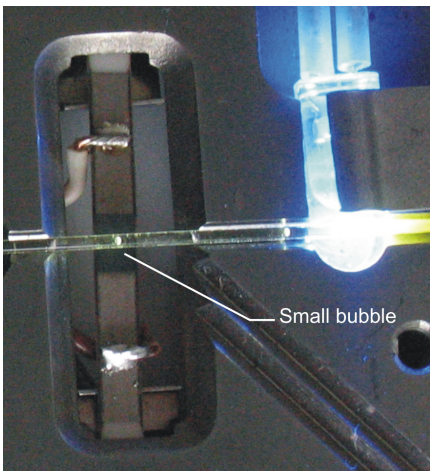


Figure 21. Small bubble remaining after heating of rapeseed edible oil in the center of the micro heater.

B100 required a longer time of heating to form bubble phase than our procedure enables. Only 30% of samples showed bubble creation in 25 seconds of local heating. This was made intentionally to enable proper classification of diesel fuel with acceptable parameters while time of heating is considered as a one of key factors. Sometimes B100 showed a lower time of bubble creation than ERO. Interestingly, according to our experimental data the B100 seemed to be a worse fuel than ERO. The ERO forms vapor phase in all cycles was similar to that of B10 fuel. But ERO had the lowest τ dispersion, which was in agreement with its distillation starting point at 170-200ºC. This is not a good property for a fuel. For ERO we observed a repeatable presence of a small bubble that remained after heating of edible rapeseed oil in the center of the micro heater (see Fig. 21).

We think that was the result of thermal decomposition of rapeseed edible oil components. It was most probably connected to the degradation of vitamin E, which occurs at 200ºC. Therefore, in ERO we examined only a part of distillation parameters, but this disadvantage could be corrected by observation of optrode after sample examination.

The summarized results of experiment of fuel examination are presented in Table II. What was expected but not obvious, our head was not sensitive to local heater construction as long as the temperature conditions of heating were the same. We see an agreement of the results collected with hybrid micro heater presented in [1] and those presented in this paper measured using the SiC micro heater.

TABLE II. EXAMINED PARAMETERS OF FUELS HEATED WITH 6W

Parameter	Fuel acronym					
	P100	B10	B30	B60	B100	ERO
Average τ [s]	9	12	13	22	*	13
Average Δτ [s]	0.10	0.13	0.18	0.30	*	0.64
Percent of samples with created bubble	100	100	100	100	30	100

*- the average value does not exist.

On the base of data collected in the experiments we can set the parameters determining the useful state of biodiesel fuel as: the upper limits of average time of heating – avg(τ), the range of dispersion of time of heating – std(τ) and the upper limit of time of vapor phase creation – max(Δτ). The analysis of data showed that in our method the useful state of diesel biodiesel fuel was directly and firmly connected with the gas phase creation. For example, the unacceptable fuels are characterized by Δτ greater than 0.3s while premium fuels Δτ have to be lower than 0.13s.

CONCLUSIONS

We proposed a sensor working on the principle optical examination of fuel under local heating. Our optoelectronic devices allowed conducting the experiment in ambient room conditions. The analysis of the measured signals of diesel and biodiesel fuels showed the relationship of times of gas phase creation parameters with the useful state of diesel fuel. We showed that the information on useful state of diesel fuel could be presented in the form of recommended ranges and times of fuel heating and vapor creation. Because the heating was taking place in a closed capillary, the fuel did not ignite during experiments. We conclude that the proposed construction may be in future the base of commercially marketable instruments.

The future work will consist of optimization of the construction especially enabling observation of optrode above the center of micro heater. The sensor construction needs to be integrated into a complete portable instrument and be built more resistant for use in harsh environments outside of the laboratory. For this purpose, the first step was achieved, as thanks to indirect light coupling the optrode walls are not much sensitive to soiling. The next required step is the development of new simpler in usage optrode plug.

ACKNOWLEDGMENT

This work was supported by: the European Union structural funds grant InTechFun task 5, “Multi parametric classificatory of liquid biofuels useful state”.

REFERENCES

[1] M. Borecki, P. Doroz, J. Szmidt, M.L. Korwin-Pawlowski, A. Kociubiński, and M. Duk, “Sensing method and fiber optic capillary sensor for testing the quality of biodiesel fuel,” IARIA, Proc. Sensordevices 2013, pp. 19–24.

[2] G. Knothe, “Historical perspectives on vegetable oil-based diesel fuels,” Industrial Oils, vol. 12, 2001.

[3] http://www.compass-instruments.com/waukesha.shtml, accessed 05.12.2013

[4] http://www.aet.ca/index.php?section=20, accessed 05.12.2013

[5] Department of Industry, Science and Resources, “Setting national fuel quality standards - Discussion paper 4,” in “Operability fuel parameters (petrol and diesel),” Environment Australia, 2001.

[6] D. Mueller, M.F. Ferrão, L. Marder, A.B. da Costa, and R. de Cássia de Souza Schneider, “Fourier transform infrared spectroscopy (FTIR) and multivariate analysis for identification of different vegetable oils used in biodiesel production,” Sensors, vol. 13, 2013, pp. 4258–4271.

[7] G. Mothé, M. Castro, M. Sthel, G. Lima, L. Brasil, L. Campos, A. Rocha, and H. Vargas, “Detection of greenhouse gas precursors from diesel engines using electrochemical and photoacoustic sensors,” Sensors, vol. 10, 2010, pp. 9726–9741.

[8] N. Li, Q. Zhou, X. Li, W. Chu, J. Adkins, and J. Zheng, “Electrochemical detection of free glycerol in biodiesel using electrodes with single gold particles in highly ordered SiO_2 cavities,” Sensors and Actuators B: Chemical, vol. 196, 2014, pp. 314–320.

[9] P.L. Faccendini, M.É. Ribone, and C.M. Lagier, “Selective application of two rapid, low-cost electrochemical methods to quantify glycerol according to the sample nature,” Sensors and Actuators B: Chemical, vol. 193, 2014, pp. 142–148.

[10] I. Barabas, A. Todorut, and D. Baldean, “Performance and emission characteristics of an CI engine fueled with diesel-biodiesel-bioethanol blends,” Fuel, vol. 89, 2010, pp. 3827–3832.

[11] R.D. Misra and M.S. Murthy, “Performance, emission and combustion evaluation of soapnut oil–diesel blends in a compression ignition engine,” Fuel, vol. 90, 2011, pp. 2514–2518.

[12] C. Ketlogetswe and J. Gandure, “Blending cooking oil biodiesel with petroleum diesel: a comparative performance test on a variable ic engine,” Smart Grid and Renewable Energy, vol. 2, 2011, pp. 165–168.

[13] M. Balat and H. Balat, “A critical review of biodiesel as a vehicular fuel,” Energy Convers Manag., vol. 49, 2008, pp. 2727–2741.

[14] S. Gryglewicz, “Rapeseed oil methyl esters preparation using heterogeneous catalysts,” Bioresource Technology, vol. 70, 1999, pp. 249–253.

[15] A.R. Sadrolhosseini et al., “Physical properties of normal grade biodiesel and winter grade biodiesel,” Int. J. Mol. Sci., vol. 11, 2011, pp. 2100–2111.

[16] W. Gis, A. Zoltowski, and A. Bochenska, “Properties of the rapeseed oil methyl esters and comparing them with the diesel oil properties,” J. of KONES Powertrain and Transport, vol. 18, 2011, pp. 121–127.

[17] N.E. Leadbetter et al., “Fast, easy preparation of biodiesel using microwave heating,” Energy & Fuels, vol. 20, 2006, pp. 2281–2283.

[18] M.C. Hsiao, C.C. Lin, and Y.H. Chang, “Microwave irradiation-assisted transesterification of soybean oil to bio-diesel catalyzed by nanopowder calcium oxide,” Fuel, vol. 90, 2011, pp. 1963–1967.

[19] M.V. Twigg and P.R. Phillips, “Cleaning the air we breathe – controlling diesel particulate emissions from passenger cars,” Platinum Metals Rev., vol. 53, 2009, pp. 27–34.

[20] B. Kegl, “Numerical analysis of injection characteristics using biodiesel fuel,” Fuel, vol. 85, 2006, pp. 2377–2387.

[21] M. Gumus, C. Sayin, and M. Canakci, “The impact of fuel injection pressure on the exhaust emissions of a direct injection diesel engine fueled with biodiesel–diesel fuel blends,” Fuel, vol. 95, 2012, pp. 486–494.

[22] W. Yuan, A.C. Hansen, and Q. Zhang, “Vapor pressure and normal boiling point predictions for pure methyl esters and biodiesel fuels,” Fuel, vol 84, 2005, pp. 943–950.

[23] B. Kegl, M. Kegl, and S. Pehan, “Optimization of a fuel injection system for diesel and biodiesel usage,” Energy & Fuels, vol. 22, 2008, pp. 1046–1054.

[24] M. Borecki, M. L. Korwin-Pawlowski, and M. Bebłowska, “A method of examination of liquids by neural network analysis of reflectometric and transmission time domain data

from optical capillaries and fibers," IEEE J. Sensors, vol. 8, 2008, pp. 1208–1213.

[25] D.L. Siebers and L.M. Pickett, "Injection pressure and oritice diameter effects on soot in DI diesel fuel jets," in Thermo- and fluid dynamic process in diesel engines 2 - Selected papers from Diesel 2002 conference Valencia, Spain, J.H. Whitelaw, F. Payri, C. Arcoumanis, and J-M. Desantes, Eds., Springer-Verlag, Berlin, Heidelberg, New York, 2002, pp. 109–131.

[26] X. Wang and WH. Su, "A numerical study of cavitating flows in high-pressure diesel injection nozzle holes using a two-fluid model," Chinese Sci Bull, vol. 54, 2009, pp. 1655–1662.

[27] N. Ladommatos, Z. Xiao, and H. Zhao, "The effect of piston bowl temperature on diesel exhaust emissions," Proc. IMechE. vol. 219 Part D: J. Automobile Engineering, 2005, pp. 371–388.

[28] C. Crua, J.C. Evans, D.A. Kennaird, and M.R. Heikal, "In-cylinder study of the formation, autoignition and soot production of diesel sprays at elevated pressures," 9th International Conference on Liquid Atomization and Spray Systems (ICLASS) Sorrento, Italy, 13-17 July 2003, http://eprints.brighton.ac.uk/2184/ access 07.02.2014

[29] M. Borecki, M. Korwin Pawlowski, P. Wrzosek, and J. Szmidt, "Capillaries as the components of photonic sensor micro-systems," J. of Mater. Sci. and Technol., vol. 19, 2008, pp. 065202.

[30] M. Borecki, M.L. Korwin-Pawlowski, M. Beblowska, J. Szmidt, and A. Jakubowski, "Optoelectronic capillary sensors in microfluidic and point-of-care instrumentation," Sensors, vol. 10, 2010, pp. 3771–3797.

[31] www.vishay.com, Document Number: 53013, Revision: 25-Aug-09 39, pp. 39-39 [retrieved June 1, 2013].

[32] M. Borecki and M.L. Korwin-Pawlowski, "Optical capillary sensors for intelligent microfluidic sample classification," in "Nanosensors: Theory and Applications in Industry, Healthcare and Defence," T.C. Lim, Ed., CRC Press, Boca Raton, FL, USA, 2011, pp. 215–245.

[33] P. Rugeland, C. Sterner, and W. Margulis, "Visible light guidance in silica capillaries by antiresonant reflection," Optics Express, vol. 21, 2013, pp. 29217–29222.

[34] R.S. Romaniuk, "Geometry design in rerfractive capillary optical fibers," Photonics Letters of Poland, vol. 2, 2010, pp. 64–66.

[35] J. Gryglewicz, P. Firek, J. Jasiński, R. Mroczyński, and Jan Szmidt, "Characterization of thin Gd2O3 magnetron sputtered layers," Proc. of SPIE, vol. 8902, 2013, pp. 89022M.

[36] A.G. de Souza1, J.C. Oliveira Santos, M.M. Conceição, M.C. Dantas Silva, and S. Prasad, "A thermoanalytic and kinetic study of sunflower oil," Brazilian Journal of Chemical Engineering, vol. 21, 2004, pp. 265–273.

Built-In Self-Testing Methodology and Infrastructure for an EMG Monitoring Sensor Module

Antonio José Salazar Escobar, José Machado da Silva, Miguel Velhote Correia
INESC TEC e Faculdade de Engenharia,
Universidade do Porto,
Porto, Portugal
{†antonio.salazar,jms,mcorreia}@fe.up.pt

Bruno José Mendes
Unidade de Telecomunicações e Multimédia
INESC TEC (formerly INESC Porto),
Porto, Portugal
bruno.mendes@gmail.com

***Abstract* — Wearable technologies provide a refinement to personal monitoring by permitting a long-term on-person approach for capturing physiological signals. Sensors, textile integration, electronics miniaturization and other technological developments are directly responsible for advancements in this domain. However, in spite of the present progress, there are still a number of obstacles to overcome for truly achieving seamless wearable monitoring technology. That concerns, namely, improvements on the reliability of the system at the design stage, including the adoption of built-in self-test and embedded test instruments, features able to detect functional and structural failures. Biopotential monitoring has been part of medicine and rehabilitation protocols for decades now, thus its integration within wearable systems is a natural progression; nonetheless, a number of factors can affect acquisition reliability such as electrode-skin impedance fluctuations and the malfunction of the data-acquisition circuits. This article presents a built-in self-testing approach for an electromyography data acquisition unit, part of a wearable gait monitoring system. The approach makes use of the inter-integrated circuit bus in a dual purpose role, as a communication bus and for stimuli and test response propagation. The targeted tests are electrode-skin impedance checking through a straightforward threshold strategy and detection of functional deviations of the signal conditioning circuit of the electromyography unit, through a digital signature based approach.**

Keywords - BIST; EMG; electrode-skin impedance; digital signatures; structural faults; I2C.

I. INTRODUCTION

The present paper provides an updated and extended description of the work presented by the authors at the GLOBAL HEALTH 2013 conference [1].

An increasing ageing population and consequently rising number of individuals with chronic movement and neurological disorders, are forcing societies to adapt to ever-growing demands. Currently, the capacity of most countries to address such alarming issue is inadequate, due to limited, understaffed and under-resourced facilities, proving ineffectual at times. For example, there were an estimated 10.3 million first-ever stroke survivors in 2005 worldwide [2] and stroke is projected to remain a leading cause of disability-adjusted life years (DALYs) [3] through 2030. Stroke care represents a major burden on global healthcare expenditures, representing roughly 3% of healthcare costs [4]. Despite the cost, there exists a general agreement on the importance of addressing the sequelae of stroke. Concurrently, hip injuries and disorders are also likely to occur with aging. It is estimated that by the year 2030, the number of hip fractures in the USA will reach 289,000, an increase of 12% [5].

In order to safeguard the quality of life of the elderly and individuals with chronic ailments, a paradigm shift in the personal healthcare process is necessary; moving from a reactive (post-event) to a proactive (preventive) stand [6]. Nowadays, information and communication technologies (ICT) are supporting and promoting the aforementioned shift, by pushing health monitoring technologies closer to the end-user, in an effort to reduce costs through remote care. This way, the hospital based healthcare concept is being translated into smaller and more distributed health care services, including the home, up to the point where some health monitoring tasks can be done with the patient living her/his daily activities [7].

The use of portable devices for healthcare enables the early detection of abnormal conditions and facilitates the prescription of ambulatory treatments [8]. Until recently, most research involving the capture and analysis of biomedical and physiological signals has been limited to a laboratory or otherwise controlled environment, making use of cumbersome and costly equipment, which requires specialized facilities and trained personnel. Such practices, although useful in their own right, fail to consider real life scenarios and their impact on the subject. The fast paced developments of body sensor networks (BSN) and wearable technologies (including the so-called smart textiles) have allowed to open the next stage in human behaviour analysis tools, and introduce a new understanding of the interaction of individuals with their surrounding environment [9].

Although wearable and portable biomedical monitoring devices are rapidly becoming a recognized alternative, little attention has been paid to field testing protocols and methodologies, in order to insure measurement reliability, especially on long-term scenarios. When considering the reliability of wearable EMG monitoring systems, one can divide the focus in two main parts: that concerning the data

acquisition system and that dependent on the condition of the electrode-skin interface.

Testing and design for testability (DfT) have become a crucial aspect of most electronic designs; moreover, considering the structural complexities involved in modern packaging technologies. Intellectual property (IP) cores, hybrid technologies and mixed-signal systems have introduced a number of challenges that have dominated testing and development time and cost. Traditional approaches such as parametric characterization or hardware specific testing apparatus are far from providing the cost stabilizing effects achieved by automatic testing equipment (ATE) during the last decades of digital technology revolution. This is of particular concern when considering scenarios for remote or on-location monitoring solutions, which require constant self-diagnostic strategies in order to insure data reliability, such as the ones presented by wearable technologies. Additionally, the continuous tendency for mixed analog and digital (MAD) signal integration within modern designs drives towards new testing solutions, adapted to an evolving set of needs.

Although significant advances in the last decades have been made on the development and use of standard test infrastructures for digital circuits, such is not the case for analog or mixed-signal scenarios, in spite of the availability of the IEEE 1149.4 test bus [10] [11]. Nevertheless, a number of *ad hoc* contributions and strategies exist, where the general idea is the evaluation of an analog response to controlled stimuli, in order to verify expected response behaviors and correlating deviations to specific faults in certain cases [12] [13] [14].

In contrast, the electrode-skin interface [15] [16] [17], as well as its effect on biosignals measurements [18] [15] [16] [17] have been well studied. A number of studies and strategies exist on the electrode-skin impedance characterization domain [19] [20] [21], but the introduction of new electrode types (textile and capacitive for example) present novel challenges, especially in the case of electromyographic (EMG) signals acquisition [22]. The traditional approach for insuring quality electrode based bioelectric monitoring resorts to a thorough skin preparation of the target area, while qualified personnel positions the electrodes based on specific anatomical landmarks, verified with a portable skin-impedance meter or utilizing a test signal of the acquisition system. This approach is not readily applicable to certain subjects, such as elderly, allergenic and pediatric [23], or for most forseen wearable strategies; moreover, variations of the electrode-skin interface impedance are to be expected [24]. Alternatively, methods such as those presented in [25] [26] provide a continuos monitoring of electrode-skin interface through the inclusion of additional hardware such as signal generators, current sources, and filters, used in parallel with the target signal acquisition components.

During biological signals capture, faults within modules (either catastrophic or parametric) can occur in both sensors and signal conditioning circuitry. This is even more acute when these modules are integrated within wearable systems, due to the harsh conditions they are subjected to. The imprecise nature of electrical biosignals, combined with the parametric tolerance of the involved components; not to mention addressing transient variations caused by temperature changes, triboelectric and piezoelectric effects, positioning fluctuations, and sensor contact variation, require adopting testing approaches different from those used in traditional electronic scenarios. In order to improve the reliability of wearable systems, built-in testing and calibration functionalities are required for fault detection, localization and diagnosis prior data is erroneously captured.

This article presents a built-in self-testing (BIST) solution for an EMG sensor module of a wearable system intended for gait analysis. The strategy focuses on resource reutilization and component count minimization, through the reuse of an inter-integrated circuit (I2C) bus as a stimuli/response transport, managed through a novel protocol. Section II provides an overview of the wearable acquisition system for gait analysis on which the present work was based. Section III presents the implemented BIST strategies, as well as the management framework. Section IV summarizes the experimental test results, and Section V highlights the main conclusions and future developments of the work.

II. WEARABLE DATA ACQUISITION SYSTEM FOR GAIT ANALYSIS

Current instrumentation and methods for gait analysis are still expensive and complex, difficult to setup by healthcare staff, hard to operate and uncomfortable for the patient, while requiring a very high level of expertise for data gathering,

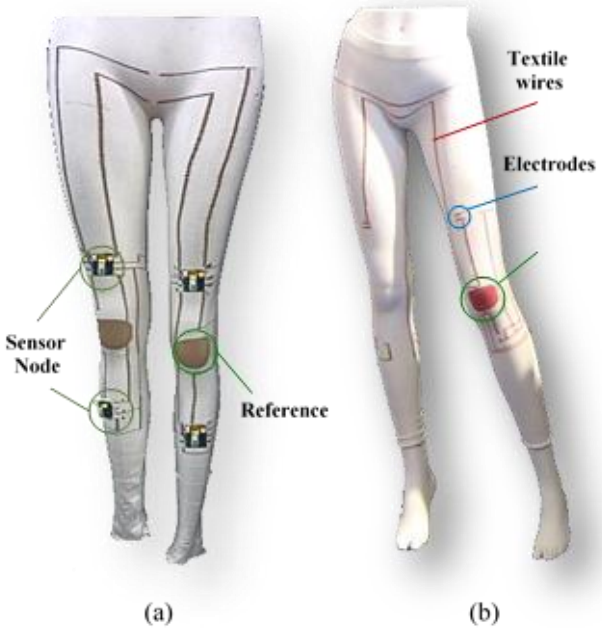


Figure 1. (a) Early prototype of gait analysis system. (b) Textile embedded wires and electrodes.

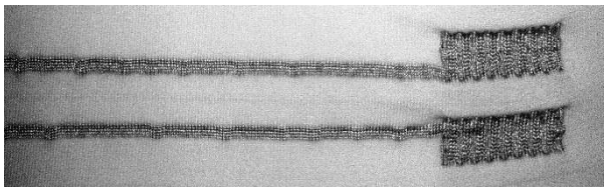


Figure 2. Embedded textile EMG electrodes.

analysis and interpretation. A new wearable instrument infrastructure specifically dedicated to capturing locomotion data is being developed [27]; an early prototype can be observed in Fig. 1 (a), while Fig. 1 (b) presents the textile embedding strategy that permits replacing cumbersome wiring; a close-up of the textile electrodes can be seen in Fig. 2.

This system includes, in a single infrastructure, the means to capture inertial and surface electromyographic signals (sEMG) of the lower limbs. It is presented as a network of sensor nodes interconnected through textile-conductive yarns and provides the measurement of kinematic variables, as well as the sEMG signals that are most important for locomotion. Each node comprises a sEMG sensor, an accelerometer, and a gyroscope, as well as an operation managing microcontroller responsible also for routing data in the established mesh network. EMG electrodes and the interconnections among sensor nodes are sewed on the leggings using yarns made with twisted filaments, each one a polymeric filament covered by a very thin layer of silver. Aggregated information is sent to a personal computer through a Bluetooth wireless link from a central processing module (CPM), as seen in Fig. 3.

The objective is to develop instrumented leggings for measuring human locomotion parameters in a practical and non-invasive way, even for people with strong impairments

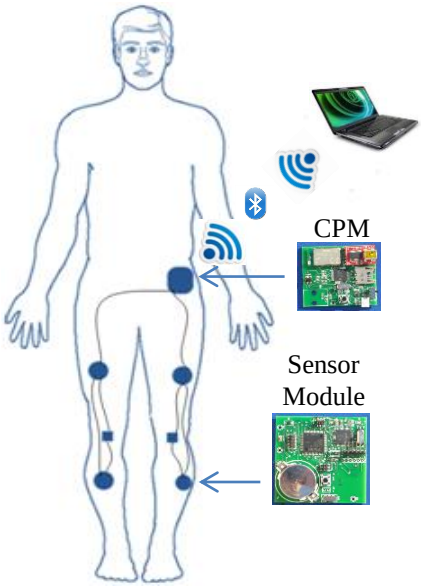


Figure 3. Gait analysis infrastructure.

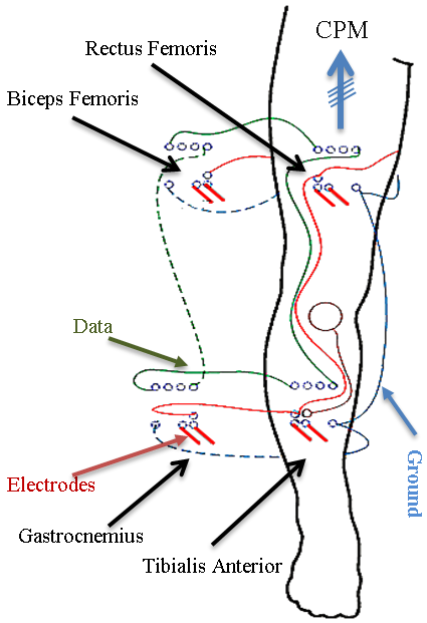


Figure 4. Gait analysis structure detailed view.

or disabilities. It is meant for capturing data, for prolonged periods of time, of typical movement activities under everyday living conditions, without interference or discomfort to the subject. The system allows the measurement of typical kinematic variables of the lower limbs, namely linear and angular movement of thighs and shanks, as well as the myoelectric signals of strategic muscles for locomotion analysis, as seen in Fig. 4, following recommendations from the Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles (SENIAM) project [28] and a team of physiotherapists and specialists in gait analysis.

A. EMG Module

The EMG module contained within each sensor node, shown in Fig. 5, can be divided in two main sections: the electrodes and the signal conditioning circuitry (SCC). The electrodes are grouped in sets of two acquisition electrodes per targeted muscle plus a reference electrode per leg placed on the knee. The SCC comprises the following stages: an instrumentation amplifier, drift removal, filtering, gain adjustment, and a body reference drive feedback connected to the reference electrode. These stages have a predictable behavior established by their configuration and/or combination of elements such as resistors and capacitors, which show an acceptable dispersion of values among them, maintaining the proper functioning of the system. However, variations in the manufacturing process of the components, different life-time degradations, electrical faults (shorts and open circuits), or environmental issues such as, humidity, pressure or temperature, can alter such balance of values.

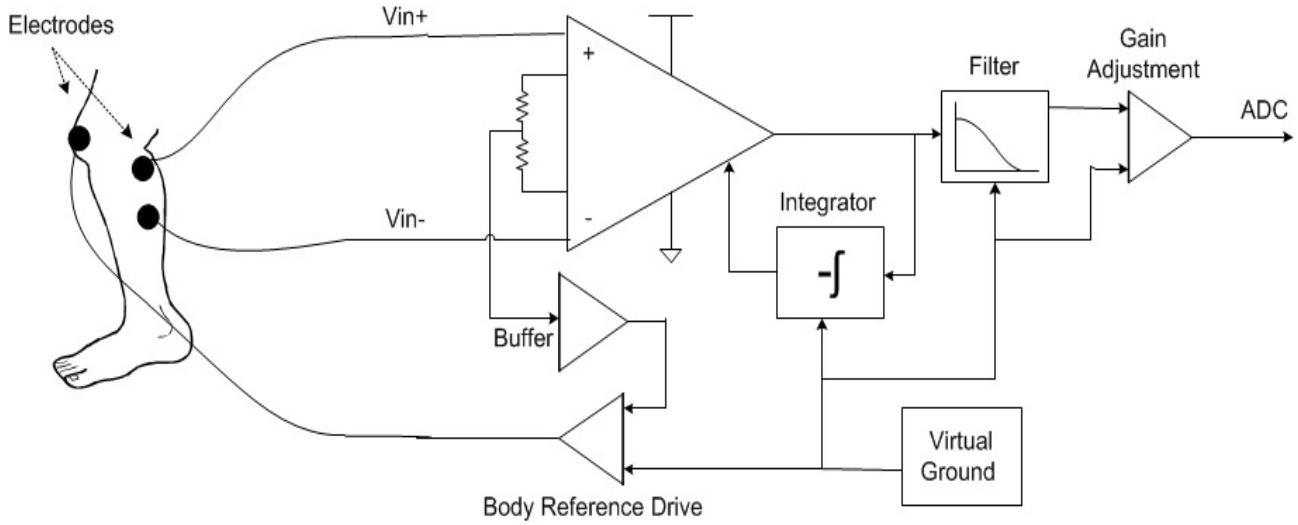


Figure 5. EMG signal conditioning module structure.

Therefore, it is important to ensure that the system is operating within the defined limits before and during its usage, in order to insure the reliability of the captured data.

III. BUILT-IN SELF-TESTING

Built-in self-testing/calibration (BISTC) strategies have traditionally focused on performing detection, diagnosis and repair actions of a specific module, section, component, or IP core [29] [30]. The increasing complexity of modern wearable monitoring technology (WMT) can seldom benefit from strategies that are either too centralized, external data/equipment dependent, or component focused. Communication and area overhead, increased complexity and resources, or energy expenditure, are just a few factors that limit traditional approaches.

In order to address some of the aforementioned limitations, a BIST structure was proposed, which reduces implementation overhead, in terms of design time, pin-count and board area, through the reuse of the I2C bus (already used for connecting the accelerometer and the gyroscope) for testing management purposes, as seen in Fig. 6. Such approach permits taking advantage of the I2C bus, generally present within wearable systems for multi-component communication, as a means for stimuli/response transport, as well as for testing management. Further explanation of the methodology can be found in Section IIIC.

In this particular scenario the embedded instrument refers to the EMG module previously described. The approach seeks to integrate within the module elements required for testing different aspects, such as the electrode-skin impedance for proper sensor contact verification, as well as the signal conditioning circuitry functional response. In such setup, a switching matrix that manages the different signals

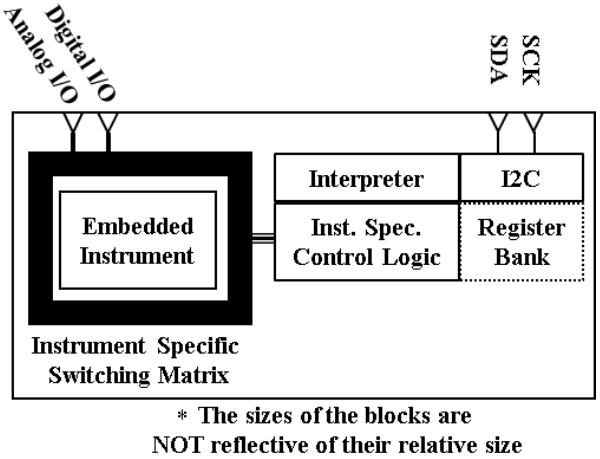


Figure 6. Overview of generic embedded instrument with proposed infrastructure.

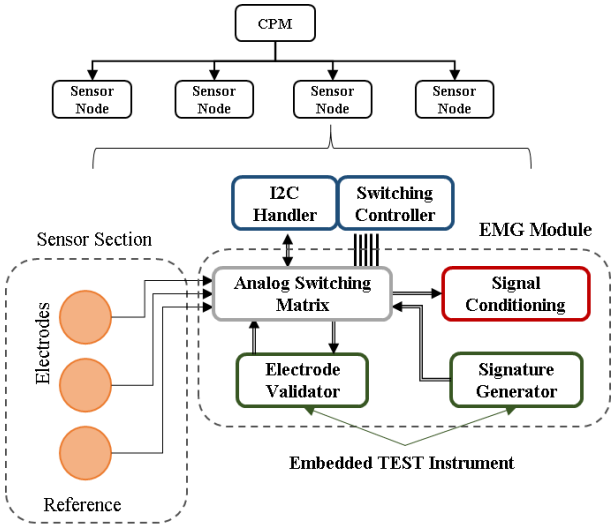


Figure 7. EMG module BIST structure.

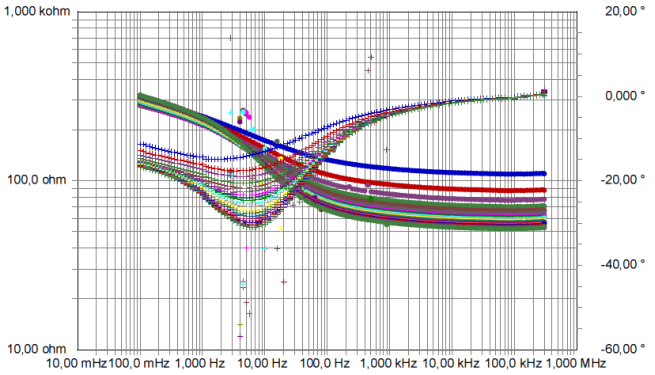


Figure 8. Electrode-Gel time lapse impedance.

routing is necessary to perform the necessary actions, with special consideration to active operation time synchronization, i.e., meaning that safety considerations are also in play due to the nature of the electrode-skin interface. Fig. 7 presents an overview of the strategy applicable to the described scenario, where the signature generator and electrode validator stand for the circuitry utilized for testing; which will be described in the next sections.

A. *Electrode-skin Verification*

Surface electrodes are likely the most utilized sensors for capturing electrical biosignals measurements, such as electrocardiography, electromyography, electro-encephalography, electrooculography, bioimpedance, impedance tomography, among others. The contact impedance achieved in the electrode-skin interface affects biosignals measurements (as stated earlier), a matter of concern, traditionally solved through, namely, skin preparation procedures, equipment checking, and electrode replacement. Even under such controlled conditions, variations of the electrode-skin contact impedance are to be expected. However, in applications such as daily activities monitoring and the performance measurement of an athlete, or other scenarios where the individuals will have to position the electrodes themselves or the electrodes are integrated within a garment, careful positioning and skin preparation cannot be guaranteed.

Fig. 8 presents an eight hour time lapse measurement of the impedance of a commercially available disposable pediatric Ag/AgCl foam electrodes, of 1 cm of diameter core and 3 cm of diameter foam (DORMO, ref SX-30) over an Agar based gel, following the preparation procedures presented in [31]. A conventional three electrode setup with a GAMRY Series G-300 Galvanostat with 50 µA stimuli compatible with IEC 60601-1 standard [32] and ANSI/AAMI EC12:2000 recommendations [33], was performed in a controlled environment in order to ascertain the time variation of the electrode material interface impedance. As can be observed in Fig. 8, there is a prolonged settling period caused mainly by hydrophilic effects and temperature equilibration between the two interacting elements, similar to

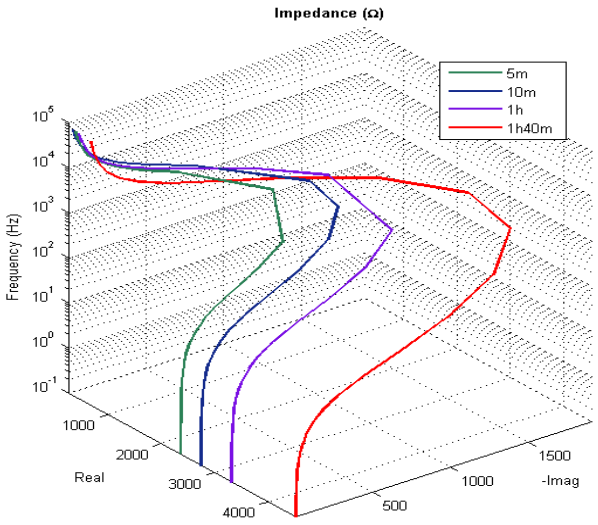


Figure 9. Time lapsed electrode-skin impedance.

the effect reported on the electrode-skin interface [34], observed in Fig. 9 as well.

Fig. 9 presents the measurements of an electrode-skin interface of a human volunteer. The skin was prepared with a straightforward technique, limited to light shaving, degreased with alcohol, dead cells removed with a soft brush, later cleansed with soap and water, and allowed to rest for 10 minutes. The target electrode and its reference were located roughly 3 cm proximal to the elbow and the signal injection electrode roughly 6 cm from the center point of the target-reference electrode line, proximal to the forearm mid-point (all measurements were considered from the center of the electrodes), the ground electrode was located at the contralateral posterior side elbow, as seen in Fig. 10. The subject was then placed within a Faraday cage, in order to reduce electromagnetic noise, following a sitting position with the forearm containing the electrodes resting

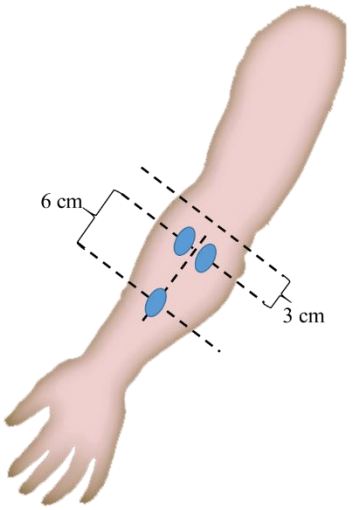


Figure 10. Electrode placement.

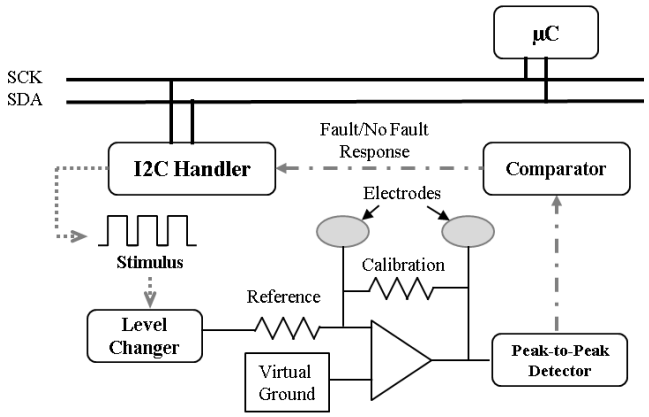


Figure 11. Electrode-skin verification structure.

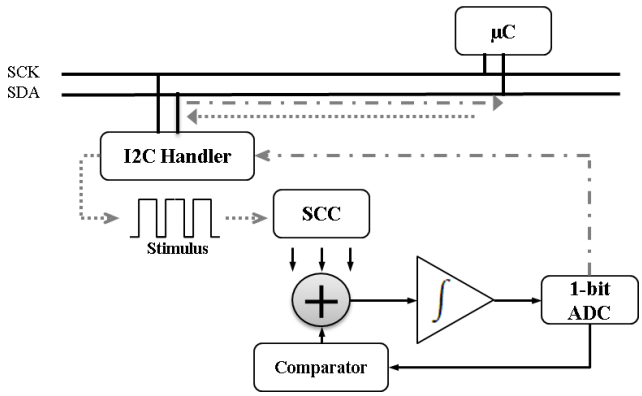


Figure 12. SCC test infrastructure.

horizontally. As can be observed in Fig. 9, a similar hydrophilic phenomenon occurred as in the case of the electrode-gel. Such phenomenon is known by healthcare personnel, reason for which most electrode related procedures include a settling period prior to measurements.

The phenomenon described above is worth mentioning in order to illustrate the variability of the electrode-skin interface; although it is not the greatest concern within a wearable system. Conventionally, analog faults are classified as hard (catastrophic) or soft (parametric), referring to the trace continuity; however, when considering sensors, the fault classes are not so well defined. For instance from a data centric point of view one can summarize, following [35] [36] [37] [38] [39]:

• *Constant* or dead: measures provide invariant arbitrary values, uncorrelated to the observed phenomenon.

• *Random noise*: increased variance of the target sensor measurements.

• *Short*: sharp momentary irregularities between measurement points.

• *Accumulative or drift*: continuous deviation trend from the correct value, expressible through a deterministic relation with true value, possibly cause by age, decay, damage, etc.

In the case of textile electrodes the problems are exacerbated, due to their sensitivity to pressure, fabric stretching, and motion artifacts [40] [41] [42]. In addition, textile electrodes and wires, such as those presented back in Fig. 2, are relatively new technologies and strong behavioral models have not been well established when compared to pre-gel electrodes; which complicates issues related to interface impedance variability.

Several approaches have been implemented, through the years, for the measurement and monitoring of the electrode-skin impedance, such as the ones presented in [18] [20] [21] [25] [26] [34]. An electrode-skin impedance verification circuit was developed following a straightforward approach, based on the injection of a low amplitude stimulus current (less than 10µA) in order to ascertain an electrode pair target load. Individual electrode-skin interface strategies generally utilize a three electrodes approach (one electrode-skin contact target and two others for sinking and voltage reference respectively).

However, an electrode pair-wise verification was preferred in this case, in order to maintain simplicity. A single-supply current to voltage converter was used as observed in Fig. 11, which includes a calibration resistor in parallel with the target load in order to control threshold limits and avoid open feedback scenarios. The configuration follows that of a current controlled voltage source, where the current observed across the reference resistor flows towards the electrodes and calibration resistance generating a voltage proportional to the impedance under test. Momentary sharp irregularities are limited by the calibration resistor, as well as stabilizing capacitors (not observed in the figure). The magnitude of the stimulus current is a paramount consideration due to the possible negative effects in the human body, hereby achieved through the introduction of a limiting reference resistor. A local DC reference can be applied as stimulus in addition to a virtual ground compensated square wave signal sent through the I2C bus.

B. Signal conditioning circuitry verification

Common-mode rejection, amplification and filtering are regular stages of any electrode based signal conditioning circuit [42][48]. These are required to reduce the effects of common-mode potentials, random noise, motion and power-line artifacts, as well as to effectively retrieving the components of interest of the measured signal. Amplification factors and cut-off frequencies are dependent on the signal type [42][48], and deviations can cause unwanted elements to be introduced into the captured signal.

The test of the SCC (see Fig. 12) is achieved by means of the injection of an impulse stimulus at the input, fusion of the response of targeted nodes within the SCC, and the collection of the final response in the form of a digital signature that can be compared against a response table, composed by a set of signatures corresponding to the tolerance determined by acceptable components variations.

Initially, a Built-In Logic Block Observer (BILBO)-like [43] based approach was attempted in which the stimulus was

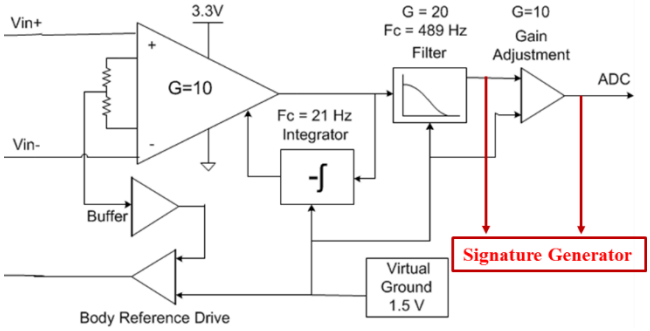


Figure 13. SCC target test nodes.

provided by an LFSR (Linear Feedback Shift Register), and the response of the SCC was collected by a Multiple Input Signature Register (MISR). These are solutions commonly found in the structural test of digital circuits, which are prone to aliasing errors, i.e., there is a (small) probability that the signature of a bad circuit is the same as that of a good circuit. In fact, such solution proved to be ineffective in the present case, since large variations of some components of the SCC module rendered signatures not so different from those obtained considering valid values, thus providing unreliable and ambiguous error detecting methods for this specific purpose.

Alternatively, a different testing approach was chosen, where a delta-sigma (ΔΣ; or sigma-delta, ΣΔ) like modulator is used to convert the SCC response into a bit stream, being the I2C bus used for stimulus generation and response capture purposes (Fig. 12). An I2C bus driven stimulus was preferred over a locally generated one, in order to reduce local sources of noise (such as clocks), gain increased stimulus shaped flexibility, and reuse of existing resources. The target observation nodes were determined through a sensitivity analysis after a SPICE simulation, which established that the low-pass filter output and the ADC input are the nodes that best reflect variations within the components of the SCC, seen in Fig. 13.

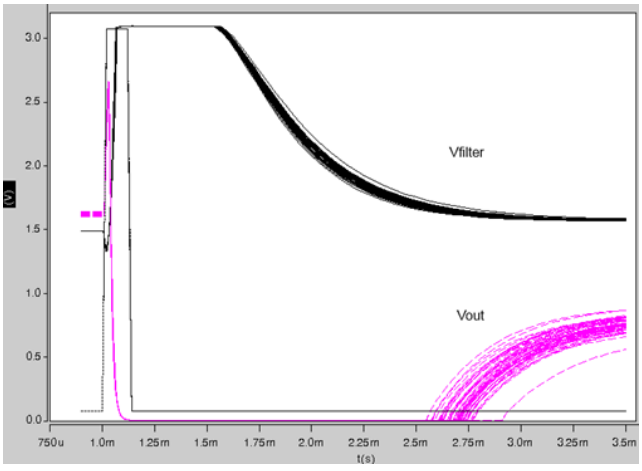


Figure 14. SCC response to the test stimulus.

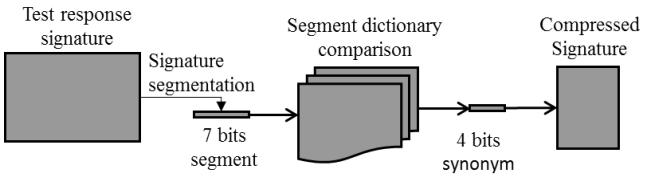


Figure 15. Compression algorithm overview.

The SCC test impulse is designed to stimulate the SCC frequency bandwidth and amplitude full range. The observation of different analog nodes and their compression into a single bit stream improves observability and saves test response resources and time. This way, the need for an analog test bus line, the inclusion of a bulky analog to digital converter, and the need for a multiplexed test response acquisition are avoided. Fig. 14 shows the test impulse response, of the two selected inputs to the signature generator for a Monte Carlo simulation considering 10% variations of the SCC's capacitors and resistors values.

In order to reduce noise along the communication lines, complexity and total area of the test circuit, it was decided to differ from traditional ΔΣ modulators, by eliminating the flip-flops generally present between comparators. The output of the signature generator is kept in a non-ground state through the use of a pull-up resistor until the test stimulus forces the first '0', to ensure a known initial condition and thus a predictable start sequence, compatible with I2C as well. After such start event, the signal is captured every 10 µs during 1.05 ms generating a 105-bit long signature.

The resulting signature is acquired through the I2C bus by the local processing module, which applies a window bit density filtering and Ziv-Lempel based lossless compression algorithm [44]. As the SCC test response presents variations due to the acceptable tolerances of its components, the golden signature is actually a set comprising the signatures of different admissible responses.

The Ziv-Lempel based lossless compression algorithm replaces repetitive bit sequences by a shorter code, as described in the following pseudo-code and observed in Fig. 16:

array $= \Delta \Sigma$ *output*
foreach segment from array
  if segment $\in$ *dictionary*
    then signature + = segment
foreach segment from signature
  if segment $\in$ *2nd dictionary*
    then final signature += segment

Figure 16. Pseudo-code for compression algorithm.

The use of this compression algorithm is twofold. On one hand, compression allows reducing the length of data to be transmitted along the wired network from the sensor node to the CPM, as well as through the Bluetooth link, thus reducing communication time and power. On the other hand, it allows to recover the original analog response at an external

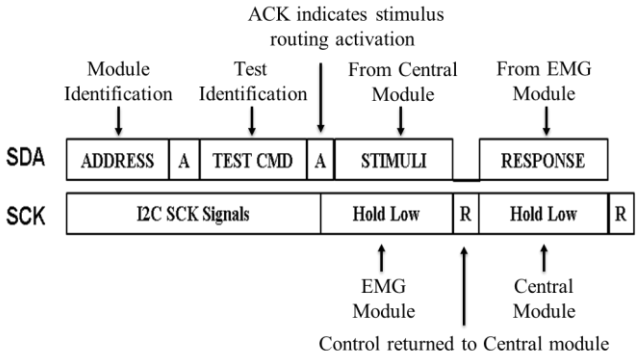


Figure 17. I2C compatible sequence for Stimulus/Response transport.

processing unit, using a corresponding decompressing algorithm.

C. I2C bus and test management handler

A testing and/or calibration strategy for WMT benefits from distributed or multi-sensor aware approaches. Such approach could seek to maintain data reliability after the recognition of deviating degradation patterns on sensors that could provide insight into system problems due to, e.g., improper sensor positioning, induced electrical effects due to movement (turboelectric and piezoelectric effects), structural flaws and other factors that require the coverage provided by the analysis of multiple temporal instances, redundant structures comparison, or introspection into fused data components. In order to manage the previously mentioned approach, a testing framework was designed based on a protocol named SCPS [45]; which seeks to standardize the command sequence for sensor acquisition/testing access.

In the present case, an instruction enables a testing procedure, activating pre-determined routing configurations. It is possible as well to use an I2C compatible sequence for transporting stimuli and responses to and from the target module as described in Fig. 17.

The sequence sets up the appropriate routing configuration through acknowledgement of a test command and uses the next two SCK low-state for stimulus and response transport. In order to avoid start/stop events from occurring, the master element insures a low-state of the SDA prior to SCK high. A re-start or stop event can then be used at the event of the sequence to finalize the action.

IV. RESULTS

The electrode verification circuit was first simulated on Multisim 11.0.775, for functional and electrical parameters validation and to confirm the suitability of the arrangement. Preliminary experiments were then performed on an Agar based gel for performance verification prior to testing on human volunteers. A number of signals were used for behavioral confirmation, as can be seen in Figs. 18, 19, and 20.

The electrode-skin impedance was changed through variations of the contact surface between the electrode and the skin, Fig. 18 presents different responses of the circuit to such variations for a 10 µA DC stimulus, demonstrating its sensivity to the electrode-skin impedance variations, thus compatible with a threshold based fault detection approach. Fig. 19 and Fig. 20 present corresponding responses to square-wave and sine-wave, respectively, stimuli of matching peak to peak amplitude, respectively (limited to 50 µA). These responses also demonstrate their sensitivity to the reactive component of the electrode-skin impedance (through phase and time response effects), by presenting a measurable

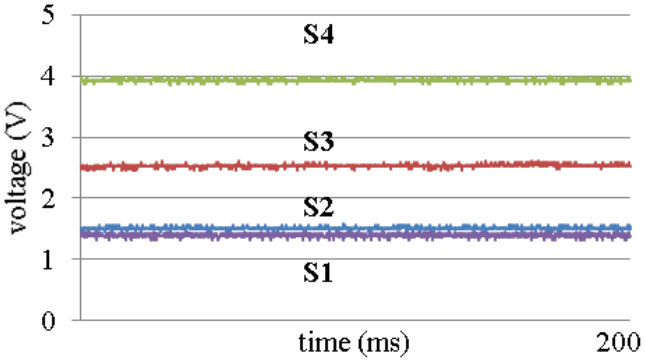


Figure18. DC Stimuli response for varying conditions, where S1 is the stimuli, S2 is low impedance response, S3 is an expected impedance response and S4 is a high impedance response.

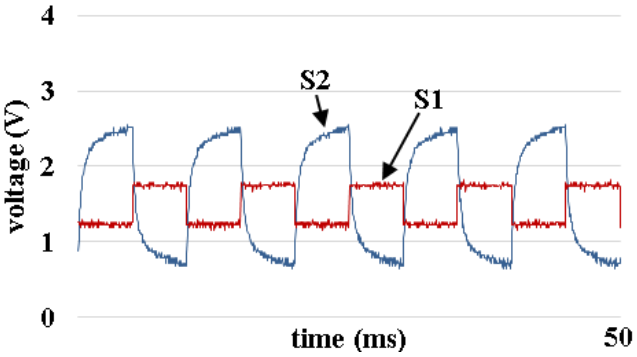


Figure 19. Square wave of 100 Hz stimuli, where S1 is the stimuli, and S2 is an expected impedance response.

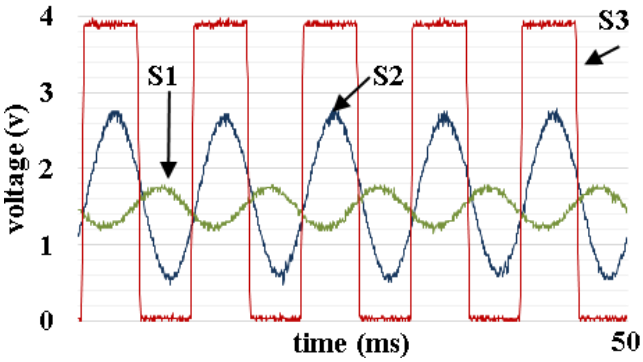


Figure 20. Sine wave of 100 Hz stimuli response for varying conditions, where S1 is the stimuli, S2 is a normal impedance response and S3 is a high impedance response.

phase difference in the case of S2 within Fig. 19 and through the behavior of S2 in Fig. 20, showing a classical charging/discharging behavior. On the other hand, the S3 saturation level seen in Fig. 19 corresponds to an unacceptable over the limit electrode-skin impedance case.

A. *Signature generation results*

The SCC and the ΔΣ circuits were simulated within a SPICE like simulator, using the models of manufacturers for the operational amplifiers, comparators and analog switches. Figs. 21 to 24 show the waveforms obtained in response to the test impulse, for golden and faulty cases. The input pulse stimulus was designed considering the circuit time constants and the I2C time specifications – I2C's fast-mode and fast-mode plus specifications impose minimum durations of 0.6 µs and 0.26 µs high periods, respectively [46].

Figs. 22, 23, and 24 present faulty responses for the cases of, respectively, a 5% reduction of the filter gain, a 30% reduction of the low-pass filter capacitor value, and an open connection in the instrumentation amplifier – Fig. 21 shows the golden response. The sequences of pulses presented in each case are the corresponding outputs of the ΔΣ modulator. It can be seen that, after comparing these sequences with the golden case, the three faults are detectable as different bit streams are generated.

The direct capture of these test responses is possible because the I2C sampling frequency allows doing it with an adequate resolution, i.e., no pulses are lost.

Experimental results were obtained with a demonstration prototype. For that purpose faults were introduced in the SCC in some of the most critical components for the proper operation of the circuit. According to the literature, the low-pass filter cutoff frequency (f_c) for EMG signal conditioning

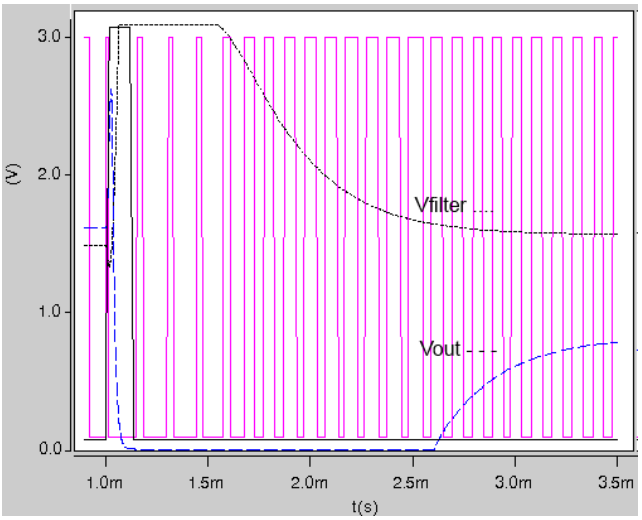


Figure 21. Signature generator input signals (Vfilter; Vout) and ΔΣ output – golden case.

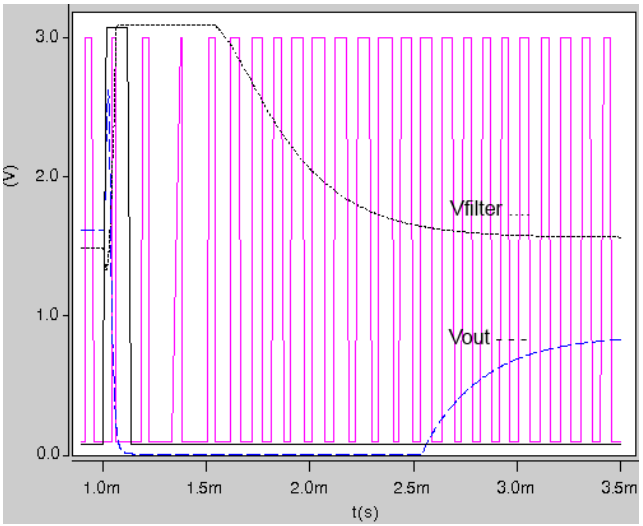


Figure 22. Signature generator input signals (Vfilter; Vout) and ΔΣ output – 5% reduction of the filter gain.

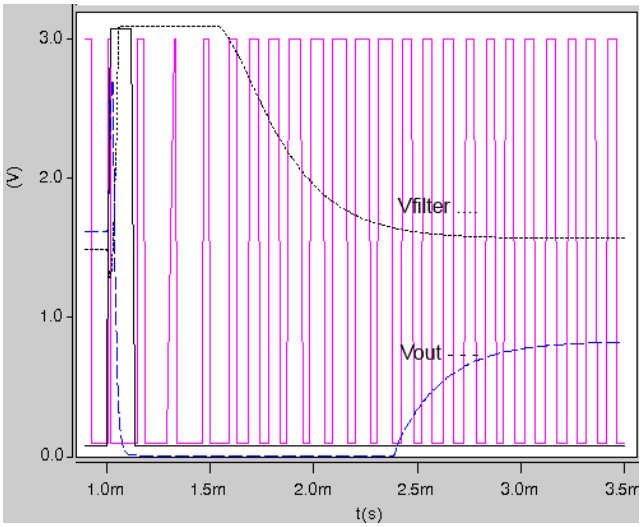


Figure 23. Signature generator input signals (Vfilter; Vout) and ΔΣ output – 30% reduction of a capacitor value.

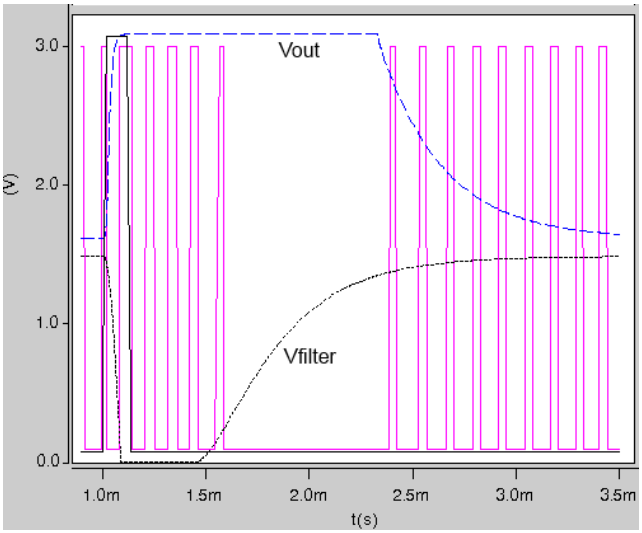


Figure 24. Signature generator input signals (Vfilter; Vout) and ΔΣ output – open circuit in the instrumentation amplifier.

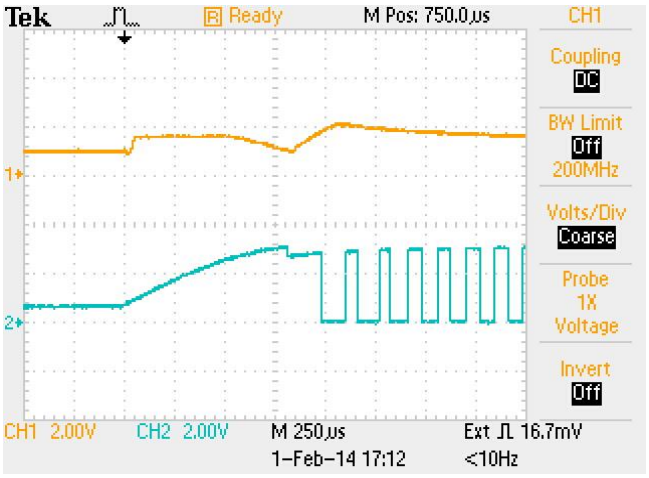


Figure 25. Cutoff frequency at 482 Hz.

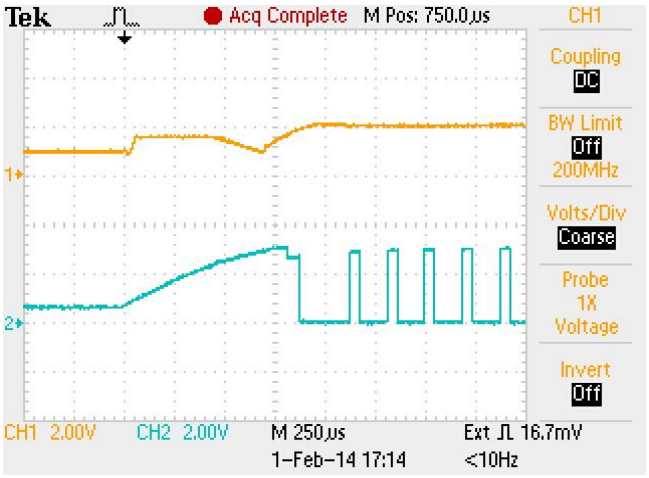


Figure 26. Cutoff frequency at 1.45 kHz.

circuits should be considered in the range from 500 Hz [28] to 700 Hz [47] or even 1000 Hz [48]. In our case, the tolerance band of the cut-off frequency imposed by the admissible components variations, shows a width from 480 Hz to 800 Hz. Deviations were introduced in some components in order to change the cutoff frequency to values inside and out of the tolerance band, i.e., fc = [300, 492, 688 Hz, 1450, 10130] Hz.

In Figs. 25 and 26 one can see the input (yellow/top signal) and the output (blue/bottom signal) of the test signature generator in case of, respectively, an acceptable and an unacceptable cutoff frequency. The bit-stream that is then obtained presents a duration of 1.05 ms, or a length of 105 bits.

After compressing, the captured bit-streams are compared against the expected golden responses. This is achieved by evaluating the bits in specific windows, were common pattern are produced among good responses. Fig. 27 shows the compressed signatures obtained for valid (top three) and faulty (bottom four) responses, corresponding to deviations of the cutoff frequency to fc = [300, 1450, 2500, 10130]. A

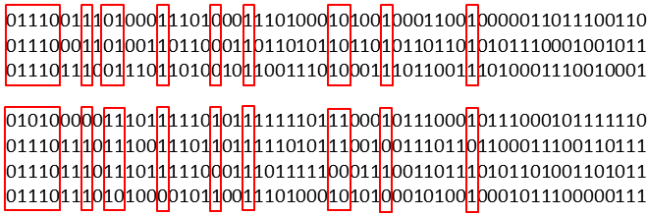


Figure 27. Valid (top three) and faulty (bottom four) test signatures.

circuit is considered faulty when the bits in the evaluation windows are different from the expected ones.

V. Conclusion and Future Work

A significant research effort has been made to develop economic and reliable tools capable of providing, as non-obtrusive as possible, portable monitoring of biological signals. Many of the solutions that have been proposed so far are based on placing sensors on garments that can be worn without any extra special care. However, the testing of these electronic systems has not deserved the same attention and solutions are required in order to improve that reliable data is captured and used in medical protocols.

The inclusion of testing features within wearable technologies is of paramount importance due to their portable nature and target monitoring scenarios. In contrast with medical devices found within hospital and healthcare facilities, wearable healthcare systems do not necessarily count with the support of professional personnel to verify their placement and continuous operation.

A mixed-signal built-in self-test infrastructure and methodology is presented that addresses the in-situ verification of the electrode-skin interface, as well as the functionality of the signal conditioning circuitry of a wearable electromyographic data acquisition system. The approach being proposed uses an I2C bus for test event management and stimuli/response transport, through a protocol meant for resource optimization and sensor group testing strategies. The electrode-skin interface is evaluated after the measurement of the complex impedance and the signal conditioning circuitry is tested after comparing its impulse response with the expected golden response. A Ziv-Lempel compression algorithm is used to allow transferring a shorter version of the captured bit stream, thus saving communication time and power, while preserving the possibility of reconstructing the impulse response of the circuit. The simulations and circuit implementation results confirm the validity of the approaches being proposed and reveal their compatibility to the target system and available resources.

This work is expected to be further developed with the design of new versions of the proposed test instruments, namely with the on-chip implementations, and the design of instruments to test other parts of the systems, always using the SCPS infrastructure for test data and operations management.

ACKNOWLEDGMENT

This work was financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the *Fundação para a Ciência e a Tecnologia* (Portuguese Foundation for Science and Technology) within project ProLimb PTDC/EEA-ELC/103683/2008, grant SFRH/BD/61396/2009, and project ELESIS/ENIAC - European Library-based flow of Embedded Silicon test Instruments.

REFERENCES

[1] A. Salazar, B. Mendes, J. Da Silva, and M. Correia, "Built-in self-testing infrastructure and methodology for an EMG signal capture module," Proc. 2nd International Conference on Global Health Challenges (GLOBAL HEALTH 2013), Lisbon, Portugal, 2013, pp. 43-48.

[2] K. Strong, C. Mathers, and R. Bonita, "Preventing stroke: saving lives around the world," Lancet Neurol., vol. 6, no. 2, 2007, pp. 182-187.

[3] A. D. Lopez, C. D. Mathers, M. Ezzati, D. T. Jamison, and C. Murray, "Global and regional burden of disease and risk factors, 2001: systematic analysis of population health data," The Lancet, vol. 367, no. 9524, May 2006, pp. 1747 – 1757.

[4] S. M. Evers, J. N. Struijs, A. J. Ament, M. L. van Genugten, J. H. Jager, and G. A. van den Bos, "International comparison of stroke cost studies," *Stroke*, vol. 35, no. 5, April 2004, pp. 1209-1215 , doi: 10.1161/01.STR.0000125860.48180.48.

[5] J. A. Stevens and R. A. Rudd, "The impact of decreasing U.S. hip fracture rates on future hip fracture estimates," Osteoporosis International, vol. 24, no. 10, Oct. 2013, pp. 2725-2728, doi:10.1007/s00198-013-2375-9.

[6] J. Yoo and H. Yoo, "Emerging low energy wearable body sensor networks using patch sensors for continuous healthcare applications," Proc. of the Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC), Aug.-Sept. 2010, pp. 6381-6384, doi: 10.1109/IEMBS.2010.5627299.

[7] M. Engin, A. Demirel, E. Z. Engina, and M. Fedakarc, "Recent developments and trends in biomedical sensors," Measurement, vol. 37, no. 2, March 2005, pp. 173-188, doi: 10.1016/j.measurement.2004.11.002.

[8] A. Pantelopoulus and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," IEEE Transactions on Systems, Man, and Cybernetics, vol. 40, no. 1, Jan. 2010, pp. 1-12, doi: 10.1109/TSMCC.2009.2032660.

[9] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. Leung, "Body area networks: a survey," Mobile Networks and Applications, vol. 19, no. 2, April 2011, pp. 171-193, doi:10.1007/s11036-010-0260-8.

[10] L. T. Wang, C. W. Wu, and X. Wen, VLSI test principles and architectures: design for testability, Academic Press, 2006.

[11] "IEEE 1149.4 Mixed-Signal Test bus Working group," [Online]. Available: http://grouper.ieee.org/groups/1149/4. [Accessed May 2014].

[12] L. S. Milor, "A tutorial introduction to research on analog and mixed-signal circuit testing," IEEE Trans. on Circutis and Syst., vol. 45, no. 10, Oct. 1998, pp. 1389-1407, doi: 10.1109/82.728852.

[13] B. Kaminska and K. Arabi, "Mixed signal DFT: a concise overview," Proc. Int. Conf. on Computer Aided Design, San Jose, Nov. 2003, pp. 672-679, doi:10.1109/ICCAD.2003.1257882.

[14] S. R. Das, J. Zakizadeh, S. Biswas, M. H. Assaf, A. R. Nayak, E. M. Petriu, W. Jone, and M. Sahinoglu, "Testing analog and mixed-signal circuits with built-in hardware: a new approach," IEEE Trans. on Instr. and Measurement, vol. 56, no. 3, June 2007, pp. 840-855, doi:10.1109/TIM.2007.894223.

[15] J. J. Huang and K. S. Cheng, "The effect of electrode-skin interface model in electrical impedance imaging," in Proc. Int. Conf. on Image Processing, vol. 3, Oct. 1998, pp. 837-840, doi:10.1109/ICIP.1998.727384.

[16] D. J. Hewson, J. Druchêne, and J. Y. Hogrel, "Changes in impedance at the electrode-skin interface of surface EMG electrodes during long-term EMG recordings," in Proc. 23rd IEEE EMBS, vol. 4, Oct. 2001, pp. 3345-3348, doi: 10.1109/IEMBS.2001.1019543.

[17] E. Kappenman and S. Luck, "The effects of electrode impedance on data quality and statistical significance in ERP recordings," Psychophysiology, vol. 47, no. 5, Sept. 2010, pp. 888-904, doi:10.1111/j.1469-8986.2010.01009.x.

[18] I. Zepeda-Carapia, A. Márquez-Espinoza, and C. Alvarado-Serrano, "Measurement of Skin-Electrode Impedance for a 12-lead Electrocardiogram," Proc. Int. Conf. on Electrical and Electronic Eng,, Sept. 2005, pp. 193-195, doi: 10.1109/ICEEE.2005.1529606.

[19] Z. Li, Z. He, W. Wang, and C. Ren, "Measurement System for electrical impedance and EGG," Proc. 5th Int. Conf. on Information Techn. and Application on Biomedicine, May 2008, pp. 495-498, doi: 10.1109/ITAB.2008.4570598.

[20] C. A. Grimbergen, A. C. MettingVanRijn, and A. Peper, "A method for the measurement of the properties of individual electrode-skin interfaces and the implications of the electrode properties for preamplifier design," Proc. 14th Annual Int. Conf. IEEE EMBS, vol. 6, Oct. 1992, pp. 2382-2383, doi: 10.1109/IEMBS.1992.5761414.

[21] E. Spinelli, M. A. Mayosky, and R. Pallás-Areny, "A practical approach to electrode-skin impedance unbalance measurement," IEEE Trans. on Biomedical Eng., vol. 53, no. 7, July 2006, pp. 1451-1453, doi: 10.1109/TBME.2006.875714.

[22] R. Merletti, "The electrode–skin interface and optimal detection of bioelectric signals," Physiol. Meas., vol. 31, no. 10, 2010, doi: doi:10.1088/0967-3334/31/10/E01.

[23] C. Assambo, A. Baba, R. Dozio, and M. J. Burke, "Determination of the parameters of the electrode-skin impedance model for ECG measurements," Proc. of the 6th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Comm., Feb. 2007, pp. 90-95.

[24] K. L. Kilgore, P. H. Peckham, M. W. Keith, and G. B. Thrope, "Electrode Characterization for functional application to

upper extremity FNS," IEEE Trans. Biomedical Eng., vol. 37, no. 1, Jan. 1990, pp.12-21, doi: 10.1109/10.43606.

[25] T. Degen and H. Jäckel, "Continuous monitoring of electrode-skin impedance mismatch during bioelectric recordings," IEEE Transactions on Biomedical Eng., vol. 55, no. 6, June 2008, pp. 1711-1715, doi: 10.1109/TBME.2008.919118.

[26] S. Kim, R. F. Tazicioglu, T. Torfs, B. Dilpreet, P. Julien, and C. Van Hoof, "A 2.4 uA continuous-time electrode-skin impedance measurement circuit for motion artifact montoring in ECG acquisition systems," Proc. IEEE Symposium on VLSI Circuits, June 2010, pp. 219-220, doi: 10.1109/VLSIC.2010.5560290.

[27] A. Zambrano, F. Derogarian, R. Dias, M. J. Abreu, A. Catarino, A. M. Rocha, J. M. da Silva, J. C. Ferreira, V. G. Tavares, and M. V. Correia, "A wearable sensor network for human locomotion data capture," Proc. 9th Int. Conf. on Wearable Micro and Nano Tech. for Personalized Health (pHealth), Porto, vol. 177, 2012, pp. 216-223.

[28] H. J. Hermens, B. Freriks, R. Merletti, D. Stegeman, J. Blok, G. Rau, C. Disselhorst-Klug, and G. Hägg, "European Recommendations for Surface ElectroMyoGraphy", Enschede, the Netherlands: Roessingh Research and Development, 1999.

[29] M. H. Zaki, S. Tahar, and G. Bois, "Formal verification of analog and mixed signal designs: a survey," Microelectronics Journal, vol. 39, Dec. 2008, pp. 1395-1404, doi:10.1016/j.mejo.2008.05.013.

[30] M. Burns and G. W. Roberts, "An introduction to mixed-signal IC test and measurement", 2nd ed., New York, USA: Oxford University Press, 2011.

[31] P. Tallgren, S. Vanhatalo, K. Kaila, and J. Voipio, "Evaluation of commercially available electrodes and gels for recording of slow EEG potentials," Clinical Neurophysiology, vol. 116, no. 4, Nov. 2005, pp. 799-806, doi:10.1016/j.clinph.2004.10.001.

[32] Commission Electrotechnique Internationale, IEC 60601-1, 3rd ed., Geneva, Switzerland: IEC, 2005.

[33] Association for the Advancement of Medical Instrumentation, Disposable ECG electrodes, American National Standards Institute, Inc., 2000.

[34] S. Rogers, "Sensor noise fault detection," Proc. of American Control Conference, vol. 5, June 2003, pp. 4267-4268, doi: 10.1109/ACC.2003.1240506.

[35] J. Feng, G. Qu, and M. Potkonjak, "Sensor calibration using nonparametric statistical characterization of error models," Proc. of IEEE Sensors, vol. 3, Oct. 2004, pp. 1456-1459, doi: 10.1109/ICSENS.2004.1426461.

[36] T. Bourdenas and M. Sloman, "Towards self-healing in wireless sensor networks," Proc. of Workshop on Wearable and Implantable Body Sensor Networks, June 2009, pp. 15-20, doi: 10.1109/BSN.2009.14.

[37] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: detection methods and prevalence in real-world datasets," ACM Transactions on Sensor networks, vol. 6, no. 3, pp. 1-34, June 2010, doi: 10.1145/1754414.1754419.

[38] E. U. Warriach, K. Tei, T. A. Nguyen, and M. Aiello, "Fault detection in wireless sensor networks: a hybrid approach," Proc. of Int. Conf. on Information Processing in Sensor Net., pp- 87-88, April 2012, doi: 10.1145/2185677.2185690.

[39] M. M. Puurtinen, S. M. Komulainen, P. K. Kauppinen, and J. A. Malmivuo, "Measurement of noise and impedance of dry and wet textile electrodes and textile electrode with hydrogel," Proc. 28th IEEE Eng. in Medicine and Biology Society, New York City, vol. 1, pp. 6012-6015, 2006.

[40] J. C. Marquez, F. Seoane, E. Välimäki, and K. Lindecrantz, "Textile Electrodes in Electrical Bioimpedance Measurements - A Comparison with Conventional Ag/AgCl Electrods," Proc. IEEE Eng. in Medicine and Biology Society, Minneapolis, Sept. 2009, pp. 4816-4819, doi: 10.1109/IEMBS.2009.5332631.

[41] L. Beckmann, C. Neuhaus, G. Medrano, N. Jungbecker, M. Walter, T. Gries, and S. Leonhardt, "Characterization of textile electrodes and conductors using standardized measurement setups," Physiological Measurements, vol. 31, no. 2, Feb. 2010, pp. 233-247, doi: 10.1088/0967-3334/31/2/009.

[42] J. G. Webster, Medical Instrumentation: Application and Design, 4th ed., Wiley, 2009.

[43] M. L. Bushnell and V. D. Agrawal, Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits, vol. 17, SCI-TECH Publishing Co., 2000.

[44] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," IEEE Trans. on Information Theory, vol. 23, no. 3, May 1977, pp. 337-343, doi: 10.1109/TIT.1977.1055714.

[45] A. J. Salazar, M. V. Correia, and J. M. Da Silva," SCPS: Mixed-signal Test and Measurement Framework for Wearable Monitoring Systems," unpublished patent request submitted to Patent Office.

[46] "I2C-bus specification and user manual, UM10204, Rev. 5," 2012.

[47] V. R. Zschorlich, "Digital filtering of EMG-signals," Electromyography and clinical neurophysiology, vol. 29, no. 2, pp. 81-86, 1989.

[48] B. Gerdle, S. Karlsson, S. Day, and M. Djupsjöbacka, "Acquisition, Processing and Analysis of the Surface Electromyogram," in Modern Techniques in Neuroscience, Berlin, Germany, Springer, 1999, pp. 705-755.

2D-Packing Images on a Large Scale: Packing a Billion Rectangles under 10 Minutes

Dominique Thiebaut
Dept. Computer Science
Smith College
Northampton, Ma 01063
Email: dthiebaut@smith.edu

***Abstract*—We present a novel heuristic for 2D-packing of rectangles inside a rectangular area where the aesthetics of the resulting packing is amenable to generating large collages of photographs or images. The heuristic works by maintaining a sorted collection of vertical segments covering the area to be packed. The segments define the leftmost boundaries of rectangular and possibly overlapping areas that are yet to be covered. The use of this data structure allows for easily defining ahead of time arbitrary rectangular areas that the packing must avoid. The 2D-packing heuristic presented does not allow the rectangles to be rotated during the packing, but could easily be modified to implement this feature. The execution time of the present heuristic on various benchmark problems is on par with recently published research in this area, including some that do allow rotation of items while packing. Several examples of image packing are presented. A multithreaded version of our core packing algorithm running on a 32-core 2.8 GHz processor packs a billion rectangles in under 10 minutes.**

Keywords–bin packing; rectangle packing; multi-threaded and parallel algorithms; heuristics; greedy algorithms; image collages.

I. INTRODUCTION

We present a new heuristic for placing two-dimensional rectangles in a rectangular surface. The heuristic keeps track of the empty area with a new data structure that allows for the natural packing around predefined rectangular areas where packing is forbidden. The packing flows in a natural way around these "holes" without subdividing the original surface into smaller packing areas. The main application for this heuristic is the creation of collages of large collections of images where some images are disproportionally larger than the others and positioned in key locations of the original surface. This feature could also be applied in domains where the original surface has defects over, which packing is not to take place.

This article is an extended version of a paper presented at Infocomp 2013 [1]. Here we include new results relating to the performance of the core algorithm, and extend our original results by reporting the execution times of a multithreaded version of the core algorithm running on an Amazon EC2 32-core instance, and packing a billion rectangles.

We are especially interested in avoiding packings that place the larger items concentrated on one side of the surface, and keep covering the remainder of the surface using decreasingly smaller items. These are not aesthetically pleasing packings.

This form of *2D-packing* is a special case of the *2D Orthogonal Packing Problem* (OPP-2), which consists in deciding whether a set of rectangular items can be placed, rotated or not, inside a rectangular surface without overlapping, and such that the uncovered surface area is minimized. In this paper we assume that all dimensions are expressed as integers, and that items cannot be rotated during the packing, which is important if the items are images. 2D-packing problems appear in many areas of manufacturing and technology, including lumber processing, glass cutting, sheet metal cutting, VLSI design, typesetting of newspaper pages, Web-page design or data visualization. Efficient solutions to this problem have direct implications for these industries [2].

Our algorithm packs thousands of items with a competitive efficiency, covering in the high 98 to 99% of the original surface for large collections of items. We provide solutions for several benchmark problems from the literature [3]–[5], and show that our heuristic in some cases generates tighter packings with less wasted space than previously published results, although running slower than the currently fastest solution [6].

To improve the aesthetics of the resulting packing, we use Huang and Chen's [7] surprising quasi-human approach borrowed from masons who pack patios by starting with the corners first, then borders, then inside these limits (similarly to the way one solves a jigsaw puzzle). Our algorithm departs from Huang and Chen's in that it implements a greedy localized best-fit first approach and uses a collection of vertical *lines* containing *segments*. Each vertical segment represents the leftmost side of rectangular area of empty space extending to the rightmost edge of the area to cover. The collection keep the lines ordered by their x-coordinate. All the segments in a line have the same x-coordinate and are ordered by their y-coordinate. Representing empty space in this fashion permits the easy and natural definition of rectangular areas that can be excluded from packing, which in turn offers two distinct advantages: the first is that some rectangular areas can be defined ahead of time as containing images positioned at key locations, and therefore should not be packed over. The second is that subsections of the area to pack can easily be delineated and given to other threads/processes to pack in parallel. Simple scheduling and load-balancing agents are required to allow such processes to exchange items as the packing progresses.

II. The Aesthetics of Photo Collages on a Large Scale

The impetus for this algorithm is to pack a large number of images, typically thousand to millions, in a rectangular surface of a given geometry to form large-scale *collages*. In such applications items are not rotated 90 degrees since they represent images. This type of packing is referred to as *nesting* [8].

Large collages of images are challenging, both because of the packing required, and also because of the required aesthetics. Herr et al. [9] present a large collage of images associated with Wikipedia articles organized as a graph that groups together images based on the similarity of the pages on which they appear. In this context, similar pages are pages that have the same number of shared links. The packing is a simple 2D packing where all images are given the same rectangular frame, and is made to occupy a large circle fit for printing on a poster. In this data visualization, articles are represented by green, blue and yellow disks overlapping each to form clusters around text labels identifying concepts, and all overlay the mosaic of packed images. The aim of this visualization is to present a qualitative aesthetics, rather than to use the packing of variously sized images to convey some quantified relationship. Little effort is made to make the images carry any significant information. The number of images displayed is less than a thousand and already illustrates the challenge of displaying a large collection of images.

On a much larger scale is the packing of the over one billion faces of Facebook users attempted by Natalia Rojas [10]. In this visualization, Rojas presents the visitor of the app.thefacesoffacebook.com page with an approximately 1,200 by 1000 pixel image, where each of the 1.2 million randomly colored pixels represent a Facebook profile image. Packing in this case is straightforward: each rectangle is 1x1 and randomly placed. The visitor of the Web site can zoom in on any of the pixels and is presented by a grid of 100 by 100 pixel images, each image representing an actual profile picture of a Facebook user. The uniform size for the images makes for a trivial packing. It is worth noting that Rojas picked a fairly large 100x100 pixel format for each image, allowing visitors to quickly spot the various faces.

In [11], Wattenberg, Viegas and Hollenbach use *chromograms*, colored fixed-height rectangles aligned in a horizontal bar, to show the edits by users on various Wikipedia pages. While their techique is not a collage, it involves using rectangles of various colors to convey some information pertinent to Wikipedia contents. They cite the large-scale historics containing more than 100,000 events or the irregular structure of the edit logs as significant challenges for making the visualization both effective and aesthetically pleasing.

New developments in display technology that covers walls with video screens and displays billion-pixel digital images have been embraced recently by museum, research centers, and corporate offices. The Cleveland Art Museum [12] is an example where museum goers are presented with a packing of images from the museum collection on a wall of large connected touch-screens. The visitors interact with the display, picking images for additional information. The packing is in bands of same-height images. The result is a pleasing collage of images that are allowed to overlap when the user interacts with them. The Texas Advanced Computing Center's Massive Pixel Environment library [13] allows users to display Processing sketches/citeProcessingOrg over multiple large screen displays. Its use is mostly for visualizing simulation results.

These various efforts are all based in part on the availability of new libraries such as Shiffman's most-pixels-ever Processing package [14], which makes it feasible to display very large images or a large collection of small images, making the problem of packing them efficiently a timely one to address.

Algorithm 1 Simplified Packing Heuristic

```
1:  N = dimension( rects )
2:  VL = {L_0, L_∞}
3:  while not VL.isempty() do
4:    success = false
5:    for all line vl in VL do
6:      list = { } // empty collection
7:      for all segment sl in vl do
8:        rect = rectangle in Rects with height closest to
          but less than sl
9:        if rect not null then
10:         list.add( Pair( rect, sl ) )
11:       end if
12:     end for
13:     if list.isempty() then
14:       continue
15:     end if
16:     sort list in decreasing order of ratio of rect.length to
        sl.length
17:     for all pair in list do
18:       rect, sl = pair.split()
19:       if rect fits in VL then
20:         pack rect at the top of sl
21:         update VL
22:         success = true
23:         break
24:       end if
25:     end for
26:     if success == true then
27:       break
28:     end if
29:   end for
30:   if Rects.isEmpty() then
31:     break
32:   end if
33: end while
```

III. Review of the Literature

Possibly because of its importance in many fabrication processes [2], different forms of 2D-packing have evolved and been studied quite extensively since Garey and Johnson categorized this class of problems as NP-hard [15]. It is hence

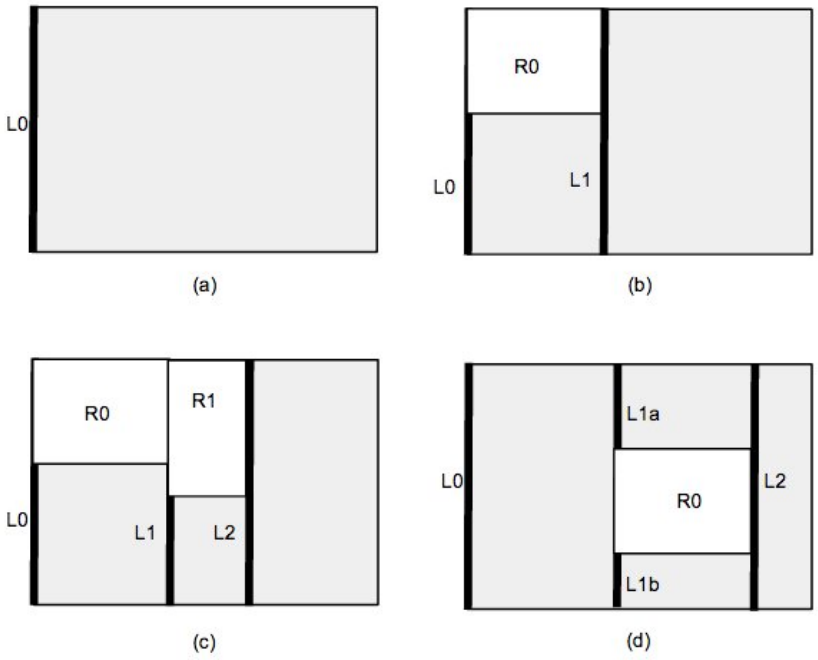


Figure 1. The basic concept of the packing heuristic. (a) The algorithm starts with one vertical line L0 in the left-most position. (b) A rectangle is added, shortening the L0 line and forcing the addition of a second vertical line L1. (c) A second rectangle is added on L1, cutting L1 and forcing the addition of a third line L2. (d) Starting from (a) a rectangle is added in the middle of the available space, creating the addition of a segmented Line L1a, L1b, and a full line L2.

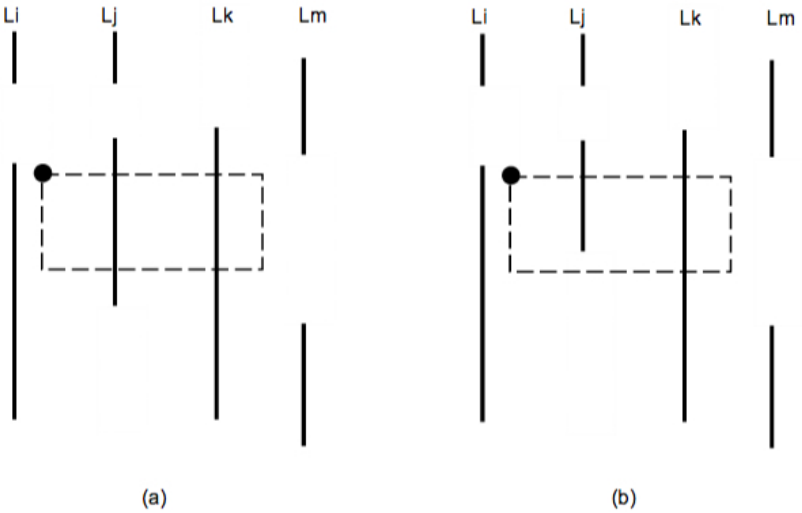


Figure 2. Two examples of potential rectangle placements. In (a) the proposed location for the rectangle (shown in dashed line) is valid and will not intersect with other placed rectangles (not shown) because 1) its horizontal projection on the line L_i directly left of it is fulling included in a segment of L_i, and 2) its intersection with Lines L_j, L_k, and L_m is fully covered by segments of these lines. In (b) the proposed location for the rectangle is not valid, and will result in its overlapping with already placed rectangles since its intersection with Line L_j is not fully included in one of L_j's segments.

a challenge to create a comprehensive review of the literature, as any 2-dimensional arranging of rectangular items in a rectangular surface can be characterized as packing. Burke, Kendall and Whitwell [3] and Verstichel, De Causmaecker, and Vanden Berghe [16] provide among the best encompassing surveys of the literature on 2D-packing and strip-packing research.

While exact solutions are non-polynomial in nature and slow, researchers have achieved optimal solutions for small problem sizes. Baldacci and Boschetti, for example, reports four known approaches to the particular problem of 2D orthogonal non-guillotine cutting problem [17], Beasley's optimal

algorithm [18] probably being the one most often cited. Unfortunately such approaches work well on rather small problem sets. Baldacci and Boschetti, for example, report execution times in the order of tens of milliseconds to tens of seconds for problem sets of size less than 100 on a 2GHz Pentium processor.

Scientists from the theory and operations-research communities have also delved on 2D-packing and have generated close to optimal solutions [19], [20]. The *Bottom-Left* heuristic using rectangles sorted by decreasing width has been used in various situations yielding different asymptotic relative performance guarantees [21]–[24]. Other approaches concentrate on local search methods and lead to good solutions in practice, although computationally expensive. *Genetic algorithms*, *tabu search*, *hill-climbing*, and *simulated annealing* [25], [26] are interesting techniques that have been detailed by Hopper and Turton [2], [4]. These meta-heuristics have heavy computational complexities and have been outperformed recently by simpler best-fit based approaches, including those of Hwang and Chen [5], [7], or Burke, Kendall and Whitwell [3]. Huang and Chen show that placement heuristics such as their *quasi-human* approach inspired by Chinese masons outperforms the meta-heuristics in minimizing uncovered surfaces in many cases, although requiring relatively long execution times. Burke et al. propose a best-fit heuristic that is a close competitor in the minimization of the uncovered surface but with faster execution times.

Probably the fastest algorithm to date is that of Imahori and Yagiura [6], which is based on Burke et al.'s best-fit approach. Their algorithm is very efficient and requires linear space and $O(n \log n)$ time, and solves strip-packing problems where the height of the surface to pack can expand infinitely until all items are packed. They report execution times in the order of 10 seconds for problems of size 2^{20} items. Our serial application is slower, as our timing results show below, but provide a better qualitative aesthetic packing in a fixed size surface with similarly small wasted area. A multithreaded version of our heuristic, however, will pack a million rectangles under 4 seconds, and is presented in details in Section VII. Furthermore, the ability to pack around rectangular areas make for easy parallelization of the algorithm, as we illustrate below. Because the time consuming operation of a collage of image is in the resizing and merging of images on the canvas that vastly surpasses our packing time by several orders of magnitude, the added value of the quality of the aesthetics of the packing makes our algorithm none-the-less attractive compared to the above cited faster contenders.

In the next section we present the algorithm, its basic data structure, and an important proposition that controls the packing and ensures the positioning of items without overlap. We follow with an analysis of the time and space complexities of our algorithm, and show that the algorithm uses linear space and requires at most $O(N^3 \log(N)^2)$ time, although experimental results show closer to quadratic evolution of the execution times. This is due to the fact that the algorithm generally finds a rectangle to pack in the first few steps of the process, and the execution time is proportional mostly to the number of rectangles. Only the last few remaining rectangles take the longest amount of time to pack in the left over space. We compare our algorithm to several test cases taken from the literature in the benchmark section, and close with several examples illustrating how the algorithm operates. We then take the core packing loop and show that by subdividing the area to pack into thin horizontal bands, the speed of the packing can be significantly sped up, making the packing of billions of rectangles possible in the space of minutes. The conclusion section presents possible improvements and future research areas.

IV. The Algorithm

A. Basic Data-Structures

The algorithm is a *greedy*, *localized best-fit* algorithm that finds the best fitting rectangles to pack closest to either one of the left side or top side of the surface. Figure 1 captures the essence of the algorithm and how it progresses.

The algorithm maintains ordered collections of vertical *segments* representing rectangular areas of empty space. Segments are vertical but could also be made horizontal without impeding the operation of the algorithm. These vertical segments can be thought of as the left-most height of a rectangle extending to the right-most edge of the surface to pack. Vertical segments with the same x-coordinate relative to the top-left corner of the surface to cover are kept in vertical *lines*. The algorithm's main data structure is thus a *collection* of lines ordered by their x-coordinates, each line itself a collection of segments, also ordered by their y-coordinates. The collections are selected to allow efficient *exact searching*, *approximate searching* returning the closest item to a given coordinate, *inserting* a new item (line or segment) while maintain the sorted order. *Red-black trees* [27] are good implementations for these collections.

The main property on which the algorithm relies to position a new rectangle on the surface without creating an overlap with already positioned rectangles is expressed by the following proposition:

Proposition 1: A new rectangle can be positioned in the surface such that its top-left corner falls on the point of coordinates (x_{tl}, y_{tl}) and such that it will not intersect with already positioned rectangles if it satisfies two properties relative to the set of vertical lines:

1) Let L_{left} be the vertical line whose x-coordinate x_{left} is the floor of x_{tl}, i.e., the largest x such that $x <= x_{tl}$. In other words, L_{left} is the vertical line the closest to or touching the left side of the rectangle. For the rectangle to have a chance to fit at its present location, the horizontal projection of the rectangle on L_{left} must intersect with one of its segments that completely contains this projection.
2) The horizontal projection of the rectangle on *any* vertical line that intersects it must also be completely included in a segment of this line.

Figure 2 illustrates this proposition.

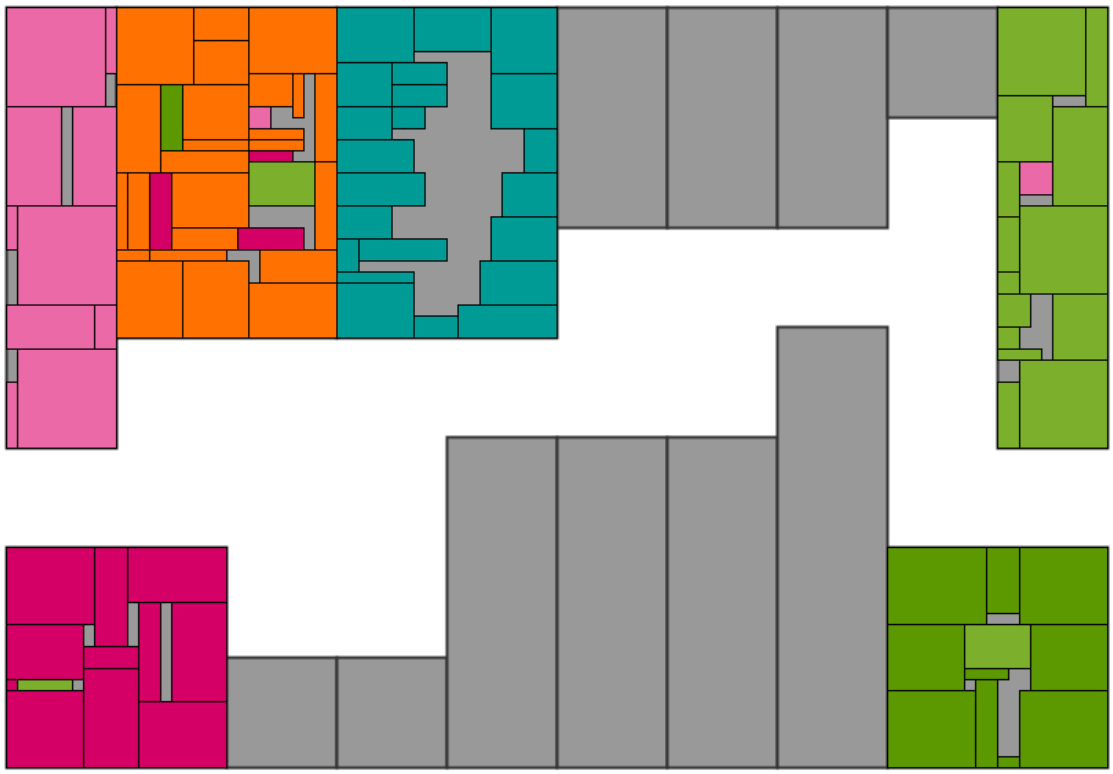


Figure 3. A solution generated by our algorithm for the packing of 100 items in 16 objects as proposed by Hopper as the "M1a" case.

B. Basic Operation

The algorithm starts with two vertical lines, L_0 and L_∞. The first line originates at the top-left corner of the surface to cover, and contains a single segment whose length defines the full *height* of the surface to pack. L_∞ is a vertical line located at an x-coordinate equal to the width of the surface to pack. L_∞ contains no segments. It identifies the end of the area to pack. Any rectangle that extends past the end of the area to cover will cross L_∞, and because this one does not contain segment, the second part of the proposition above will reject the rectangle.

To simplify the description of the algorithm, we use the generic term *line* to refer to lines and segments. The algorithm packs from left to right, and favors top rather than down locations. Starting with the vertical line L_0 it finds the item R_0 with the largest height less than L_0. If several items have identical largest height, the algorithm picks the one with the largest perimeter and tests whether it can be positioned without overlapping any other already placed items. The algorithm tries three different locations: at the top of L_0, at the bottom of L_0, or at the centre of L_0. The item is positioned at the first location that offers no overlap, otherwise the next best-fitting item is tested, and so on.

The positioning of R_0 shortens L_0, as shown in Figure 1(b). A new line L_1 is added to the right of R_0 to indicate a new band of space to its right that is free for packing.

The goal is to place all larger items first and automatically the smaller ones find places in between the larger ones.

In Figure 1(c), the algorithm finds R_1 as the rectangle whose width is the largest less than L_1 and positions it against the left most part of L_1. Adding R_1 shortens L_1, indicating that all the space right of the now shorter L_1 is free for packing. Again, a new line L_2 is added to delineate a band of empty space to the right of R_1.

We implement the data-structures for the lines as trees sorted on the line position relative to the top-left corner of the initial surface, so that a line or a group of lines perpendicular to particular length along the width or height of the original surface can be quickly found.

Note that in our context these line-based data-structures allow for the easy random positioning of rectangles in the surface before the packing starts, as illustrated in Figure 1(d) where a rectangle R_0 is placed first in the middle of the surface before the packing starts.

C. The Code and its Time and Space Complexities

We now proceed to evaluate the time complexity of our heuristic, whose algorithmic description is given in Algorithm 1. In it, N is the number of items to pack, $rects$ is the list of items to pack, VL the collection of vertical lines, and vl one such individual line.

Since N is the original number of items to pack, then clearly the size of VL is $O(N)$. Given a line vl of VL, we argue that the average number of segments it contains (exemplified by

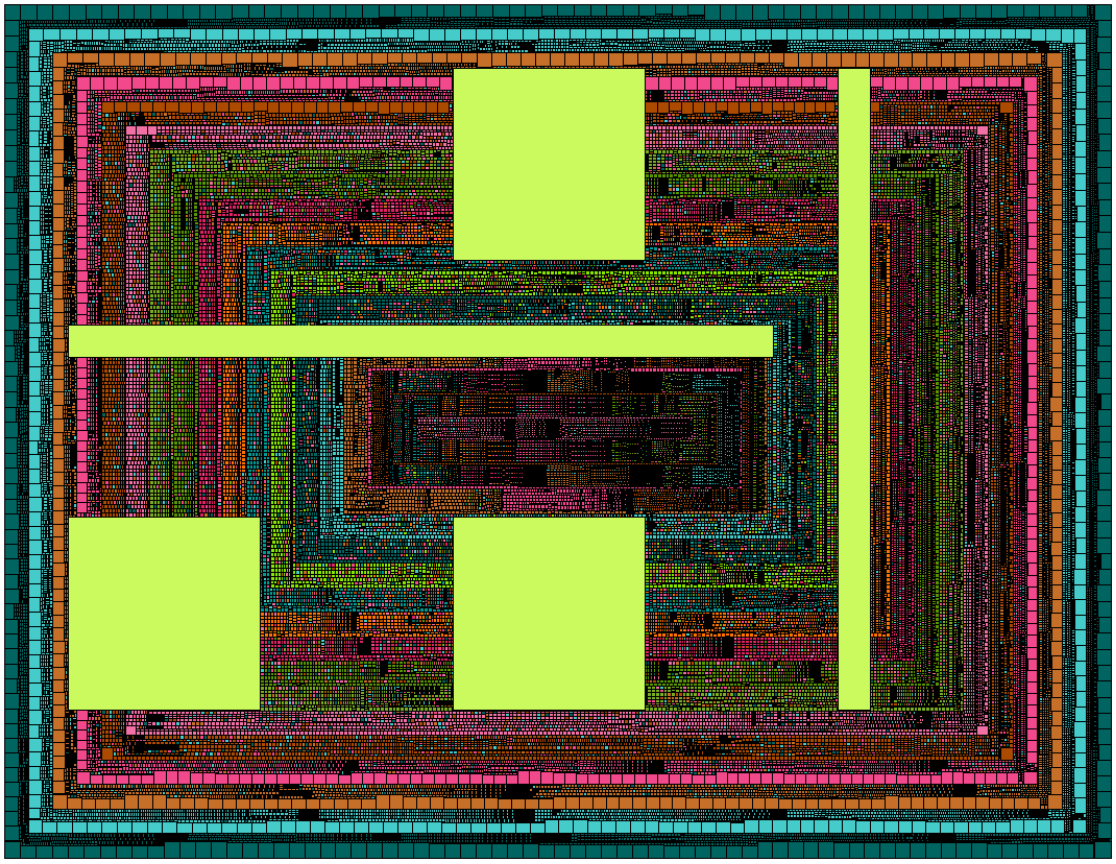


Figure 4. The packing of 97,272 randomly generated items in a a rectangular surface. The application is multithreaded, each thread associated with a rectangular border. 5 large lime-green rectangles with different geometries are placed in various locations before the computation starts.

$L1a$ and $L1b$ in Figure 1) is $O(N)$. The goal of the Loop starting at Line 3 is to pack all rectangles, and it will repeat N times, hence $O(N)$. The combined time complexity of the loops at Lines 5 and 7 is $O(N)$ because they touch at most all segments in all the lines, which is bounded by $O(N)$. The time complexity of Line 16 is clearly $O(N \log N)$, although on the average the number of pairs to sort will be $O(\sqrt{N})$ rather than $O(N \log N)$. The loop starting at Line 17 processes at most $O(N)$ pairs, and for each rectangle in it, must compare it to at most $O(N)$ line vl. So it contributes $O(N^2)$, which overpowers the sorting of the list. Therefore, the combined complexity of the whole loop starting on Line 3 is $O(N^3)$.

Empirically, however, the algorithm evolves in quasi quadratic fashion as illustrated in Figure 6, where various selections of rectangles with randomly set dimension are packed in a rectangular surface that is selected ahead of time to be of a given aspect ratio, and whose total area is 1% larger than the sum of all the items to pack. We found this approach the best for packing quickly. The dimensions of the randomly-sized rectangles for all the experiments reported here are computed by the following equations:

$$\begin{aligned} width &= max(20, RandInt(500)) \\ height &= max(20, RandInt(500)) \end{aligned}$$

where $RandInt()$ returns a random integer between 0 and 500, excluded. This translate in 230,400 uniformly distributed possible geometries. Remember that we do not allow for rectangles to rotate, so all geometries are unique.

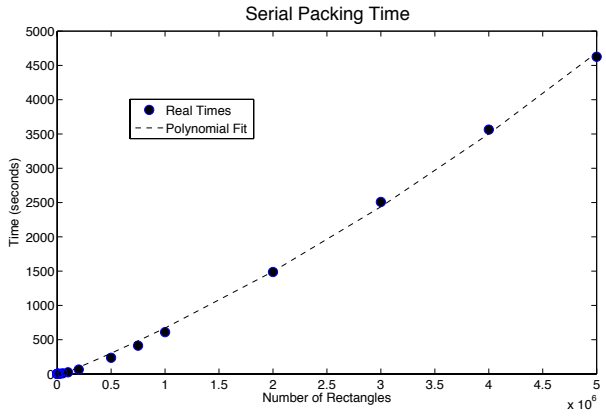


Figure 6. Running times and regression fits for packings of 100 to 5,000,000 random rectangles on one core of a 3.5 GHz 64-bit AMD 8-core processor.

Note that the times reported are user times, and that

Figure 5. The packing of 2,200 photos of various sizes and aspect-ratios, as many are cropped for artistic quality. The size of the photos is randomly picked by the algorithm. All the photos belong to this author.

only one core of the processor is used, corresponding to a totally serial execution time. A second-degree polynomial fit of the measured times is shown. The fit has equation $y = 5.985\ 10^{-11}\ x^2 + 6.447\ 10^{-4}\ x - 3.514\ 10^{1}$, with a correlation coefficient r^2 of 0.99898, and a standard error of 49.263.

The space complexity is clearly $O(N^2)$, since the addition of a rectangle to a group of R already packed rectangles will cut at most R lines and introduce at most R new segments. The cumulative effect results in a quadratic variation of the number of segments.

D. Algorithmic Features

Our heuristic sports one feature that is key for our image-collage application: Rectangular areas in which packing is forbidden can easily be identified inside the main surface to be packed, either statically before starting the packing or even dynamically during run time. We refer to these areas as *empty zones*. This feature offers the user the option of positioning interesting images at key positions on the surface to be packed ahead of time. In other domains of application these could be areas with defects. Additionally, it allows parallel packing approaches where rectangular empty zones can be given out to new processes to pack in parallel, possibly shortening the execution time.

V. Benchmarks

A set of benchmark cases used frequently in the literature are those of Hopper and Turton [4], and of Burke, Kendall and Whitwell [3]. For the sake of brevity we select a sample of representative cases and run our heuristic on each one. The computer used to run the test is one core of a 64-bit Ubuntu machine driven by a 3.5GHz AMD 8-core processor, with 16GB of ram. The heuristic is coded in Java. Note that all published results do not always provide a derivation of the time complexity of the heuristic presented, and the goodness of the algorithm is measured by its execution time on various benchmark cases. Unfortunately, all experiments are run are on different types of computers, ranging from ageing memory-limited laptops to supped up desktops, all with different processor speed and memory capacities. To provide a more objective comparison, we make the following assumptions: *a*) all results reported in the literature corresponded to compiled applications that are all memory residents, *b*) they are the only workload running on the system, *c*) MIPS are linearly related to CPU frequency, and thus we scale the execution times of already published data reported by the ratio of their operating CPU frequencies to that of our processor (3.5GHz).

We follow the same procedure used by the researchers whose algorithms we compare ours to, and we run our application multiple times (in our case 30 times) on the same

TABLE I. PERFORMANCE COMPARISON TABLE

			Burke		GRASP		3-way		DT	
Case	Number items	optimal height	diff.	time (s)	diff.	time (s)	diff.	time (s)	diff.	time (s)
N1	10	40	0	∼14.571	0	∼34.286	5	<0.009	0	0.05
N2	20	50	0	∼14.571	0	∼34.286	3	<0.009	6	<0.01
N3	30	50	1	∼14.571	1	∼34.286	4	<0.009	10	<0.01
N4	40	80	2	∼14.571	1	∼34.286	6	<0.009	49	<0.01
N5	50	100	3	∼14.571	2	∼34.286	4	<0.009	5	0.03
N6	60	100	2	∼14.571	1	∼34.286	2	<0.009	22	0.01
N7	70	100	4	∼14.571	1	∼34.286	7	<0.009	14	<0.01
N8	80	80	2	∼14.571	1	∼34.286	3	<0.009	23	<0.01
N9	100	150	2	∼14.571	1	∼34.286	13	<0.009	5	0.04
N10	200	150	2	∼14.571	1	∼34.286	2	0.01	10	0.03
N11	300	150	3	∼14.571	1	∼34.286	2	0.01	2	0.49
N12	500	300	6	∼14.571	3	∼34.286	5	0.02	7	0.07
N13	3152	960	4	∼14.571	3	∼34.286	4	0.20	5	0.927
C7-P1	196	240	4	∼14.571	4	∼34.286	6	<0.009	17	0.02
C7-P2	197	240	4	∼14.571	3	∼34.286	4	<0.009	41	0.02
C7-P3	196	240	5	∼14.571	3	∼34.286	5	<0.009	24	0.01

problem set and keep the best result.

Table I shows the scaled execution times of the various heuristics for problem sets taken from the literature. Column 1 identifies the various cases from Burke et al. [3], with the number of items packed in Column 2, and the optimal height of the packing in Column 3. The difference between the resulting height of algorithm's packing and optimal along with the execution time in seconds are shown for each of 4 algorithms, in Columns 4-5, 6-7, 8-9, and 10-11. Our heuristic's data covers the last two columns. The times are those reported in the literature multiplied by a scaling factor equal to the 3.5GHz/*speed of processor*, where the processor is the one used by the researchers. For the Burke column, the speed of the processor is 850MHz. For the GRASP column, 2GHz, and for the 3-way column, 3GHz.

We observe that, as previously discovered [6] our packing efficiency improves as the number of items gets larger (in the thousand of items), which is the size of our domain of interest. The execution times of our heuristic are faster than those of Burke's best-fit, or of GRASP, and at most five times slower than the fast running 3-way best-fit of Imahori and Yagiur [6]. This difference might be attributed to either the choice of language used to code the algorithm, Java in our case, versus C for theirs.

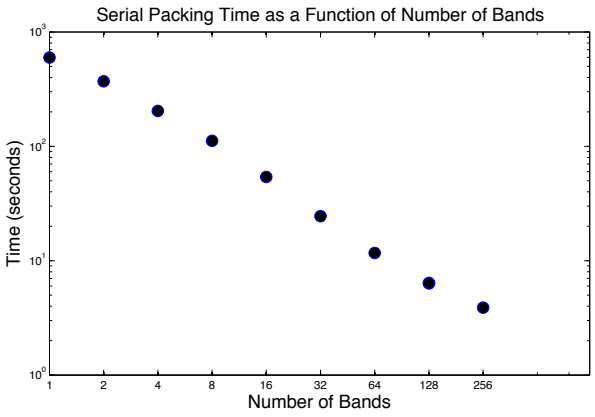


Figure 7. Running times for packing 1,000,000 rectangles with 1 to 256 bands dividing the packing area. User times measured on one core of a 3.5 GHz 64-bit AMD 8-core processor.

VI. PERFORMANCE FOR LARGE SCALE PACKING

A. Subdividing the space into horizontal bands

In this section, we report on experiments conducted on a modified version of our heuristic where we skip the packing of the corners and borders first, and divide the rectangular packing area in small non overlapping horizontal bands of the same length as the large area to pack. We have found that limiting the packing to smaller bands significantly decreases the packing time by reducing the size of the Red-Black tree data-structures holding all the lines and segments. Typically, the area to pack is divided in 256, 512, 1024 or more horizontal bands as long as all the rectangles can be packed in the given area (small-height bands decrease the packing efficiency). For packing 1,000,000 rectangles without dividing the space into band requires 603 seconds on one core of our 3.5 GHz processor. Dividing the space into 256 bands and packing each band one after the other, serially, on one core, brings the user execution time to 4.03 seconds and maintains a packing efficiency greater than 99%. Figure 7 shows the user-times in seconds for packing 1,000,000 randomly selected rectangles when the area to pack is divided in 1, 2, 4, 8, ... 256 bands. Note that when we increase the number of bands by two, the execution time is almost halved. Therefore, applications that require high-

speed 2D-packing should organize the area to pack in as many narrow horizontal bands as possible while maintaining a target efficiency.

In the next section, we show several packings generated by our heuristic.

VII. Packing Examples

In this section, we provide several examples of packing under various conditions and constraints, some of them taken from the literature.

In Figure 3, we apply our heuristic to Hopper's *M1a case* [2] where 100 items must be packed into 16 different objects. Our algorithm also packs the objects, although this is not a requirement of the test. In this experiment, our heuristic is multithreaded and several threads pack the different objects. A scheduler simply distributes the objects to separate threads, picking the largest object first and assigning it to a new thread implementing our packing heuristic. Then the scheduler picks the next largest object (in terms of its area) and assigns it to a new thread, and so on. The earliest starting threads are given a random sample of the items to pack. Threads that start last have to wait until earlier threads finish packing and return items that could not be packed. This automatically packs objects in such a way that as few objects as possible are packed, and some left empty, which may be desirable.

In Figure 4, the original surface is divided at run time into smaller surfaces, or *borders* one inside the other as the packing progresses, and individual threads are running the packing on individual borders. Here again the threads are given random samples of the original population of items and a load balancing scheme allows for the exchange of items between threads. This is represented by items with different colors. For example, the items associated with the first thread are all dark green, and some can be found in the light green, orange or pink borders as they are rejected by the first thread once it has packed the dark green band. Note that the utilization of the surface is 99.30%.

In Figure 4, we have placed five large items (yellow-green rectangles) on the surface before launching the packing algorithm. Notice how the heuristic naturally packs around these areas. Note also that as in Figures 3 and 4, we follow Huang and Chen's quasi-human approach [7] and pack corners and borders first before proceeding with the inside areas. Note that this modification of the algorithm fits completely with the natural properties of the heuristic, and enhances the visual aspect of the final packing.

VIII. 2D-Packing a Billion Random Rectangles

The ability to divide the rectangular area to pack into thin horizontal bands leads us to consider the challenge of packing a billion rectangles with random dimensions. The main obstacle to solve this problem is not in a long execution time, but rests in the ability to keep a billion objects in random-access memory (RAM). Given that a rectangle is defined by a geometry requiring a minimum of two *longs* for the coordinates, and two *ints* for the dimensions, at least 24 to 32 Gigabytes of storage are required to store a billion such objects, depending on whether the application runs on a 32 or 64-bit system. If packing speed is of the essence, then the objects must reside in memory.

To keep the computation CPU-bound as well as RAM-bound, and measure the best possible packing performance, we multi-thread our packing heuristic and run it on an Amazon *c3.8xlarge* instance, which, at the time of this writing, sports the following characteristics:

- **CPU Architecture:** 64 bits.
- **Cores:** 32 hyperthreaded 2.8 GHz Intel Xeon E5-2680v2 cores (Ivy Bridge).
- **Performance:** Combined CPU speed equivalent to 108 *m1.small* Amazon instances. An m1.small typically has the same performance as a 1.0-1.2 GHz 2007 Intell Opteron or 2007 Xeon processor.
- **RAM:** 60 Gigabytes.
- **Disk Storage:** two 320-GB Solid-Stated Device disks.
- **Network Speed:** 10 GBits.

Note that the c3.8xlarge processor frequency can be turbo-boosted to 3.6 GHz if enough thermal room is available.

A. Multithreaded Algorithm

We adopt a simple scheduling and load-balancing of the different threads. Assuming that there are N rectangles to pack and that the area to cover is divided into B bands, we assign each one to a single thread, and we run in parallel the packing of the first $B-1$ bands. We perform a *join* operation on all the running threads. We allocate $alpha\ N/B$ rectangles ($alpha > 1$) to each thread so that the packing can benefit from a greater collection of rectangles than what can fit during the packing. We have found that $alpha$ = 1.01 yields good packing efficiencies. When the first $B-1$ bands have been packed, the threads return the rectangles that could not be packed and these are returned to the pool of unpacked items. This collection, along with all remaining unallocated rectangles, is given to a final thread that packs the last band.

The algorithm is detailed in Algorithm 2 below.

B. Execution Time

Dividing the total area to be covered in 4096 bands, and launching the multithreaded algorithm to pack a billion rectangles on an Amazon c3-8xlarge instance takes **8 minutes and 56 seconds of user time**. For comparison, dividing the space in 2048 bands, instead, results in a user time of 11 minutes and 52 seconds. Packing 1 million rectangles in 256 bands now takes only 3.2 seconds. We keep the RAM usage low by observing that since we need randomly sized rectangles, they do not need to be stored ahead of time, but instead they can be generated on the fly as they are passed to each thread. Only the rectangles that have been packed and assigned coordinates relative to each band's top-left corner are kept.

Algorithm 2 Multithreaded Packing Scheduling and Load-Balancing

```
1: rects = list of all rectangles to pack
2: N = dimension( rects )
3: B = dimension( bands )
4: noBandsPacked = 0
5: α = 1.01
6: while noBandsPacked < B-1 do
7:    create Thread t_i for Band b_i
8:    allocate (N/B * α) rectangles to t_i
9:    rects.remove( all allocated rectangles )
10:   start t_i
11:   noBandsPacked ← noBandsPacked + 1
12: end while
13: join on all threads t_i
14: for all joined thread t_i do
15:   rects.append( rectangles unpacked by t_i )
16: end for
17: create Thread t_{B-1}
18: allocate rects to t_{B-1}
19: start t_{B-1}
20: join on t_{B-1}
```

C. New Application Domains

The ability to quickly pack large collections of rectangles opens applications based on 2D-packing to the realm of *real-time* and *interactive* implementations.

It is now conceivable that a user may interact with a large display, say in a museum showing its entire collection as a packing of images, pick one or a group of images and have the application automatically remove, reposition, or resize them, quickly refreshing the space around or underneath them with a new 2D-packing. This new feature requires the implementation of fast data structures for quickly locating packed rectangles inside an area or overlapping a given point. *R-trees* [28], which maintain groupings of objects in space by geographical closeness, offer interesting possibilities, and are the subject of ongoing research.

IX. Conclusions

We have presented a new heuristic for packing or nesting two-dimensional images in a rectangular surface. The heuristic packs the items by creating a collection of segments that are maintained in two data structures, one for horizontal segments, and one for vertical segments. The segments represent the leftmost and topmost side of rectangular surfaces that extend to the edges of the original surface to pack. These data structures permit to test quickly whether a new item can be positioned in the surface without overlapping a previously placed item.

Our packing heuristic does not rotate items, but none-the-less compares favourably with other heuristics published in the literature that solve 2D-strip packing with rotation of items allowed. If rotation of items is required, a possible modification of the algorithm is to give a packing thread two versions of the same rectangle, one the 90-degree rotated version of the other, both linked to each other. Whenever one of the versions is packed, the algorithm quickly searches its list of unpacked items and removes the item linked to the one just packed.

The data structure used to maintain the empty areas lends itself well to positioning items in key places ahead of time, or in subdividing the original surface into multiple holes that can be either left empty, reserved for large size items, or assigned to separate processes that will pack in parallel. Such holes may contain defects (for example in a sheet of metal, or glass) that need to be avoided by the packing process.

It is possible to significantly speed the core packing algorithm up by slicing the area to cover into individual thin horizontal rectangular bands. This limits the amount of searching for the best fitting place for the next rectangle, and the execution time drops inversely proportionally with the width of the bands.

If the slicing of the packing area creates undesirable horizontal dividing lines on which rectangles align themselves during the packing, one can easily pre-pack small rectangles over the boundaries of bands, hiding in effect the dividing line.

Because our domain of application is that of image collages, we have found that the the quasi-human approach of Huang and Chen, along with subdividing the surface into nested rectangular area significantly improves the aesthetic quality of the packing compared to most heuristic that privilege one side or corner and put all largest items there and finish packing with the smaller items at the opposite end.

References

[1] D. Thiebaut, "2d-packing images on a large scale," in Proceedings of INFOCOMP 2013, Nov 17-22 2013, pp. 19–26.

[2] E. Hopper, "Two-dimensional packing utilising evolutionary algorithms and other meta-heuristic methods," Ph.D. dissertation, Cardiff University, United Kingdom, 2000.

[3] E. K. Burke, G. Kendall, and G. Whitwell, "A new placement heuristic for the orthogonal stock-cutting problem," Oper. Res., vol. 52, no. 4, Aug. 2004, pp. 655–671.

[4] E. Hopper and B. C. H. Turton, "An empirical investigation of meta heuristic and heuristic algorithms for a 2d packing problem," European Journal of Operational Research, vol. 1, no. 128, 2000, pp. 34–57.

[5] W. Huang, D. Chen, and R. Xu, "A new heuristic algorithm for rectangle packing," Computers and Operations Research, vol. 34, no. 11, November 2007, pp. 3270–3280.

[6] S. Imahori and M. Yagiur, "The best-fit heuristic for the rectangular strip packing problem: An efficient implementation and the worst-case approximation ratio," Comput. Oper. Res., vol. 37, no. 2, Feb. 2010, pp. 325–333.

[7] W. Huang and D. Chen. Simulated annealing. Accessed 05/12/2014. [Online]. Available: http://cdn.intechopen.com/pdfs/4629/InTech-An_efficient_quasi_human_heuristic_algorithm_for_solving_the_rectangle_packing_problem.pdf [retrieved: July, 2008]

[8] R. D. Dietrich and S. J. Yakowitz, "A rule-based approach to the trim-loss problem," International Journal of Production Research, vol. 29, 1991, pp. 401–415.

[9] B. W. Herr, T. Holloway, E. F. Hardy, K. W. Boyack, and K. Brner, "Science-related wikipedian activity," 3rd Iteration (2007): The Power of Forecasts: Places and Spaces: Mapping Science, 2007.

[10] N. Rojas. The faces of facebook. Accessed 5/12/2014. [Online]. Available: http://app.thefacesoffacebook.com/ (2013)

[11] M. Wattenberg, F. Vigas, and K. Hollenbach, "Visualizing activity on wikipedia with chromograms," in Human-Computer Interaction INTERACT 2007, ser. Lecture Notes in Computer Science, C. Baranauskas, P. Palanque, J. Abascal, and S. Barbosa, Eds. Springer Berlin Heidelberg, 2007, vol. 4663, pp. 272–287. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74800-7_23

[12] Collection wall. Accessed 5/12/2014. [Online]. Available: http://www.clevelandart.org/gallery-one/collection-wall (2013)

[13] B. Westing, H. Nieto, H. Nieto, and K. Gaither, "Massivepixelenvironment: A tool for rapid development with distributed displays," in CHI 2013, 2013.

[14] D. Shiffman. Most pixels ever. [Online]. Available: https://github.com/shiffman/Most-Pixels-Ever-Processing/wiki (2008)

[15] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman and Company, 1979.

[16] J. Verstichel, P. D. Causmaecker, and G. V. Berghe, "An improved best fit heuristic for the orthogonal strip packing problem," International Transactions in Operational Research, no. 20, June 2013, pp. 711–730.

[17] R. Baldacci and M. A. Boschetti, "A cutting-plane approach for the two-dimensional orthogonal non-guillotine cutting problem," European Journal of Operational Research, vol. 183, no. 3, 2007, pp. 1136–1149, accessed 5/12/2014.

[18] J. Beasley, "An exact two-dimensional non-guillotine cutting tree search procedure," Operations Research, vol. 33, no. 1, January 1985, pp. 49–64.

[19] E. G. Coffman, M. R. Gazey, and D. S. Johnson, Approximation algorithms for bin-packing an updated survey. Springer-Verlag, 1984.

[20] H. Dyckhoff, "Typology of cutting and packing problems," European Journal of Operational Research, vol. 44, 1990, pp. 145–159.

[21] B. S. Baker, E. G. C. Jr., and R. L. Rivest, "Orthogonal packings in two dimensions," SIAM Journal on Computing, vol. 9, 1980, pp. 846–855.

[22] B. S. Baker, D. J. Brown, and H. P. Katseff, "A 5/4 algorithm for two-dimensional packing," Journal of Algorithms, vol. 2, 1981, pp. 348–368.

[23] D. Sleator, "A 2.5 times optimal algorithm for packing in two dimensions," Information Processing Letter, vol. 10, 1980, pp. 37–40.

[24] C. Kenyon and E. Remilia, "Approximate strip-packing," in Proceedings of the 37th Annual Symposium on Foundations of Computer Science, 1996, pp. 31–35.

[25] D. Liu and H. Teng, "An improved bl-algorithm for genetic algorithm of the orthogonal packing of rectangles," European Journal of Operational Research, vol. 112, no. 2, January 1999, pp. 413–420.

[26] T. W. Leung, C. K. Chan, and M. Troutt, "Application of a mixed simulated annealing genetic algorithm heuristic for the two-dimensional orthogonal packing problem," European Journal of Operational Research, vol. 145, no. 3, March 2003, pp. 530–542.

[27] L. J. Guibas and R. Sedgewick, "A dichromatic framework for balanced trees," Foundations of Computer Science, IEEE Annual Symposium on, vol. 0, 1978, pp. 8–21, accessed 5/12/2014.

[28] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '84. New York, NY, USA: ACM, 1984, pp. 47–57.

State Space Reconstruction in UPPAAL: An Algorithm and its Proof

Jonas Rinast, Sibylle Schupp
Institute for Software Systems
Hamburg University of Technology, Hamburg, Germany
Email: {jonas.rinast,schupp}@tuhh.de

Dieter Gollmann
Security in Distributed Applications
Hamburg University of Technology, Hamburg, Germany
Email: diego@tuhh.de

***Abstract*—Efficient state space reconstruction is necessary to carry out on-line model checking, a variant of model checking where parameters of a model are continually adjusted to remedy possible modeling faults. On-line model checking is, for example, useful in the medical domain where models of patient states and reactions are always inaccurate, but ensuring patient safety with model checking techniques is still desirable. In this paper, we propose a transformation reduction method based on use-definition chains to efficiently carry out the required state space reconstruction. We provide a formal specification of the general algorithm and proofs for its correctness. For evaluation we applied our reduction approach to the state space of the model checker UPPAAL. The experiments resulted in a reduction of the number of transformations necessary to reach a certain state by 42% on average.**

***Keywords*—State Space Reconstruction; On-line Model Checking; UPPAAL; Reference Counting; Use-Definition Chain.**

I. Introduction

Medical treatment facilities have grown to rely significantly on medical devices for monitoring and treatment. Most devices are still operated manually today and need to be configured, maintained, and supervised by medical staff. Recently, closed-loop monitoring and treatment of patients became a research topic as experience shows that human errors are prevalent. Patient-in-the-loop systems try to autonomously assess the patient's state using a monitoring device and if necessary treat the patient automatically, e.g., via a remote infusion pump. Clearly, such a system must be shown to cause no harm to the patient. Safety must be ensured to prevent harm from the patient not only during normal operation but also in case emergency situations arise.

Model checking is a well developed technique for verifying that a system model conforms to its specification and thus may be applied to show the safety of such systems. However, to make meaningful conclusions about the system's behavior it is necessary to have detailed and accurate models of the individual components of the system. In the medical domain, model checking is therefore severely hampered if the patient needs to be modeled accurately, e.g., to make estimates on a drug concentration in the patient. Generally, a patient model is likely to be inaccurate as the physiology of the human body is complex and varies between individuals, e.g., blood oxygen and heart rate depend on the patient's condition. A generalized model will always miss individual characteristics. Patient-in-the-loop systems thus could be proved safe with such models but might still put patients at risk.

On-line model checking is a recent model-checking variant that relaxes the need for models to be accurate far into the future. On-line model checking provides safety assurances for short time frames only and renews these assurances continually during operation. Appropriate models for the system thus only need to be correct for the short time frame they are used in. The renewal of safety assurances then is carried out on models adapted to the current system state to ensure the system's safety for the next time window. This on-line approach thus allows safety assessment at all times and provides means to react before safety violations occur.

Because parts of the previous state should be maintained and cross-correlations between state variables could be destroyed inadvertently a model adaptation step may first create an initialization sequence that recreates the previous model state before adjusting single values. The reconstruction is necessary to allow the simulation of the model to continue from the state it was interrupted in. This paper presents an automated state reconstruction approach for the Uppsala and Aalborg model checker (UPPAAL) that eliminates the need for custom reconstruction procedures for every application. The developed reconstruction method serves as a base for an on-line model checking interface with UPPAAL as the underlying verification engine.

Naively, the state space can be reconstructed by executing the same transition sequence that was used to create the state in the beginning. However, if the simulation has already run a significant time the executed transition sequence is likely to be long and only continues to grow over time. A more direct way to the desired state space is needed to keep the reconstruction process fast and on-line model checking feasible. For our reconstruction approach we adopted use-definition chains, a data flow analysis capturing relations of data sources and sinks, to eliminate transformations that have no effect on the final state space. Such transformations occur when their results are overwritten before they are read. Our reconstruction method has been applied to seven different test models. The method always reconstructed the original state space while yielding a reduction of the executed transformations in the range from 23% to 84%.

To summarize, in this paper we, first, contribute to the field of on-line model checking by presenting a transformation reduction algorithm together with its proof that may be used to reconstruct a particular model-checking state space, and, second, also contribute to the applicability of the UPPAAL model checker by providing a specialization of the algorithm

together with its implementation evaluation.

The rest of the paper is organized as follows: Section II gives an overview on related literature. Section III briefly introduces model checking, on-line model checking, and the model checker UPPAAL. Section IV provides necessary information on UPPAAL's state space and its transformations. Section V then explains our reconstruction approach with a focus on the reduction algorithm using use-definition chains. Section VI presents our evaluation results and, lastly, Section VII summarizes the paper and suggests further research.

This paper is an extended version of the paper published at VALID 2013 [1]. In contrast to the explanation of the reduction approach by way of a running example, this paper provides detailed information on the reduction by formalizing the approach and providing correctness proofs for the algorithm.

II. Related Work

The on-line model checking approach our reconstruction method is complementing and thus is closest to has recently been proposed by Li et al. [2][3]. They employ a hybrid automata model to ensure correct usage of a laser scalpel during laser tracheotomy to prevent burns to the patient. Yet, the necessary model initialization and reconstruction step is a custom solution and is not presented in detail. To our knowledge there are no other reconstruction methods for a particular UPPAAL state in the context of on-line model checking. However, the UPPAAL variant UPPAAL Tron, an on-line testing tool that can generate and execute test cases on-the-fly based on a timed automata system model, has been developed [4]. While the tool focus lies on input/output testing using a static system model the fact that the underlying model is an UPPAAL model means that our reconstruction approach might be beneficial for tests when the system model is inaccurate or still needs to be developed. Other related work falls in two categories: different ways to use or implement on-line model checking, and different ways to optimize state space exploration and representation in model checkers.

Qi et al. propose an on-line model checking approach to evaluate safety and liveness properties in C/C++ web service systems [5]. Their focus lies on consistency checks for distributed states to debug a system from known source code. Reconstruction is not an issue because the source code is static during execution. Easwaran et al. use a control-theoretic approach to the general runtime verification problem [6]. They introduce a steering component featuring a model to predict execution traces. Their approach uses a fixed prediction model while our reconstruction is for adapting inaccurate models. Sauter et al. address the prediction of system properties using previously gathered time series of measurements, e.g., taken by sensors [7]. They propose a split into an on-line and an off-line computation and to precompute expensive parts of the prediction step to reduce on-line work load. While their scenario of adapting using sensor measurements is applicable to our medical scenario with inaccurate patient models they focus on the verification load problem while we address model inaccuracy. Harel et al. propose usage of model checking during the behavior and requirement specification step during development. Instead of interactively guiding the system to derive requirements a model checker executes the model and generally finds more adequate requirements. While their approach employs on-line model checking their goal thus lies on early requirement development. In contrast, our approach is useful in adaptation of deployed systems to ensure safety. Arney et al. present a recent patient-in-the-loop case study for automatic monitoring and treatment where UPPAAL and Simulink models were developed to verify safety questions beforehand [8]. They monitor heart rate and blood oxygen levels of the patient and automatically control drug infusion via a remote pump. On-line model-checking could benefit this scenario as currently a generalized patient model is employed and drug absorption rates may vary per patient.

Alur and Dill introduced timed automata and the underlying theory in 1994 [9] and Yi et al. developed the first implementation of the model-checker UPPAAL shortly after [10]. Many improvements have been made to the model-checking approach over the years. Larsen et al. proposed symbolic and compositional approaches to reduce the state-space explosion problem [11]. Partial order reduction on the state space was employed by Bengtsson [12]. Larsen et al. reduced memory usage on-the-fly using an algorithm that exploits the control structure of models [13][14]. Further memory reductions were achieved by Bengtsson et al. with efficient state inclusion checks and compressed state-space representations [15]. Behrmann et al. provide an overview on current functionality and the usage of UPPAAL [16]. They also provide a more detailed presentation of UPPAAL's internal representations [17]. For a summary on timed automata, the semantics, used algorithms, data structures, and tools see [18]. Bengtsson's PhD thesis provides more in-detail information on difference bounded matrices [19].

III. On-Line Model Checking

This section introduces model checking and its on-line variant, on-line model checking. The technique is shown by way of example using the model checker UPPAAL; for a formal specification of UPPAAL see [18].

Generally, model checking explores the state space of a given system model in a symbolic fashion to check whether the state space satisfies certain properties. Such properties are mostly derived from a requirement specification for the system, e.g., one could check whether or not a certain system state is actually reachable. The modeling and property languages vary greatly depending on the model-checking tool. Tools for various programming languages coexist with dedicated tools that support their own modeling language. Dedicated tools often use finite state automata as a base formalism for their models. UPPAAL is such a well-established, dedicated model checking tool, jointly developed by Uppsala University, Sweden, and Aalborg University, Denmark [13][16]. It is based on the formalism of timed automata, an extension of finite state automata with clock variables to allow modeling of time constraints. A finite state automaton defines a transition system by defining locations and edges that connect these locations. Edges are fired to execute a transition from one location to another. The system state in this case is the current location of the automaton and the possible valuations of the clock variables.

Figure 1 shows the example model that will be used to demonstrate the proposed state space reconstruction method.

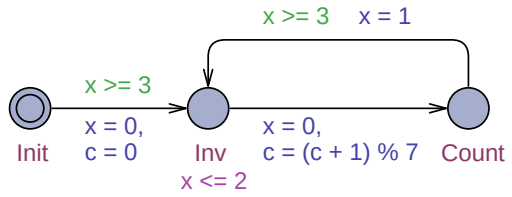


Figure 1. UPPAAL Model Example

The model consists of three locations, *Init*, *Inv*, and *Count*, where *Init* is the initial location indicated by the double circle. The model uses two variables: x, a clock variable, and c, a bounded integer variable. Clock variables are special variables that synchronously advance indefinitely unless they are bounded by one or more invariants on the current locations. The location *Inv* has such an invariant, `x <= 2`, to bound the clock x, thus the value of x in *Inv* can be any value between its value when it entered the location and 2. The model has a single transition from the initial location to *Inv*. This transition is annotated with a guard, `x >= 3`, and an update, `x = 0, c = 0`. Guards are used to enable and disable edges depending on the current state. Here, the clock x needs to be greater or equal to 3 for the edge to be enabled. Only then can it be fired and a transition occurs. Indeed, as there is no invariant on x on the initial location the edge is enabled for values greater or equal to 3. Upon firing of the edge the update is executed: the clock x and the bounded integer c are both reset to 0. The edge from *Count* to *Inv* is nearly identical to the previous edge: when x is greater or equal to 3 the edge may be fired but x is reset to 1 instead of 0 and c is not modified. As a consequence, the value of x in *Inv* is between 0 and 2 when the location is first entered and between 1 and 2 on every subsequent visit. The transition between *Init* and *Count* has no guard and shows that an update may consist of a complex expression: the update `c = (c + 1) % 7` increases c by 1 modulo 7.

As explained in the introduction, model checking relies on accurate long term models. On-line model checking is a variant of classic model-checking that eliminates the need for such models and thus may be applied when they are unavailable. It reduces the modeling error by periodically adjusting the current state to the observed real state, e.g., by setting a model value to the exact value measured by a sensor attached to a patient. For example, if we consider the model in Figure 1 one could assume that the counter variable c is modeling some patient's parameter. If that parameter in reality occasionally jumps the model is inaccurate and needs to be adjusted by setting c to the correct value. On-line model checking performs the adjustments and thus the jumps do not need to be modeled accurately. Note that errors may still be present in the system under on-line model checking but the method predicts them in advance to react to them.

On-line model checking requires the model analysis to finish before the next update interval to give meaningful results. Although generally the main work is done by the model checker the reconstruction still consumes some time, which our method reduces compared to the naive automatic reconstruction approach.

IV. UPPAAL's State Space

UPPAAL's state space can be divided into three parts: the time state, the location state, and the data state. The location and the data state are straightforward: every data variable is assigned exactly one value for the data state and the location state consists of the current location vector, i.e., a vector that contains the current location for every automaton instance. The time state is more complicated as it needs to capture all possible valuations for every clock in the model as well as all relations between the clocks. *Difference bound matrices* (DBM) are a common and simple representation method for such time states [18][19]. By introducing a static zero clock in addition to all the clocks in the model ($\mathcal{C}_0 = \mathcal{C} \cup \mathbf{0}$, $\mathcal{C}$ the set of all clocks) all necessary clock constraints can be written in the form $x - y \preceq n$ where x and y are clocks ($x, y \in \mathcal{C}_0$), $\preceq$ is a comparator ($\preceq \in \{<, \leq\}$), and n is an integer ($n \in \mathbb{Z}$). A value in a difference bound matrix then is a tuple of an integer and a comparator $(n, \preceq)$, $n \in \mathbb{Z}$, $\preceq \in \{<, \leq\}$ or the special symbol ∞, which indicates no bound. We denote the set of such entries by $\mathcal{K} = (\{\mathbb{Z} \times \{<, \leq\}\} \cup \infty)$. An order on the entries is given by $(n, \preceq) < \infty$, $(n_1, \preceq_1) < (n_2, \preceq_2)$ if $n_1 < n_2$, and $(n, <) < (n, \leq)$. Furthermore, addition is defined as follows: $(n, \preceq) + \infty = \infty$, $(m, \leq) + (n, \leq) = (m + n, \leq)$, and $(m, <) + (n, \preceq) = (m + n, <)$. A difference bound matrix thus contains one bound, either including or excluding, for every pair of clocks: $\mathbf{M} \in \mathcal{K}^{|\mathcal{C}_0| \times |\mathcal{C}_0|}$.

As an example a clock constraint system with two clocks a and b and the constraints $a \in [2, 4)$, $b > 5$, and $b - a \geq 3$ is transformed to the canonical constraints $a - \mathbf{0} < 4$, $\mathbf{0} - a \leq -2$, $b - \mathbf{0} < \infty$, $\mathbf{0} - b < -5$, $a - b \leq -3$, and $b - a < \infty$. The corresponding DBM is

$$\begin{array}{c|ccc} & \mathbf{0} & a & b \\ \hline \mathbf{0} & 0 & (-2, \leq) & (-5, <) \\ a & (4, <) & 0 & (-3, \leq) \\ b & \infty & \infty & 0 \end{array}$$

During simulation of an UPPAAL model its transitions are repeatedly executed. Every transition generally has multiple effects on the time state and each such effect corresponds to a transformation of the difference bound matrix that represents the current time state. The following summary lists the DBM transformations necessary to traverse the state space [19]:

- *Clock Reset* A clock reset is performed when an edge is fired that has an update for a clock variable (`x = n`). A clock reset sets the upper and lower bound on the clock x to the given value and depending constraints, i.e., constraints on a clock difference involving x are adjusted. This corresponds to modifying the matrix row and column for the clock x.
- *Constraint Introduction* A constraint introduction is performed if either a firing edge has a guard on a clock or an invariant on a clock is present in a current location and the bound is more restrictive than the current constraint on the involved clock. In that case the relevant matrix entry is set to the new constraint and for all other entries in the matrix it is checked whether the new bound induces stricter bounds.

- *Bound Elimination* Bound elimination is performed when a new location is entered. All bounds on clock constraints of the form $c - \mathbf{0} < n$ are removed, i.e., the upper bounds on clocks are removed. Bound elimination is equivalent to setting the first matrix column except the top-most value to ∞.
- *Urgency Introduction* An urgency introduction is performed if an urgent or committed location is entered or an entered location has an outgoing, enabled transition that synchronizes on an urgent channel. Unlike the previous transformations, urgency is a modeling construct specific to UPPAAL to prevent time from passing. An urgency introduction is semantically equivalent to introducing a fresh clock on the incoming edge and adding a new invariant on that clock with a bound of 0 to the location. An urgency introduction thus can be derived from a clock reset and a constraint introduction.

Returning to the example model (Figure 1) the individual transitions can now be broken down into their respective transformations. The initial location *Init* induces a bound elimination on the initial state where all clocks are set to zero. The transition from *Init* to *Inv* yields a constraint introduction for the guard (`x >= 3`) and a subsequent clock reset (`x = 0, c = 0`). The reset of the bounded integer c is ignored here as c is part of the data state. The location *Inv* results in a bound elimination and a following constraint introduction to accommodate the invariant (`x <= 2`). The transition from *Inv* to *Count* simply induces a single clock reset transformation before the location *Count* eliminates the bound on the state space again. Lastly, the transition from *Count* to *Inv* introduces the same kind of transformations as the transition from *Init* to *Inv*: both perform a constraint introduction and a clock reset. The values computed for the clock variable x are as follows:

1) Location *Init*
   a) Initial: $x = 0$
   b) Bound Elimination: $x \in [0, \infty)$
2) Transition *Init* $\longrightarrow$ *Inv*
   a) Constraint Introduction: $x \in [3, \infty)$
   b) Clock Reset: $x = 0$
3) Location *Inv*
   a) Bound Elimination: $x \in [0, \infty)$
   b) Constraint Introduction: $x \in [0, 2]$
4) Transition *Inv* $\longrightarrow$ *Count*
   a) Clock Reset: $x = 0$
5) Location *Count*
   a) Bound Elimination: $x \in [0, \infty)$
6) Transition *Count* $\longrightarrow$ *Inv*
   a) Constraint Introduction: $x \in [3, \infty)$
   b) Clock Reset: $x = 1$
7) Location *Inv*
   a) Bound Elimination: $x \in [1, \infty)$
   b) Constraint Introduction: $x \in [1, 2]$

V. State Space Reconstruction

In many models a large number of past transitions do not have an impact on the current state space. In the example model (Figure 1) this behavior can be observed: in the location *Count* the clock x is in the range $[0, \infty)$. This valuation was completely created by the clock reset of the ingoing edge and the bound elimination of the location itself. Previous state space transformations do not influence the valuation of x. Therefore, instead of executing the transition sequence *Init* $\longrightarrow$ *Inv* $\longrightarrow$ *Count* totaling 7 transformations only 3 transformations are required to reach the same state space. The introduction of a new initial state and the direct transition to *Count* with an update `x = 0` is sufficient to recreate this time state space. During reconstruction it is thus beneficial to exploit the fact that effects of certain state space transformations are overwritten by subsequent transformations.

The remainder of this section first formally defines a transformation system and conditions for removing transformations from a sequence (Subsection V-A). Subsection V-B then presents our reduction algorithm based on use-definition chains. The application of the algorithm to the reconstruction problem in UPPAAL is discussed in Subsection V-C. Lastly, Subsection V-D summarizes the complete reconstruction process in UPPAAL.

A. Transformation Elimination Formalized

We now formally derive when transformations may be removed from a general transformation sequence. Examples of the individual parts of the formalization can be found in Subsection V-C, where we specialize the formalization to UPPAAL. Parts of our formalization and its specialization to UPPAAL are also published at FM 2014, where a more recent graph-based state space reconstruction algorithm using the same transformation abstraction is described [20].

Definition 1. *An* evaluation function *is a mapping*

$$e : \mathcal{V} \rightarrow \mathcal{D}$$

where $\mathcal{V}$ is a set of variables and $\mathcal{D}$ is the valuation domain of these variables.

We denote the set of all evaluation functions by $\mathcal{E}(\mathcal{V}, \mathcal{D})$.

Definition 2. *A* transformation *of the evaluation functions is a mapping*

$$t : \mathcal{E}(\mathcal{V}, \mathcal{D}) \rightarrow \mathcal{E}(\mathcal{V}, \mathcal{D})$$

Let $e_1 \xrightarrow{t_1} e_2 \xrightarrow{t_2} \ldots \xrightarrow{t_{N-1}} e_N$ be a sequence of evaluation functions created by the transformations t_i where $\xrightarrow{t}$ indicates the application of a single transformation t. We denote the transformation sequence $t_1, t_2, \ldots, t_{N-1}$ by T and the evaluation function sequence $e_1, e_2, \ldots, e_N$ by E.

For the transformation reduction we are interested in finding a subsequence $T' \subseteq T$ such that $e_1 \overset{T'}{\Longrightarrow} e_N$ where $\overset{T}{\Longrightarrow}$ denotes the ordered application of a sequence of transformations T. Note that we use set notation for sequences although sequences are ordered and may contain duplicates. Such a transformation sequence T' then results in the same last evaluation function e_N but has potentially fewer transformations than T.

The reduction requires the specification of the transformations involved to capture their influence on the evaluation functions.

Definition 3. *An* evaluation calculation *is a mapping*

$$m : 2^{\mathcal{V}} \times \mathcal{E}(\mathcal{V}, \mathcal{D}) \to \mathcal{D}$$

where $m(V, e)$ *computes a domain value by using exactly the evaluations* $e(v)$ *where* $v \in V$.

For example, the evaluation calculation

$$\text{add} : (\{ x, y \}, e) \mapsto e(x) + e(y)$$

calculates the sum of the two variables x and y. We denote the set of all evaluation calculations by $\mathcal{C}(\mathcal{V}, \mathcal{D})$.

Definition 4. *The set of* specifications *is*

$$\mathcal{S} = \mathcal{V} \times 2^{\mathcal{V}} \times \mathcal{C}(\mathcal{V}, \mathcal{D})$$

We can specify transformations by providing a subset of $\mathcal{S}$ to the specification function:

Definition 5. *The* specification function *is a mapping*

$$\begin{aligned} s : 2^{\mathcal{S}} &\to (\mathcal{E}(\mathcal{V}, \mathcal{D}) \to \mathcal{E}(\mathcal{V}, \mathcal{D})) \\ S &\mapsto (e \mapsto e') \qquad \textit{where} \\ e'(x') &= \begin{cases} m(V, e) & \textit{if } (x, V, m) \in S \wedge x' = x \\ e(x') & \textit{otherwise} \end{cases} \end{aligned}$$

where $\forall (x, V, m), (x', V', m') \in S \, [x = x' \implies V = V' \wedge m = m']$

Definition 5 states that for each $x \in \mathcal{V}$ a specification set S may contain at most one element (x, V, m). The uniqueness of x in S ensures that the specification function $s(S)$ is well-defined as otherwise the definition for $e'(x')$ is ambiguous. We write (x, V_x, m_x) for the single element for x in S if it exists and further associate the evaluation calculation m_x with a term $m_x(V_x)$ that uses the variable symbols in V_x. We call the term $m_x(V_x)$ *irreducible* if there is no equivalent term in the underlying term algebra that uses fewer subterms, e.g., due to distributivity or the presence of zeros.

Proposition 1. *If there is no equality relationship between the variable symbols, expressed in the term algebra, then there exists a unique specification set* S *where all evaluation calculation terms are irreducible.*

Proof: Let S, S', $S \neq S'$ be two specification sets with irreducible evaluation calculation terms such that $t = s(S) = s(S')$. Then for all variables $x \in \mathcal{V} \, [m_x(V_x) = m'_x(V'_x)]$, i.e., the terms are equivalent in the term algebra. Assume w.l.o.g. V_x contains a variable symbol y not contained in V'_x. Then either the term $m_x(V_x)$ can be rewritten in a way that eliminates y and $m_x(V_x)$ was not irreducible or the variable symbol y can not be eliminated. Then $m_x(V_x) = m'_x(V'_x)$ constitutes a non-trivial equality relation between the variable symbols. Both cases contradict the assumption and thus $S = S'$. ■

Definition 6. *The* specification set *of a transformation* t *is the unique set*

$$S(t) \subset \mathcal{S}$$

with irreducible evaluation calculation terms such that $t = s(S(t))$.

The specification set $S(t)$ of a transformation t captures all modifications to the input evaluation function as every member (x, V, m) defines which (x) and how (m) a variable is modified. Thus, providing a specification set for t fully characterizes t. Furthermore, using the specification set we can derive the write and read characteristics of the transformation.

Definition 7. *The* write set *of a transformation* t *is*

$$\mathcal{W}(t) = \bigcup_{(x,V,m) \in S(t)} \{ x \}$$

Definition 8. *The* read set *of a transformation* t *is*

$$\mathcal{R}(t) = \bigcup_{(x,V,m) \in S(t)} V$$

Moreover, to conveniently chain transformations together we define *compound transformations* in addition to the simple transformations.

Definition 9. *A* compound transformation *is a transformation*

$$t_c = t_1 \circ t_2 \circ \cdots \circ t_n$$

such that $e_1 \xrightarrow{t_c} e_{n+1} \equiv e_1 \xrightarrow{t_1} e_2 \xrightarrow{t_2} \ldots \xrightarrow{t_n} e_{n+1}$.

Note that $\circ$ denotes concatenation ($(a \circ b)(x) = b(a(x))$) in contrast to composition ($(a \circ b)(x) = a(b(x))$). For compound transformations the write and read sets can be derived from the individual transformations t_i.

Definition 10. *The* write set *of a compound transformation* t_c *is*

$$\mathcal{W}(t_c) = \bigcup_i \mathcal{W}(t_i)$$

Definition 11. *The* read set *of a compound transformation* t_c *is*

$$\mathcal{R}(t_c) = \bigcup_i (\mathcal{R}(t_i) \setminus \bigcup_{j<i} \mathcal{W}(t_j))$$

Using these definitions two cases exist where a transformation t_i may be removed from the transformation sequence T without changing e_N. The cases can be described in the following way:

1) *no write*
$\mathcal{W}(t_i) = \emptyset$
The transformation t_i may be removed if the write set $\mathcal{W}(t_i)$ is empty.
Proof: As $\mathcal{W}(t_i) = \emptyset \implies S(t_i) = \emptyset \implies t_i = I$ by definition we find $e_i \xrightarrow{t_i} e_{i+1} \implies e_i = e_{i+1}$ and thus $e_i \xrightarrow{t_i} e_{i+1} \xrightarrow{t_{i+1}} e_{i+2} = e_i \xrightarrow{t_{i+1}} e_{i+2}$. ■

2) *no read overwritten*
$\forall x \in \mathcal{W}(t_i) \, [j > i \wedge x \in \mathcal{R}(t_j) \implies \exists k \, [i < k < j \wedge x \in \mathcal{W}(t_k)]]$
The transformation t_i may be removed if for all following transformations t_j that read a variable written by t_i there is a transformation t_k between t_i and t_j that writes that variable.
Proof: Let $e'_i \xrightarrow{I} e'_{i+1} \to \cdots \to e'_M$ be the transition sequence E' that replaces t_i with the identity transformation I in $e_i \xrightarrow{t_i} e_{i+1} \to \cdots \to e_M$. Note

that the new transformation sequence is equivalent to removing t_i in the old one. Now if we assume $e_M \neq e'_M$ then there must be a variable x such that $e_M(x) \neq e'_M(x)$. It follows that there must be a transformation t_j, $j < M$, such that $e_j(x) = e'_j(x)$ and $e_{j+1}(x) \neq e'_{j+1}(x)$ because $e_i = e'_i$. Hence, the variable x is written by t_j, i.e., $x \in \mathcal{W}(t_j)$. As $x \in \mathcal{W}(t_j)$, $e_{j+1}(x) = m(V, e_j)$ for $(x, V, m) \in S(t_j)$ and there must be a variable $v \in \mathcal{R}(t_j)$ such that $e_j(v) \neq e'_j(v)$ to satisfy $m(V, e_j) \neq m(V, e'_j)$. If $j = i$, we have a contradiction to the assumption $e_i = e'_i$ and are finished. Otherwise, we apply the same argument to the variable v and would get another transformation t_k, $k < j$, and a variable $w \in \mathcal{R}(t_k)$ such that $e_k(w) \neq e'_k(w)$. This process leads to a contradiction to our assumption in at most $M - i$ iterations. ■

B. Use-Definition Chain Reduction

The key idea of our approach is the construction of use-definition chains to identify transformations satisfying the requirements presented in Subsection V-A. A use-definition chain is a data structure that provides information about the origins of variable values: for every use of a variable the chain contains definitions that have influenced the variable and ultimately lead to the current value. Our idea is to adapt the definition-use chain technique from static data flow analysis on a program's source code to the state space reconstruction: every entry in the model's difference bound matrix is treated as variable and thus is observed for uses and modifications. DBM entries are only modified by applying a state space transformation on the DBM. We thus analyzed the read and write access to matrix entries for every transformation to derive the use-definition chains where the transformations are the basic operations.

We now propose an algorithm that removes the transformations in question. Our algorithm consists of two smaller algorithms: the APPLY algorithm and the ELIMINATE algorithm. The APPLY algorithm is used to perform a transformation and the ELIMINATE algorithm removes unnecessary transformations in a sequence of applied transformations. Usage of the algorithms is assumed as follows: for a sequence of transformations T first the APPLY algorithm is called on all transformations in order, then the ELIMINATE algorithm is executed to obtain the reduced transformation sequence. Note that a transformation sequence T could be split into two subsequent sequences T_1 and T_2 and the ELIMINATE algorithm could be run on the sequence T_1 as soon as all transformations in T_1 have been applied. Thus, the ELIMINATE algorithm can be run with an appropriate transformation sequence after every execution of APPLY , which achieves on-the-fly removal. However, for formalization purposes, we assume the algorithm execution sequence given in Figure 2 where the initial mappings c_1, r_1 and u_1 satisfy $\forall t \in T\, [c_1(t) = 0 \wedge u_1(t) = \emptyset] \wedge \forall x \in \mathcal{V}[r_1(x) = \perp] \wedge c_1(\perp) = |\mathcal{V}|$. The algorithm generates the following sequences:

- Transformation sequence $T = t_1, t_2, \ldots, t_{N-1}$
- Evaluation function sequence $E = e_1, e_2, \ldots, e_N$
- Reference counter sequence $C = c_1, c_2, \ldots, c_N$

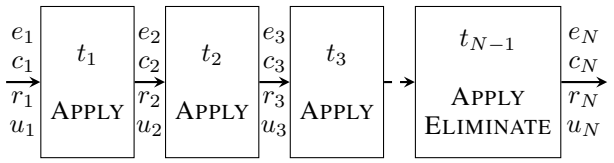


Figure 2. Algorithm Execution Sequence

- Responsibility sequence $R = r_1, r_2, \ldots, r_N$
- Use-definition sequence $U = u_1, u_2, \ldots, u_N$

where a *reference counter* is a mapping $c : T \cup \{\perp\} \to \mathbb{N}_0$, a *responsibility mapping* is a mapping $r : \mathcal{V} \to T \cup \{\perp\}$, and an *use-definition mapping* is a mapping $u : T \to 2^T$.

Figure 3 shows the APPLY algorithm. The algorithm takes the to-be-applied transformation t, an evaluation function e, a reference counter c, a responsibility mapping r, and a use-definition mapping u as parameters. Except t all the parameters are stored locally in lines 2 to 5 to allow internal manipulations and the results are returned in lines 17 to 20. The algorithm can be divided into two parts, one handling the reads of the transformation t and one handling its writes. Lines 6 to 11 process the variables read by t. For every variable x_i it is determined if the transformation responsible for its current valuation $r(x_i)$ is already used by t in line 7. If $r(x_i)$ was not marked used yet it is marked used in line 9 and the reference counter is increased accordingly in line 8. Lines 12 to 16 then handle the writes of t. First the evaluation function is updated to map x_i to the value $m_i(V_i, e)$ as specified by the transformation. Then line 14 reduces the reference counter of the transformation previously responsible for the value of x_i and line 15 updates the responsibility mapping such that now the transformation t is responsible for the value of x_i. Lastly in the return section in line 18 the reference counter is set to $|S(t)|$ to show that t is now responsible for $|S(t)|$ variable valuations.

Figure 4 shows the ELIMINATE algorithm. The algorithm takes a sequence of transformations T, a reference counter c, and a use-definition mapping u as parameters where all transformations in T must already have been applied with the APPLY algorithm. In lines 2 to 3 the transformation sequence and the reference counter are stored locally and the results of the algorithm are returned in lines 16 to 17. In between, in lines 4 to 15, a fix point is calculated by removing as many transformations from T as possible. The algorithm checks for every transformation t in T if the reference counter $c(t)$ evaluates to zero in line 7. If a transformation t satisfies $c(t) = 0$ the algorithm at first removes t from T in line 8, then adjusts the reference counter by decreasing all counters of transformations used by t in lines 9 to 11, and lastly schedules another iteration of the fix point calculation in line 12 as the modifications to the reference counters may induce additional removals.

We now show the correctness of the algorithm. We assume that there implicitly is a transformation t_e appended to the transformation sequence T that satisfies $\mathcal{R}(t_e) = \mathcal{V}$ to indicate that the final calculated values are actually read and need to be recreated correctly. Otherwise, the whole transformation

```
 1: procedure APPLY(t, e, c, r, u)
 2:     e_l ← e
 3:     c_l ← c
 4:     r_l ← r
 5:     u_l ← u
 6:     for all x_i ∈ R(t) do
 7:         if r(x_i) ∉ u_l(t) then
 8:             c_l ← c_l[r(x_i) ↦ c_l(r(x_i))+1]
 9:             u_l ← u_l[t ↦ u_l(t) ∪ { r(x_i) }]
10:         end if
11:     end for
12:     for all (x_i, V_i, m_i) ∈ S(t) do
13:         e_l ← e_l[x_i ↦ m_i(V_i, e_l)]
14:         c_l ← c_l[r(x_i) ↦ c_l(r(x_i)) − 1]
15:         r_l ← r_l[x_i ↦ t]
16:     end for
17:     e' ← e_l
18:     c' ← c_l[t ↦ |S(t)|]
19:     r' ← r_l
20:     u' ← u_l
21: end procedure
```

Figure 3. APPLY Algorithm

```
Require: ∀t ∈ T[APPLY(t, ...)]
 1: procedure ELIMINATE(T, c, u)
 2:     T_l ← T
 3:     c_l ← c
 4:     repeat
 5:         b ← true
 6:         for all t ∈ T_l do
 7:             if c_l(t) = 0 then
 8:                 T_l ← T_l \ { t }
 9:                 for all s ∈ u(t) do
10:                     c_l ← c_l[s ↦ c_l(s)−1]
11:                 end for
12:                 b ← false
13:             end if
14:         end for
15:     until b = true
16:     T' ← T_l
17:     c' ← c_l
18: end procedure
```

Figure 4. ELIMINATE Algorithm

sequence T could be removed as no data is consumed. As a transformation t can only be removed in line 8 of the ELIMINATE algorithm, which is only executed if $c(t) = 0$, the following implications need to be shown to prove the algorithm.

1) $\mathcal{W}(t) = \emptyset \implies \forall c \in C\,[c(t) = 0] \quad (\Rightarrow)$
   If a transformation t does not write any variable then the transformation may be removed at any time.
2) $\forall x \in \mathcal{W}(t_i)\,[\forall t_j \in T\,[i < j \wedge x \in \mathcal{R}(t_j) \implies \exists t_k \in T\,[i < k < j \wedge x \in \mathcal{W}(t_k)]]] \implies \forall c_l \in C\,[l > \max k \implies c_l(t_i) = 0] \quad (\Rightarrow)$
   If every variable written by a transformation t_i is overwritten by transformations t_k before it is read by a transformation t_j then the transformation may be removed as soon as the last overwriting transformation t_k is performed.
3) $i < l \wedge c_l(t_i) = 0 \implies \mathcal{W}(t_i) = \emptyset \vee \forall x \in \mathcal{W}(t_i)[\nexists t_j \in T[i < j \wedge x \in \mathcal{R}(t_j) \wedge c_l(t_j) \neq 0 \wedge \forall t_k \in T\,[i < k < j \implies x \notin \mathcal{W}(t_k)]]] \quad (\Leftarrow)$
   If a transformation t_i may be removed the transformation either does not write any variable or every variable written by it is overwritten overwritten by transformations t_k without being read beforehand.
   Derivation Note: Application of identities yields $\forall x \in \mathcal{W}(t_i)\,[\forall t_j \in T\,[i < j \wedge x \in \mathcal{R}(t_j) \implies \exists t_k \in T\,[i < k < j \wedge x \in \mathcal{W}(t_k)]]] \Leftrightarrow \forall x \in \mathcal{W}(t_i)\,[\nexists t_j \in T\,[i < j \wedge x \in \mathcal{R}(t_j) \wedge \forall t_k \in T\,[i < k < j \implies x \notin \mathcal{W}(t_k)]]]$. Adding $c_l(t_j) \neq 0$ only makes sure the transformation can not be removed, i.e., it really exists.

We first derive two lemmas that characterize execution dependencies of certain lines in the algorithms to break the proof down into smaller parts.

Lemma 1. *Every execution of line 10 in the* ELIMINATE *algorithm is preceded by an execution of line 8 of the* APPLY *algorithm where $r(x) = s$ for some variable x, i.e., for every decrease of $c_j(s)$ in line 10 there is an increase of $c_i(s)$ in line 8 where $i < j$.*

Proof: An execution of line 10 in the ELIMINATE algorithm for a transformation s implies $s \in u(t)$ for some transformation t because of line 9. As only line 9 of the APPLY algorithm modifies u an execution of it is implied where $r(x) = s$ for some variable x. Additionally, line 10 may not be executed multiple times with the same transformation s for a single execution of line 8 because $u(t)$ is a set and therefore does not contain duplicates of s and the transformation t is removed from T_l in line 8 rendering a subsequent access to $u(t)$ impossible. It follows that every execution of line 10 in the ELIMINATE algorithm is paired with a preceeding execution of line 8 in the APPLY algorithm. ■

Lemma 2. *Before the execution of the* APPLY *algorithm for a transformation t there are zero variables x_i that satisfy $r(x_i) = t$. After its execution there are at all times at most $|S(t)|$ variables x_i that satisfy $r(x_i) = t$ for a transformation t.*

Proof: Only line 15 of the APPLY algorithm may modify $r(x)$. A valuation satisfying $r(x) = t$ can only be established when it is executed for the transformation t. In that case line 15 is executed $|S(t)|$ times due to line 12 and because $\forall (x_i, V_i, m_i), (x_j, V_j, m_j) \in S(t)\,[x_i \neq x_j]$ there are exactly $|S(t)|$ variables x_i that satisfy $r(x_i) = t$ after the execution of the APPLY algorithm for t. Because subsequent executions of line 15 always result in valuations $r(x) \neq t$ for a variable x it follows that the amount of variables x_i that satisfy $r(x_i) = t$ may only be reduced. The proposition follows by taking into consideration that initially $\forall x \in \mathcal{V}\,[r(x) = \perp]$. ■

Corollary 1. *Line 14 of the* APPLY *algorithm may be executed for a transformation t at most $|S(t)|$ times, i.e., line 14 reduces $c(t)$ by one at most $|S(t)|$ times.*

Proof: An execution of line 14 for a transformation t implies that $r(x) = t$ for a variable x. Additionally, the

execution is always followed by an execution of line 15, which sets $r(x) \neq t$. It follows that every execution of line 14 for a transformation t implies a reduction of the amount of variables that satisfy $r(x) = t$ by one. It follows that line 14 may at most be executed $|S(t)|$ times for a transformation t due to Lemma 2. ■

We now prove the algorithm correct by showing that the presented requirements hold.

1) $\mathcal{W}(t) = \emptyset \implies \forall c \in C\,[c(t) = 0]$

Proof: There are four occurrences where reference counters may be modified: in lines 8, 14 and 18 of the APPLY algorithm and in line 10 of the ELIMINATE algorithm.

a) APPLY algorithm, line 8
Due to Lemma 2 $\mathcal{W}(t) = \emptyset \implies S(t) = \emptyset \implies \forall r \in R\,[\nexists x \in \mathcal{V}\,[r(x) = t]]$ and thus line 8 cannot modify $c(t)$.

b) APPLY algorithm, line 14
Due to Lemma 2 $\mathcal{W}(t) = \emptyset \implies S(t) = \emptyset \implies \forall r \in R\,[\nexists x \in \mathcal{V}\,[r(x) = t]]$ and thus line 14 cannot modify $c(t)$.

c) APPLY algorithm, line 18
As $\mathcal{W}(t) = \emptyset \implies S(t) = \emptyset$ line 18 sets $c(t)$ to $|S(t)| = 0$ if APPLY is executed for t and $c(t)$ is not modified otherwise.

d) ELIMINATE algorithm, line 10
As line 8 cannot modify $c(t)$ (see above) line 10 cannot modify $c(t)$ due to Lemma 1.

According to the case analysis it follows that $c'(t) = c(t)$ or $c'(t) = 0$ for every application of the APPLY or ELIMINATE algorithm. Using that $c_1(t) = 0$ it follows that $\mathcal{W}(t) = \emptyset \implies \forall c \in C[c(t) = 0]$ by induction. ■

2) $\forall x \in \mathcal{W}(t_i)\,[\forall t_j \in T\,[i < j \wedge x \in \mathcal{R}(t_j) \implies \exists t_k \in T\,[i < k < j \wedge x \in \mathcal{W}(t_k)]]] \implies \forall c_l \in C\,[l > \max k \implies c_l(t_i) = 0]$

Proof: There are four occurrences where reference counters may be modified: in lines 8, 14 and 18 of the APPLY algorithm and in line 10 of the ELIMINATE algorithm. We consider the transformation t_i.

a) APPLY algorithm, line 8
Assume a transformation t_j modifies $c(t_i)$ in line 8. Then due to Lemma 2 it follows that $i < j$ and $\exists x \in \mathcal{V}\,[r(x) = t_i]$. It follows that $\exists x \in \mathcal{V}\,[x \in \mathcal{R}(t_j) \wedge x \in \mathcal{W}(t_i)]$ must be satisfied. Thus, the inner premise in the proposition holds and there must be a transformation t_k satisfying $i < k < j$ and $x \in \mathcal{W}(t_k)$. The execution of the APPLY algorithm for the transformation t_k, however, then results in $r(x) \neq t_i$ (see proof of Lemma 2) and t_j can no longer modify $c(t_i)$. The premise thus prevents line 8 from modifying $c(t_i)$.

b) APPLY algorithm, line 14
As we are only interested in modifications to $c(t_i)$ we only need to consider executions for transformations $t_j, j > i$ due to Lemma 2 as line 14 requires $\exists x \in \mathcal{V}[r(x) = t_i]$ to modify $c(t_i)$. According to Corollary 1 line 14 may reduce $c(t_i)$ maximally by $|S(t_i)|$. The maximum reduction occurs if $\forall x \in \mathcal{W}(t_i)\,[\exists t_j \in T\,[i < j \wedge x \in \mathcal{W}(t_j)]]$ (see proof of Corollary 1). According to the premise of the proposition this requirement is satisfied if $\forall x \in \mathcal{W}(t_i)\,[\exists t_j \in T\,[i < j \wedge x \in \mathcal{R}(t_j)]]$. This requirement, however, is always satisfied because of the implicit transformation t_e at the end of the transformation sequence, which satisfies $\mathcal{R}(t_e) = \mathcal{V}$. It follows that the premise of the proposition implies the existence of a set of transformations $T_r \subseteq T$, which satisfies $\mathcal{W}(t_i) \subseteq \bigcup_{t \in T_r} \mathcal{W}(t)$ and, thus, line 14 reduces $c(t_i)$ by $|S(t_i)|$ in total.

c) APPLY algorithm, line 18
As we are only interested in modifications to $c(t_i)$ we only need to consider the execution for t_i. In that case line 18 sets $c(t_i)$ to $|\mathcal{W}(t_i)|$ as $|\mathcal{W}(t_i)| = |S(t_i)|$.

d) ELIMINATE algorithm, line 10
Due to Lemma 1 there is no execution of line 10 that modifies $c(t_i)$ as there is no modification of $c(t_i)$ in line 8 in the APPLY algorithm (see above).

According to the case analysis $c(t_i)$ is modified in the following way: at first line 18 sets $c(t_i)$ to $|S(t_i)|$. Then the transformations from the set T_r are executed and lead to a monotonously descending $c(t_i)$ value (no reads). The execution of the last transformation from T_r results in a $c(t_i) = 0$. This transformation is the transformation with the highest k of the transformations t_k in the proposition as all transformations t_k satisfy $\mathcal{W}(t_i) \cap \mathcal{W}(t_k) \neq \emptyset$. Thus $\forall c_l \in C\,[l > \max k \implies c_l(t_i) = 0]$ is satisfied and the proposition holds. ■

3) $i < l \wedge c_l(t_i) = 0 \implies \mathcal{W}(t_i) = \emptyset \vee \forall x \in \mathcal{W}(t_i)[\nexists t_j \in T[i < j \wedge x \in \mathcal{R}(t_j) \wedge c_l(t_j) \neq 0 \wedge \forall t_k \in T\,[i < k < j \implies x \notin \mathcal{W}(t_k)]]]$

Proof: There are three occurrences where reference counters may be set or reduced to zero after t_i is processed. Lines 14 and 18 of the APPLY algorithm and line 10 of the ELIMINATE algorithm potentially modify $c_l(t_i)$ in such a way.

a) APPLY algorithm, line 14
In this case we prove $i < l \wedge c_l(t_i) = 0 \implies \forall x \in \mathcal{W}(t_i)\,[\nexists t_j \in T\,[i < j \wedge x \in \mathcal{R}(t_j) \wedge c_l(t_j) \neq 0 \wedge \forall t_k \in T\,[i < k < j \implies x \notin \mathcal{W}(t_k)]]]$ by contraposition. Assume x to be a variable satisfying $x \in \mathcal{W}(t_i)$ and assume t_j to be a transformation satisfying $i < j \wedge x \in \mathcal{R}(t_j) \wedge c_l(t_j) \neq 0 \wedge \forall t_k \in T\,[i < k < j \implies x \notin \mathcal{W}(t_k)]$. When executed t_j increases $c_j(t_i)$ in line 8 as $x \in \mathcal{R}(t_j)$ and no intermediate transformation t_k invalidates $r(x) = t_i$. Then for the transformation t_{l-1} to reduce $c_l(t_i)$ to zero in line 14 it is necessary to revert the increase by an execution of line 10 in the ELIMINATE algorithm before t_{l-1} is executed due to Corollary 1. Due to Lemma 1 the reduction must result from $c(t_j)$ being reduced to zero (ELIMINATE algorithm, line

7) as otherwise additional increases would have happened in line 8 beforehand. Assume t_m, $j < m < l-1$ to be the transformation that reduced $c_{m+1}(t_j)$ to zero to revert the increase of $c_j(t_i)$. We find that $c_{m+1}(t_j) = 0 \implies c_l(t_j) = 0$ unless $c(t_j)$ is increased again between the execution of t_m and t_{l-1}. However, a reduction of $c(t)$ to zero implies $\nexists x \in \mathcal{V}\,[r(x) = t]$ and thus such an increase is impossible. It follows that if t_j exists $c_l(t_i)$ can not be reduced to zero.

b) APPLY algorithm, line 18
Line 18 may only modify $c_l(t_i)$ if APPLY is executed for t_i. In that case $c_l(t_i)$ is set to zero if $|S(t_i)| = 0$. As $|S(t_i)| = 0 \implies \mathcal{W}(t_i) = \emptyset$ it follows that in this case $c_l(t_i) = 0 \implies \mathcal{W}(t_i) = \emptyset$, which is part of the proposition.

c) ELIMINATE algorithm, line 10
If line 10 reduces $c_l(t_i)$ to zero then there must have been a reduction of a different reference counter to zero beforehand as line 7 requires $c(t) = 0$. As this argument holds recursively a reduction in line 10 ultimately implies a reduction due to either line 14 or line 18 in the APPLY algorithm. Thus, a reduction in line 10 implies the implications for those lines found above.

Combining the case analysis results with the premise $i < l \wedge c_l(t_i)$ results in all cases in either $\mathcal{W}(t_i) = \emptyset$ or $\forall x \in \mathcal{W}(t_i)\,[\nexists t_j \in T\,[i < j \wedge x \in \mathcal{R}(t_j) \wedge c_l(t_j) \neq 0 \wedge \forall t_k \in T\,[i < k < j \implies x \notin \mathcal{W}(t_k)]]]$ yielding the proposition. ■

C. Algorithm Application to UPPAAL

We now specialize the general formalization to UPPAAL's state space and transformation system such that we may apply the presented reduction method to UPPAAL models. The reduction is only relevant for the time state of UPPAAL because the data and location states may be modified directly during on-line model checking. Only the time state has constraints that may be invalidated by individual changes to the time state.

Consider an UPPAAL model $\mathcal{M}$ with the clock set $\mathcal{C}$. The time state of $\mathcal{M}$ can be represented with a difference bound matrix containing $|\mathcal{C}_0|^2$ entries. Therefore, in our specialization the variable set $\mathcal{V}$ contains that many variables: $|\mathcal{V}| = |\mathcal{C}_0|^2$. The domain of the variables ($\mathcal{D}$) is $\mathcal{K}$ because every variable represents an DBM entry. We refer to the variables by $\mathrm{DBM}_{r,c}$ where r denotes the row number and c denotes the column number. Next, we define UPPAAL's DBM transformations for our formalization. As presented in Section IV, the relevant transformations are the *Clock Reset* ($\textsc{Reset}(x, v)$), the Constraint Introduction ($\textsc{Constraint}(x, y, v, \preceq)$), and the *Bound Elimination* (UP):

$\textsc{Reset}(x, v)$	Sets the clock variable x to the value v and adjusts constraints on that clock accordingly
$\textsc{Constraint}(x, y, v, \preceq)$	Introduces a new upper bound on a clock or on a difference of clocks and propagates dependencies
UP	Removes the upper bounds on every clock but not on differences of clocks

Four specification calculations are necessary to specify these transformations, where two assign values based on constants and two calculate minima. The first two are the $\mathrm{assign}(v)$ calculation and the $\mathrm{add}(v)$ calculation: $\mathrm{assign}(v)$ simply assigns the constant value v to a variable; $\mathrm{add}(v)$ calculates the sum of the constant value v and the current evaluation of a variable and assigns it:

$$\begin{aligned} \mathrm{assign} : \mathcal{K} &\to (2^{\mathcal{V}} \times \mathcal{E}(\mathcal{V}, \mathcal{D}) \to \mathcal{K}) \\ v &\mapsto ((\emptyset, e) \mapsto v) \\ \mathrm{add} : \mathcal{K} &\to (2^{\mathcal{V}} \times \mathcal{E}(\mathcal{V}, \mathcal{D}) \to \mathcal{K}) \\ v &\mapsto ((\{\, x \,\}, e) \mapsto e(x) + v) \end{aligned}$$

The minima calculations are $\mathrm{minassign}(v)$ and $\mathrm{minadd}()$. Assigning a variable with $\mathrm{minassign}(v)$ results in an evaluation equal to the minimum of v and the evaluation of the comparing variable. $\mathrm{minadd}()$ checks whether the sum of two variable evaluations is smaller than a third evaluation and if so assigns the sum:

$$\begin{aligned} \mathrm{minassign} : \mathcal{K} &\to (2^{\mathcal{V}} \times \mathcal{E}(\mathcal{V}, \mathcal{D}) \to \mathcal{K}) \\ v &\mapsto ((\{\, x \,\}, e) \mapsto \min(e(x), v)) \\ \mathrm{minadd} : 2^{\mathcal{V}} \times \mathcal{E}(\mathcal{V}, \mathcal{D}) &\to \mathcal{K} \\ (\{\, x, y, z \,\}, e) &\mapsto \min(e(x), e(y) + e(z)) \end{aligned}$$

Providing adequate specification sets $S(t)$ with these specification calculates now allows defining the transformations UP, $\textsc{Reset}(x, v)$, and $\textsc{Constraint}(x, y, v, \preceq)$. For convenience we use i_x and i_y for the indices of the clocks x and y in the DBMs. The UP transformation begins straight-forward: setting all values in the first DBM column except the top-most one to ∞ removes the upper bounds on the clocks:

$$S(\textsc{Up}) = \{\, (\mathrm{DBM}_{i,1}, \emptyset, \mathrm{assign}(\infty)) \mid 1 < i \leq |\mathcal{C}_0| \,\}$$

The $\textsc{Reset}(x, v)$ transformation performs two actions. First, it sets x to v, i.e., it sets both bounds to v. Then constraints in the clock's row and column are adjusted. A compound transformation models this behavior:

$$\begin{aligned} \textsc{Reset}(x, v) &= t_s \circ t_p \\ S(t_s) &= \{\, (\mathrm{DBM}_{i_x,1}, \emptyset, \mathrm{assign}((v, \leq))), \\ &\qquad (\mathrm{DBM}_{1,i_x}, \emptyset, \mathrm{assign}((-v, \leq))) \,\} \\ S(t_p) &= \{\, (\mathrm{DBM}_{i_x,i}, \{\, \mathrm{DBM}_{1,i} \,\}, \mathrm{add}((v, \leq))), \\ &\qquad (\mathrm{DBM}_{i,i_x}, \{\, \mathrm{DBM}_{i,1} \,\}, \mathrm{add}((-v, \leq))) \\ &\qquad \mid 1 < i \leq |\mathcal{C}_0| \,\} \end{aligned}$$

At last, the $\textsc{Constraint}(x, y, v, \preceq)$ transformation first introduces the constraint $x - y \preceq v$ and then propagates

it to depending constraints. Note that the propagation itself also is divided into multiple transformations as subsequent transformations require previous calculations:

$$\begin{aligned}\textsc{Constraint}(x, y, v, \preceq) = t_c \circ \\ t_{1,1} \circ \cdots \circ t_{1,|\mathcal{C}_0|} \circ \\ t_{2,1} \circ \cdots \circ t_{2,|\mathcal{C}_0|} \circ \\ \vdots \\ t_{|\mathcal{C}_0|,1} \circ \cdots \circ t_{|\mathcal{C}_0|,|\mathcal{C}_0|}\end{aligned}$$

$$t_{i,j} = t_{i,j,1} \circ t_{i,j,2}$$

$$S(t_c) = \{\, (\mathrm{DBM}_{i_x,i_y}, \{\, \mathrm{DBM}_{i_x,i_y} \,\}, \mathrm{minassign}((v, \preceq))) \,\}$$

$$S(t_{i,j,1}) = \{\, (\mathrm{DBM}_{i,j}, \{\, \mathrm{DBM}_{i,j}, \mathrm{DBM}_{i,i_x}, \mathrm{DBM}_{i_x,j} \,\}, \mathrm{minadd}) \,\}$$

$$S(t_{i,j,2}) = \{\, (\mathrm{DBM}_{i,j}, \{\, \mathrm{DBM}_{i,j}, \mathrm{DBM}_{i,i_y}, \mathrm{DBM}_{i_y,j} \,\}, \mathrm{minadd}) \,\}$$

D. Reconstruction Summarized

The complete state space reconstruction process consists of three steps:

1) *Initialization* Canonize model by introducing general starting points for later model synthesis, extract necessary information from the model, e.g., clock and variable definitions.
2) *Simulation* Select transitions in the model according to intended behavior, execute and store them. Simultaneously break them down into matching state space transformations and use the APPLY algorithm to internally construct the use-definition chains of the transformations. Then use the ELIMINATE algorithm to remove unnecessary transformations on-the-fly by evaluating the reference counters.
3) *Synthesis* Group the sequence of reduced transformations to form valid transitions and add the transitions to a newly created model obtained from the original one. The last transitions connect to the current locations when the reconstruction was initiated. Those transitions also update the data state.

Note that the synthesis of the actual UPPAAL model from the reduced transformation sequence has to take into consideration that UPPAAL allows a single automaton to be instantiated multiple times with possibly different parameters. During initialization of the reconstruction we therefore analyze the model definitions for automaton instantiation and save the relevant parameters. Also, as the location space needs to be correctly reconstructed an automaton that is instantiated multiple times has multiple initializations transitions for every instantiation. We use a single bounded integer variable in conjunction with appropriate guards to correctly order these transitions. Another important aspect of the synthesis step is that the model initialization needs to be self-contained, i.e., the initialization of multiple automata needs to finish synchronously to prevent parts of the model from advancing prematurely. As the initialization transitions per automaton may differ in length we employ a broadcast channel to synchronize the last transition to the original model. We use these final transitions to initialize the data variables as well. In case global variables are present an additional init automaton is introduced for their initialization. Figure 5 shows the reconstruction model for the example model (Figure 1) after 2 transitions on the right side. The additional initialization automaton is shown on the left. It sets the global, bounded integer c to 1. The clock x is set to 0 and the location is correctly initialized to *Count* after an initial first transition. The reconstructed model only needs to execute a single transition in contrast to the original model, which uses two to reach the correct state. For DBM transformations the reconstructed model uses three transformations while the original model needs seven.

TABLE I. EVALUATION RESULTS

Model	Transitions			Transformations		
	Before	After	Reduction	Before	After	Reduction
2doors	100	65.89	34.1%	364.7	254.46	30.2%
bridge	100	68.21	31.8%	188.39	144.09	23.5%
train-gate	100	66.18	33.8%	320.09	214.17	33.1%
fischer	100	91.27	8.7%	345.33	249.46	27.7%
csmacd2	100	100	0%	709.71	434.19	38.8%
csmacd32	75.58	75.58	0%	1818.6	327.49	79.7%
tdma	100	68.16	31.8%	719.88	240.11	66.6%
2doors	1000	627.9	37.2%	3722.3	2612.9	29.8%
bridge	1000	641.3	35.9%	1882.8	1436.4	23.7%
train-gate	1000	606.1	39.4%	3200.1	2194.1	31.4%
fischer	1000	853	14.7%	3455.3	2486.8	28%
csmacd2	1000	1000	0%	7238.1	4375.5	39.5%
csmacd32	619.6	619.6	0%	22491.1	2540.3	84%
tdma	1000	663.1	33.7%	6446.3	2651.5	58.9%

VI. EVALUATION

We evaluated our use-definition reconstruction method by applying it to seven different UPPAAL models and comparing it to the naive reconstruction approach. The models *2doors*, *bridge*, *train-gate*, and *fischer* are part of the UPPAAL example model suite. The *csmacd* models and *tdma* were taken from case studies [21][22]. We ran two test sets for every model. The first test executed 100 times 100 random transitions of the model before reconstructing the state. The second test set executed 1000 random transitions 10 times. For the *csmacd32* model it was not always possible to execute the maximum number of transitions during simulation as the model exhibits deadlock states. Table I shows our evaluation results. In the top half the results of the first test set and in the bottom half the results of the second test set are shown. All values are averages over the respective test runs but their variances are small. In our experiments the reduction of transformations is between 23% and 84% while the reduction of transitions is between 0% and 39.4%. This difference mainly stems from the fact that to delete a single transition all induced transformations need to be removed. However, our model synthesis algorithm still is unoptimized and sometimes produces unnecessary transitions. In cases where the transition reduction is higher than the transformation reduction the removal of transformations made it possible to merge multiple transitions. Interestingly, the *csmacd* models contain use-definition chains spanning the whole simulation, which prevent removal of transitions though many transformations are irrelevant to the state. Future work will need to address this issue, e.g., by also evaluating concrete state values. Regarding total execution time, our adjustments have a small impact as the model checking procedure consumes most

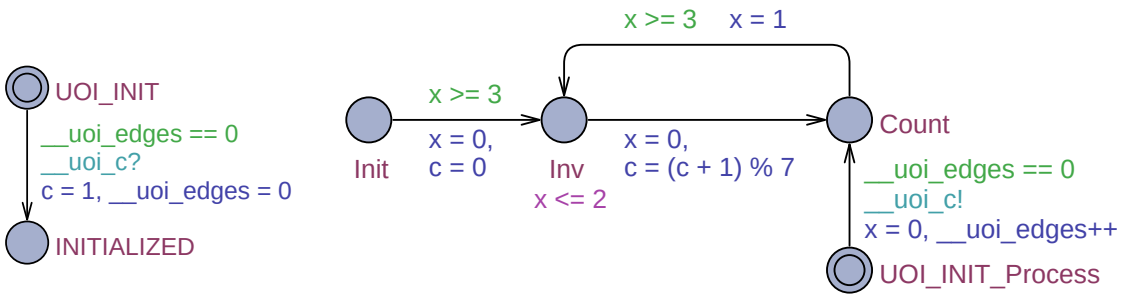


Figure 5. Reconstructed example model

of the time. Also, compared to the model checking part our approach scales well with the complexity of the used models.

VII. Conclusion and Future Work

In this paper, we addressed the problem of state space reconstruction of UPPAAL models in the context of on-line model checking. Our reconstruction method uses use-definition chains to track influence of individual transformations on the state space during model simulation. An algorithm for the chain construction with reference counters was presented. It is able to identify and remove transformations in the transformation sequence that do not have an impact on the final state space. We provided proofs for the algorithm itself and the requirements for the removal of a transformation without altering the final state. The reconstruction process was implemented in a prototype implementation and compared to the naive reconstruction approach, which does not remove any transformations. Seven UPPAAL models from different sources were analyzed and our approach reduced the amount of transformations necessary for reconstruction by 23% to 84% and reduced model transitions by up to 39.4%.

The prototype implementation is part of our UPPAAL on-line model checking interface that is currently in development. Interestingly, this interface could not only be used to automatically carry out on-line model checking. The interface also allows generic dynamic adaptation of model parameters. In the future enhancing on-line model checking by combining it with parameter learning algorithms and model calibration methods might broaden the applicability of model checking even further.

However, the proposed reconstruction method still yields infeasible reconstruction sequences for real-time on-line model checking in general as the reconstruction sequence length still grows over time. A reconstruction sequence of constant length is desirable to ensure real-time properties. Future research thus also need to focus on further optimizing the proposed reconstruction method. For example, the proposed method currently only relates transformations according to read and write accesses. Concrete variable values are not taken into account. Transformations that produce the same values could be removed, but are currently not. Experience during development has shown that such transformations occur often especially in periodic use-definition chains that arise due to cycles in the model. Removal of them could improve the reconstruction sequence significantly by breaking such cycles.

References

[1] J. Rinast, S. Schupp, and D. Gollmann, “State space reconstruction for on-line model checking with UPPAAL,” in VALID 2013, The Fifth Internation Conference on Advances in System Testing and Validation Lifecycle, 2013, pp. 21–26.

[2] T. Li, Q. Wang, F. Tan, L. Bu, J.-n. Cao, X. Liu, Y. Wang, and R. Zheng, “From offline long-run to online short-run: Exploring a new approach of hybrid systems model checking for MDPnP,” in 2011 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2011), 2011.

[3] T. Li, F. Tan, Q. Wang, L. Bu, J.-N. Cao, and X. Liu, “From offline toward real-time: A hybrid systems model checking and cps co-design approach for medical device plug-and-play (MDPnP),” in Proceedings of the 3rd ACM/IEEE International Conference on Cyber-Physical Systems - ICCPS ’12. Beijing, China: IEEE, April 2012, pp. 13–22.

[4] A. Hessel, K. G. Larsen, M. Mikucionis, B. Nielsen, P. Pettersson, and A. Skou, “Testing real-time systems using UPPAAL,” in Formal Methods and Testing, R. M. Hierons, J. P. Bowen, and M. Harman, Eds. Springer Berlin Heidelberg, 2008, pp. 77–117.

[5] Z. Qi, A. Liang, H. Guan, M. Wu, and Z. Zhang, “A hybrid model checking and runtime monitoring method for C++ web services,” in 2009 Fifth International Joint Conference on INC, IMS and IDC. Seoul, South Korea: IEEE, 2009, pp. 745–750.

[6] A. Easwaran, S. Kannan, and O. Sokolsky, “Steering of discrete event systems: Control theory approach,” Electronic Notes in Theoretical Computer Science, vol. 144, no. 4, 2006, pp. 21–39.

[7] G. Sauter, H. Dierks, M. Fränzle, and M. R. Hansen, “Light-weight hybrid model checking facilitating online prediction of temporal properties,” in 21st Nordic Workshop on Programming Theory, NWPT 09, vol. 2, Lyngby, Denmark, 2009.

[8] D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky, “Toward patient safety in closed-loop medical device systems,” in Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems - ICCPS ’10. Stockholm, Sweden: ACM New York, NY, USA, 2010, pp. 139–148.

[9] R. Alur and D. L. Dill, “A theory of timed automata,” Theoretical Computer Science, vol. 126, no. 2, 1994, pp. 183–235.

[10] W. Yi, P. Pettersson, and M. Daniels, “Automatic verification of real-time communicating systems by constraint-solving,” in 7th International Conference on Formal Description Techniques, D. Hogrefe and S. Leue, Eds., 1994, pp. 223–238.

[11] K. G. Larsen, P. Pettersson, and W. Yi, “Compositional and symbolic model-checking of real-time systems,” in Real-Time Systems Symposium, Pisa, Italy, 1995, pp. 76–87.

[12] J. Bengtsson, B. Jonsson, J. Lilius, and W. Yi, “Partial order reductions for timed systems,” in CONCUR’98 Concurrency Theory, D. Sangiorgi and R. de Simone, Eds. Springer Berlin Heidelberg, 1998, pp. 485–500.

[13] K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi, “Efficient verification of real-time systems: compact data structure and state-space reduction,” in Real-Time Systems Symposium, San Francisco, CA, USA, 1997, pp. 14–24.

[14] K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi, “Compact data struc-

tures and state-space reduction for model-checking real-time systems," Real-Time Systems, vol. 25, no. 2-3, 2003, pp. 255–275.

[15] J. Bengtsson, "Reducing memory usage in symbolic state-space exploration for timed systems," Department of Information Technology, Uppsala University, Uppsala, Sweden, Tech. Rep. May, 2001.

[16] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on Uppaal 4.0," Department of Computer Science, Aalborg University, Aalborg, Denmark, Tech. Rep., 2006.

[17] G. Behrmann, J. Bengtsson, A. David, K. G. Larsen, P. Pettersson, and W. Yi, "UPPAAL implementation secrets," in Formal Techniques in Real-Time and Fault-Tolerant Systems, W. Damm and E.-R. Olderog, Eds. Oldenburg, Germany: Springer-Verlag Berlin, 2002, pp. 3–22.

[18] J. Bengtsson and W. Yi, "Timed automata: Semantics, algorithms and tools," in Lectures on Concurrency and Petri Nets, J. Desel, W. Reisig, and G. Rozenberg, Eds. Springer Berlin Heidelberg, 2004, ch. 3, pp. 87–124.

[19] J. Bengtsson, "Clocks, dbms and states in timed systems," Ph.D. dissertation, Uppsala University, 2002.

[20] J. Rinast, S. Schupp, and D. Gollmann, "A graph-based transformation reduction to reach UPPAAL states faster," in 19th International Symposium on Formal Methods 2014 (FM2014), ser. Lecture Notes in Computer Science. Springer Verlag, 2014, pp. 547–562.

[21] S. Yovine, "KRONOS: a verification tool for real-time systems," International Journal on Software Tools for Technology Transfer, vol. 1, no. 1-2, 1997, pp. 123–133.

[22] H. Lönn and P. Pettersson, "Formal verification of a tdma protocol start-up mechanism," in Pacific Rim International Symposium on Fault-Tolerant Systems (PRFTS '97). Taipei: Ieee, 1997, pp. 235–242.

An Integrated SDN Architecture for Application Driven Networking

Andy Georgi
Technische Universität Dresden
E-Mail: Andy.Georgi@tu-dresden.de

Reinhard G. Budich
Max Planck Institute for Meteorology
E-Mail: Reinhard.Budich@zmaw.de

Yvonne Meeres
Max Planck Institute for Meteorology
E-Mail: Yvonne.Meeres@zmaw.de

Rolf Sperber
Embrace HPC-Network Consulting
E-Mail: Rolf.Sperber@embrace-net.de

Hubert Hérenger
T-Systems Solutions for Research GmbH
E-Mail: Hubert.Herenger@t-systems-sfr.com

***Abstract*—The target of our effort is the definition of a dynamic network architecture meeting the requirements of applications competing for reliable high performance network resources. These applications have different requirements regarding reliability, bandwidth, latency, predictability, quality, reliable lead time and allocatability. At a designated instance in time a virtual network has to be defined automatically for a limited period of time, based on an existing physical network infrastructure, which implements the requirements of an application. We suggest an *integrated Software Defined Network* (SDN) architecture providing highly customizable functionalities required for efficient data transfer. It consists of a service interface towards the application and an open network interface towards the physical infrastructure. Control and forwarding plane are separated for better scalability. This type of architecture allows to negotiate the reservation of network resources involving multiple applications with different requirement profiles within multi-domain environments.**

Keywords – Software Defined Networking, Huge Data, Network Architecture

I. Introduction

The amount of data to be handled by networks and associated resources across industry, universities and supercomputing centers for research, education, commerce and the internet business at large, grew significantly over the past few years. Furthermore, international collaboration became standard. Federation techniques implement a consolidated view on distributed data for end users. On the other hand, hybrid network architectures, multi-vendor environments and heterogeneous infrastructures steadily increase the complexity of data mining, computing and networking. Software Defined Networking (SDN) is a promising solution for the reduction of complexity: It opens the control layer allowing for direct programmability. Our target – first introduced in 2013 for geographically, dispersed datasets [1] – is to provide an automated arbitration layer between applications and network, thus reducing the operational complexity within heterogeneous environments. Furthermore, network resources can be distributed in a more efficient way by offering an open network interface for the application layer, which can be used to specify user requirements even beyond mere networking. Additionally, a central management provides a global view on the underlying infrastructure and enables the optimization of demands and available resources.

In this extended journal paper, we introduce an integrated multi-domain SDN architecture, in which network control is decoupled from forwarding and directly programmable by open interfaces between different layers. Therefore, Section II gives an introduction into SDN, followed by SDN use cases in Section III. In Section IV some approaches providing automated configuration of network services are discussed and differentiated from our approach. Our architecture as well as an automated network configuration process arbitrating between the concurrent applications competing for network resources is described in Section V. Section VI describes a feasible migration process from existing network architectures to an application driven network architecture step by step. Finally, we present our results from a first prototype in a 400-Gigabit/s-Testbed in Section VII and summarize our approach in Section VIII.

II. Software Defined Networking

Today, network intelligence and state are inherent part of network devices and distributed among the entire infrastructure. Decisions can only be made based on local information, which often results in an inefficient distribution of global resources. Implementing network-wide policies can increase the efficiency of resource distribution, but therefore, all participating devices have to be configured. This results in long delays and implies additional expenditure and can also lead to inconsistencies, security breaches or non-compliance to regulations. In addition, large discrete sets of protocols are used to connect hosts over arbitrary distances, link speeds and topologies. Most of these protocols tend to be defined in isolation, without any abstraction layer. This leads to a more and more increasing complexity and as a result, to static networks in a dynamic IT environment.

Based on this heterogeneous, complex and static infrastructure, applications and network services with different requirements try to utilize the network at the same time. In most cases these resources are offered as a best effort service, which means they are distributed between the streams, depending

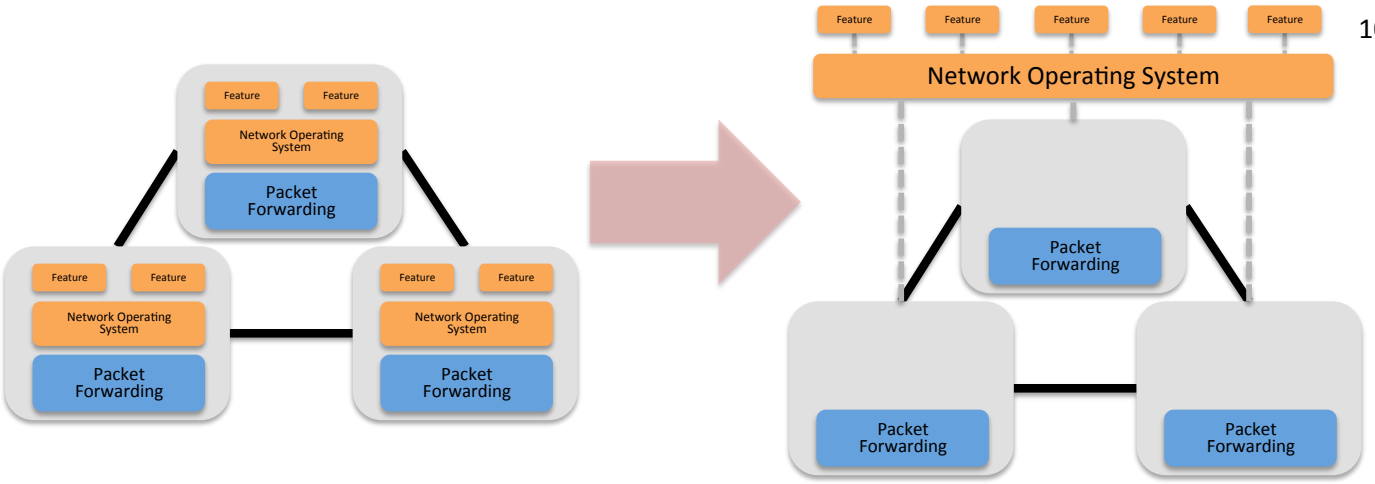


Figure 1: Conventional vs. SDN driven network, full separation of forwarding and control plane

on the current load. Because of missing information at the application layer, users do not know about the optimal point in time to start a data transfer. On the other hand, there is also no mechanism available which allows users to announce their requirements, so that the infrastructure can plan the upcoming traffic in advance. To enable an efficient resource management and reduce the complexity within networks, an open and programmable control layer is necessary, which interacts with users to manage their requirements, and with the underlying infrastructure, to map incoming requests to the available network devices.

Software Defined Networking (SDN) is an upcoming trend, promising the reduction of complexity: it opens the control layer and makes it directly programmable. SDN enabled networks provide an automated arbitration layer between applications and network and in consequence reduce the operational complexity within heterogeneous environments. Furthermore, network resources can be distributed in a more efficient way by offering an Open Network Interface for the Application Layer which can be used to specify user requirements. Additionally, a central management provides a global view on the underlying infrastructure and enables the optimization of demands and available resources. A comparison between the traditional network architecture and an SDN-enabled network is depicted in Figure 1.

III. Use Cases for SDN

In this section we introduce two applications, one from the climate community and one from high energy physics, both of which provide a use case for our architecture. So far if viewed separately. These two applications using the same SDN network simultaneously form a third use case.

Common buzz words of today's IT landscape are "Data Tsunami" or "Big Data". Whether countless small data packages have to be moved with minimal latency and highest safety, or humongous volumes of data have to be reliably moved from one place to another: The activities in data management rely on fast, broadband, reliable networks. At the moment, intercontinental network speeds are limited to 40 Gb/s [2]. The project *Advanced North Atlantic 100G Pilot* (ANA100G) tries to reach the 100 Gb/s barrier [3]. But practical experiences show, that the opportunistic networks (WANs) available today do not offer enough reliability, predictability and speed per cost necessary for the applications from the "Data Tsunami". Consequently, every "Big Data" application is a use case.

A practical example for these facts is the international *Coupled Model Inter-comparison Project* (CMIP) [4], which conducts sets of co-ordinated experiments with numerical climate models to compare them against each other. The project recently completed its 5th edition (CMIP5) [5], Such comparisons of climate models serve as basis for the *Assessment Reports*, on which the Nobel Laureate *International Panel on Climate Change* (IPCC) bases its recommendations for policy makers.

Numerical Climate Models are complex numerical realisations of the physical, chemical, biological and other processes that play a role in the climate system. They regularly utilise to a high percentage high performance computers like those to be found in the TOP500 list [6]. They also swamp these computers with data volumes at the bleeding edge of the most current technologies available (for an overview of some of the problems see, e.g. [7]). CMIP5 produced a sum of about 100 PB world-wide [8], produced in about 30 centres [9]. As the name of the project suggests (Coupled Model Inter-*comparison* Project), these data need to be compared. Since the models and their data are situated at different places, they have to be transported. Or the applications that compare the data have to be available near to the data – unfortunately practical experience shows that it is much easier to organise data near to applications than applications near to data, e.g., see the results of the German C3-Grid initiative [10].

The climate modelling community agreed upon sub-setting the data to a volume of about 1.5 PB, containing only those

Table I: TIME TO TRANSPORT 1 TB AND 1 PB OF DATA FROM ONE CLIMATE CENTER TO ANOTHER [11]

Transfer Rate	*Time to Transport Data of Size* 1 TB	1 PB
10 Mbps	9.7 days	27.20 years
50 Mbps	1.94 days	5.44 years
100 Mbps	23.3 hours	2.72 years
1 Gbps	2.28 hours	97.1 days
10 Gbps	13.65 minutes	9.7 days
100 Gbps	81.9 seconds	23.3 hours

Table II: CMIP6 CENTRES

Type	*EU*	*America*	*Asia*	*Australia*
PRODUCING	10	6	6	1
PROVIDING	2	3	3	1
DOWNLOADING	100s	10s	100s	10s

data most relevant for the comparison, and to make these data available in five centers world-wide for easier access and replication [4]. But, as experience shows, the process to do so took much more effort and time than expected, and stressed the scientific community considerably, leaving less time to creative scientific work that potentially would benefit the scientific value of the assessment report. Apart from a lot of hassle in the upper layers (co-ordination, federation, meta-data-systems, applications, formats) it was obvious that the network posed a crucial problem here. Not only is it error prone and unreliable, but just simply much too slow in many instances. ESnet says: "The fastest we could hope to move only 1 PB of data from PCMDI to one of the RCA data centers is essentially one day at 100 Gbps, whereas with a peak of 10 Gbps, it would take almost 1.5 weeks." An estimation of the speeds following the ESnet can be found in Table I, whereas the last row – the 100 Gbps – are not reached yet, but only addressed in the ANA100G project [3]. The fact that many network lines outward bound from centers seem rather underutilised does not contradict this observation: Burst-wise utilisation is commonplace, people try to get their job done, but the unreliability of the connections and the fact that the slowest part of the complete connection limits the transfer speed, make life of the users difficult: Maintaining constantly high data transfer speeds is near to impossible today.

If we interpret current negotiations about CMIP6, the future edition of CMIP, correctly, it can be expected that the data volumes will be 1 to 2 orders of magnitude higher than in CMIP5, with the intercontinental network speeds staying about the same (see Table II). More participating centres in Asia and South America will put higher demands on the network architecture in terms of geographical coverage and network quality. With respect to the architecture of the application layer it can be expected that the available system (ESGF) will be stabilized and possibly extended by a federated file system. The situation for the CMIP5 data: Until now the scientists have to search through a data jungle by clicking through web portals, looking at folders on different servers or using scripts. With neither of these methods all data can be accessed. E.g., the script bundle called synchro-data [12] can access only the data available with the new ESGF login method, not with the old one, and requires a special port which is then locked. Two users at one time cannot download from the same machine at the same time. Again: many networks seem to be underutilised, but not because the scientists do not need their data, but because retrieving them is intransparent. The scientists want to know how to get the data and how long the transfer takes.

A totally different, but also very demanding application for the network is state-of-the-art turbine development as it is performed at DLR (*German Aerospace Centre*). It requires a multitude of different process chains to be completed. Such process chains typically consist of different simulation tools such as Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM) solvers, which are executed in a specific collaborative order. The data is needed "just in time": At DLR different clusters of different sizes and configurations are available at geographically distributed locations. Optimal resource usage implies high flexibility in where to run jobs. In order to avoid necessity of moving data to a selected resource in order to be able to run a job it is desirable to provide reliable and fast access to all data from all different resources and locations. This is not "Big Data" application but it urges the network to be prioritized.

A recurrent task here is the simulation of flow response to different Eigenmodes and phase angle combinations. An initial steady state CFD simulation of, e.g., a turbine runner blade passage is done to obtain boundary conditions for a subsequent CSM simulation, which results in the m Eigenmodes of the respective blade. Now for each Eigenmode a specific set of n relevant phase angles is identified. In the next step corresponding displacements are applied to the blade mesh resulting in $n \times m$ different CFD simulation setups that have to be solved. These simulations run for a fixed iteration count after which convergence analysis is performed. Based on the result of this analysis for each job a decision is made whether they need to run for further iterations or not.

Single simulation jobs hereby typically run on 32-64 cores and produce result files in the range of 100 MB written at the end of the simulation. Considering a real world setup with $n \times m = 300$ leads to a relatively moderate data volume of 30 GB. Ideally all jobs can be run at the same time, thus requiring $300 \times 64 = 19.200$ cores.

Therefore, the data replication mechanisms of the General Parallel File System (GPFS) are applied, to simultaneously replicate data to all clustered resources whenever write access to the filesystem occurs.

Now, looking at the ideal but none the less likely situation where 300 simulation jobs start at the same time and all writing their results within a time frame off 15 minutes quickly leads to peak bandwidth requirements up to 400 GB. In the case of less regularity in the workflow, where potentially all jobs are started at different points in time, write access might occur

over a time frame that corresponds to the duration of the overall workflow. In this case a much lower but sustained bandwidth can be observed.

To enable an application adapted virtual network configuration the respective applications need to have an interface communicating their requirements with regard to the available network resources.

Thus, we have the climate application which needs high bandwidth for a long time and can manage interruptions, and the turbine application which needs prioritization to receive the data as fast as possible. These two applications do not cumber each other and are a good example for challenging our SDN architecture.

IV. Related work

Approaches to provide automated configuration of network services already exist for some time. Protocols for traffic engineering were specified to enable dedicated resource allocation to different applications. Most of these solutions were limited to a single carrier domain, in many cases to a single equipment vendor. Vertical interaction was achieved, i.e., communication between different layers, is well defined. Automation of configuration across domain borders was not considered. All control interaction had its origin in operator actions. The applications itself did not have any direct influence. In this section, we give an overview over the development starting with an information model and first attempts of implementation.

Within the ITU-T recommendation G.805 03/2000 [13] the authors abstract from the actual network elements. The recommendation describes a set of functional elements instead and the relationship between these elements. It is a generic multi layer information model, which is open to any kind of implementation. The notion of a layer in G.805 does not necessarily coincide with one layer of the OSI model. A layer can best be described as a set of all connection points of the same type, i.e., sources and sinks of data that can communicate without adaptation. Adaptation allows for communication between the layers. The information model developed in G.805 is the basis for any multi-layer interaction model of the future, the actual communication processes, both vertical and horizontal can be described based on this agnostic approach. Networks described in G.805 are connection oriented whereas the follow-up, ITU-T recommendation G.809 03/2003 [14], describes connectionless networks. However, both have in common an implementation agnostic information model. Communication between layers in both recommendations is enabled by adaptation functions.

Generalized Multi-Protocol Label Switching (GMPLS) [15] is a generalization of IP/MPLS for the connection oriented transport layer. It was originally defined to provide a control plane for Synchronous Digital Hierarchy (SDH), Optical Transport Hierarchy (OTH) and Wavelength-Division Multiplexing (WDM). The network elements are equipped with a GMPLS Routing Engine (GMRE) which made dynamic configuration and automated restoration possible. In a first iteration there was no interaction with layers above transport. With the definition of a northbound User Network Interface (UNI), communication with higher layers was enabled. In consequence, there were means to adapt transport bandwidth to the requirements of upper layers. Requirements from applications are not automatically considered. They still rely on operator intervention. Generalized MPLS is a method to do multi layer provisioning and traffic engineering, but it is normally restricted to one carrier and often to a single-vendor domain. Hence, it is sufficient for vertical integration but not for horizontal configuration purposes. Furthermore, flapping due to changing IP address spaces can be a problem. In the course of the VIOLA project [16] UNI-Client and UNI-Server interworking was implemented between the transport layer and an IP/MPLS layer. This way bandwidth requirement from the IP/MPLS layer could be communicated to the transport layer.

In 2008 a Multi-layer Network Model based on the ITU-T recommendation G.805 was published by Freek Dijkstra et al. [17], containing a proposal for a multiple layer control interaction model. Making use of the functional elements defined in G.805 it is possible to implement a multi layer data model. From GMPLS the technique of label switching was taken. The translation of client or application requirements still remains with the operator.

The common Network Information Service Schema Specification (cNIS) activities [18] by Geant2 community are targeted at supplying domain related network information to the application layer regardless of the network layers present in the respective domains. Inter domain exchange of information is part of the service. The cNIS activities are vital for the NSI definition.

The ITU-T recommendations G.805 and G.809 as well as the multi-layer network model from Dijkstra abstract from hardware related description of transport networks. GMPLS defines cross network layer interworking and cNIS finally makes the networking layer transparent for the application layer. The missing link is a control interface between the application and the network, enabling fully automated network resource management. Furthermore, this interface should not only address network resources but should take into account other virtualized functions. Our integrated approach will address both, network and other resources as building blocks of a final functional graph.

V. Integrated SDN Architecture

To provide application-oriented network services, we suggest an architecture consisting of three layers, depicted in Figure 2. The infrastructure layer defined by the network providers and hardware vendors is usually characterized by a vast heterogeneity. It lays at the bottom of our architecture. The control layer in the middle abstracts from the infrastructure layer and prevents direct user access to the hardware. At the top level sits the application layer representing the users view on this network architecture.

Interoperability between these layers enables an application- and user-oriented network infrastructure. To achieve this, additional communication protocols have to be specified providing

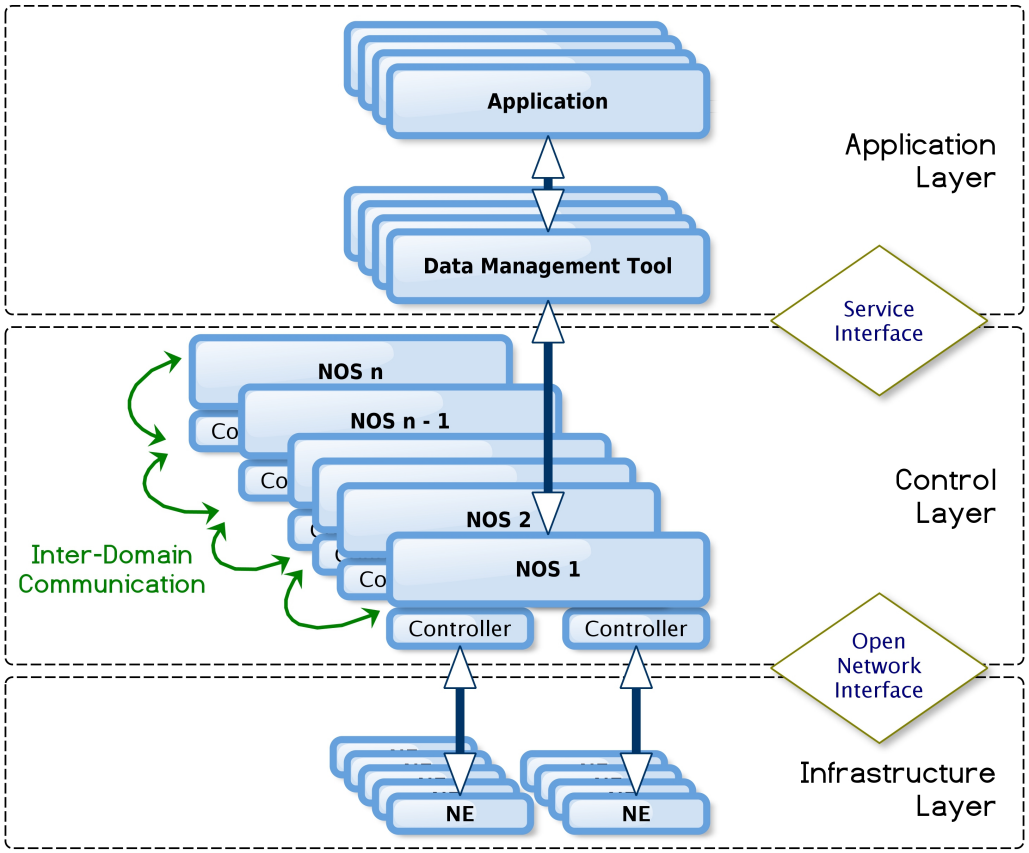


Figure 2: Integrated multi layer, multi domain SDN architecture

the required functionality. Therefore, we introduce an Open Network Interface (ONI) between infrastructure and control layer as well as a Service Interface (SI) with an appropriate connection service protocol between control and application layer.

In the following, we describe the layers and the intermediate communication protocols in more detail and give an example of a communication process within this architecture.

A. Layer description

In the following sections we describe the three layers of the introduced architecture. The application layer representing the users view, the control layer as intermediary between application and network hardware, and the infrastructure layer, consisting of the network elements and their interconnects.

1) Application Layer: The requirements of both applications and predefined services are not just network resources. We can think of storage, compute capacity and additional functionality related to the network. This of course makes the general model more complex to construct, but it takes strain from both, user and carrier. The main feature of this fully integrated model will be access to a building block repository [19]. Here we have infrastructure building blocks and functional building blocks. Infrastructure building blocks on the one hand, are network segments, covering different layers and different domains. Functional building blocks on the other hand, represent network services, e.g., encryption, compression or acceleration. An application link between Lawrence Livermore National Laboratory and German Data Centre for Climate Research involves for example multiple layers and multiple network domains. The application queries for the link and the extended SDN-enabled network protocol combines the required infrastructure building blocks to form a virtual network. Furthermore, the application requires additional services, like WAN acceleration, storage and a tool for ensuring data integrity. Depending on availability, the appropriate building blocks are added to the network graph. A real-time multi-site application like TV production involves multiple layers and possibly multiple domains. It requires a highly elastic network configuration, extremely low latency and high peak bandwidth. Data integrity must be guaranteed and synchronization must be provided. WAN acceleration would impose too much latency for a real time life production. Selected building blocks would again be network segments to form a virtual network as required, plus a tool to guarantee data integrity and synchronization functionality. Genome Sequencing to support surgeons requires high bandwidth for medium periods on short notice. While handling medical data a high level of privacy must be maintained. Again we have infrastructure building blocks in form of network segments

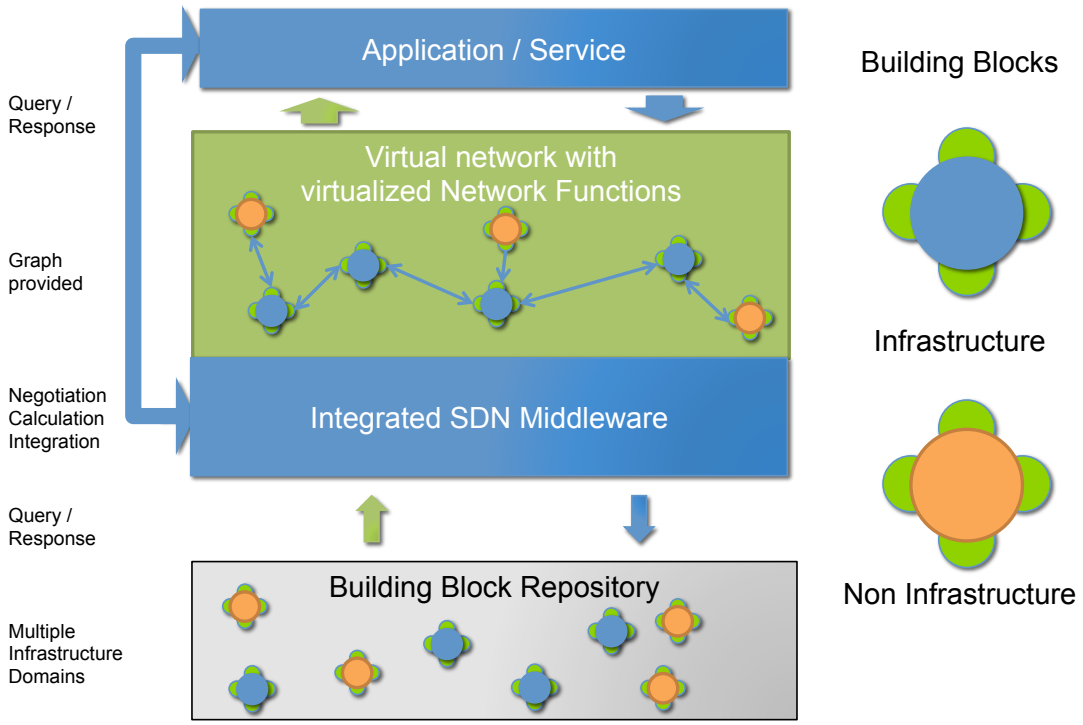


Figure 3: Functional model of an application-driven network architecture

plus functions to guarantee privacy, e.g., encryption and a Public-Key Infrastructure (PKI). The public Internet today is a major medium of social interaction and it is of utmost importance for an open society to guarantee equal rights and secure access to its resources. The provider would have to select suitable network segments to commission an Internet platform plus additional building blocks to guarantee privacy and security for the individual user.

Figure 3 shows our functional building block model to realize the work flow for the described application scenarios. Applications communicate their certain requirements towards the network-building stack. This in turn checks, if available resources satisfy these requirements and answers with a proposal of a combined graph.

2) Control Layer: Our control layer, depicted in Figure 4, consists of two main components – the Network Operating System (NOS) on the application side and one or more Controllers on the side of the infrastructure. The communication between the upper and lower layer is realized by a north- and southbound API. Additionally, the NOS module within the control layer needs an interface for inter-domain communication to enable multi-domain interoperability.

The NOS we introduce operates similar to typical operating systems. Within its domain it interacts as an intermediary between applications and network hardware, to avoid direct access to the network hardware and to hide unnecessary information. This increases the security on the one hand and enables the possibility to virtualize the network on the other hand. Therefore, the integrated Broker compares the requirements – transmitted through the northbound API – with the available network resources – which can be requested through the southbound API – and instructs the reservation if available resources meet the requirements. Negotiation between application layer and network layer should be possible. The requesting application receives only a partial graph, which can be a direct link with the corresponding characteristics between ingress and egress at the end.

Besides the virtualization our approach also takes traffic engineering into account. Link state information can be updated periodically or requested on demand via the southbound API. This way, weighted graphs are composed for the entire domain, in which the weights can represent any link parameter – like bandwidth, latency, utilization or costs – or any combination of them. Based on these graphs the route is optimized w.r.t. the requirements of the applications. Link parameters should not change during transmission. However, if a change is inevitable, the network resources dedicated to a certain application should be adapted within feasible bounds.

Real time communication is another feature which can be implemented within this network architecture. Especially with respect to large data volumes the transfer completion time is often more important than the entire transfer time. Knowing this point in time allows more efficient resource planning which can result in reduction of costs. This functionality is enabled by allowing reservations of network resources for specific periods of time. The reservations for specific flows are managed by the Broker, which has a global view on the entire domain. Thereby, overcommitment can be avoided and start and end points of the data transfer can be guaranteed.

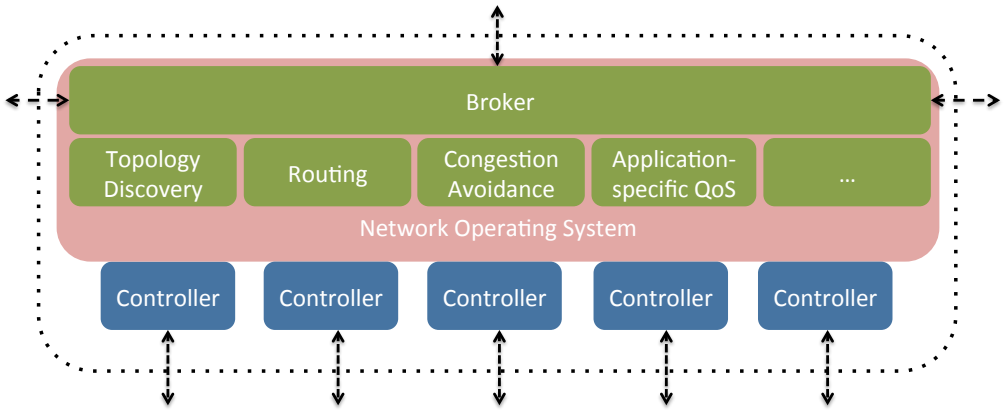


Figure 4: Single domain control layer overview

All described features are necessary to transfer the amounts of data described in Section V-A1 efficiently through a shared multi-user network. Many different algorithms and other features exist, which can be realized by this architecture. So, we encapsulate these functionalities in different modules, which can be added, replaced or removed during runtime without interruption, similar to loadable kernel modules. Thereby, every domain can optimize its control layer for its requirements as long as the interoperability is still guaranteed.

To enable the described functionality between multiple domains, an inter-domain communication is required. Those incoming and outgoing requests are also handled by the Broker and will be processed similar to intra-domain requests. Therefore, external and internal request messages may only differ in the source tag which determines the security group classification and the consequential permissions of the service requestor.

Beside the loadable kernel modules and the Broker we suggest to encapsulate the Controller as executive unit. Controllers implement the interface to the network infrastructure and perform requests or reservations, instructed by the Broker. Since this can result in a bottleneck depending on the number of requests and the domain size, we recommend to use more than one Controller. Thereby, the separation from the NOS enables scalability by varying the number of Controllers depending on the network size and work load. An additional aspect which motivates for separation of NOS and Controller are the heterogeneous interfaces we expect to be provided by the network hardware vendors. To enable compatibility, at least one Controller for every network interface implementation has to be provided. The integration is mainly realized by the hardware vendors, similar to hardware drivers in conventional operating systems. Hence, the encapsulation of the Controllers guarantees scalability and interoperability in our proposed architecture.

In summary, the control layer we introduce provides required functionalities – like network virtualization, adaptive routing or real time communication – for the application layer, to enable an efficient transfer of big data volumes. The layer is highly customizable by integrating the functionality within loadable kernel modules which can be added, substituted or removed on demand. Additionally, we took into account scalability and compatibility of an heterogenous infrastructure by encapsulating the Controllers as executive units.

3) Network Layer: In data networks there is a hierarchy of deterministic transport and statistical multiplexing. Deterministic transport can be utilized for client-to-client communication and as a transport layer for, e.g., routed services. The Broker instance shown in Figure 5 arbitrates between the requirements of multiple applications and available network services. Based on requirements communicated by the Network Service Agent (NSA), it will decide if the requested capacity will be provided on a deterministic or routed path.
Multi domain networks suffer from a lack of homogeneity. This in turn requires abstraction that allows for a unified network description language. The Network Description Language (NDL), introduced in [20], is a modular set of schemata. The topology schema describes devices and interactions between them on a single layer. The layer schema takes into account the existence of multiple layers and interactions between these layers. Capabilities of network devices are described in the capability schema and domain schemata have to deal with different domains and in consequence with administrative entities and services linked to these entities. Finally, the physical schema describes the physical aspects of network elements. This set of schemata defines the ontology of network functionality.
Since most applications rely on resources from different domains, information about services and capabilities of these domains will have to be interpreted and coordinated. An application and its related data management is attached to a single domain. All information from external domains should be gathered here and communicated to the data manager to enable negotiation.

B. Communication Interfaces

Interoperability between the layers introduced in Section V-A requires information exchange. Therefore, interfaces

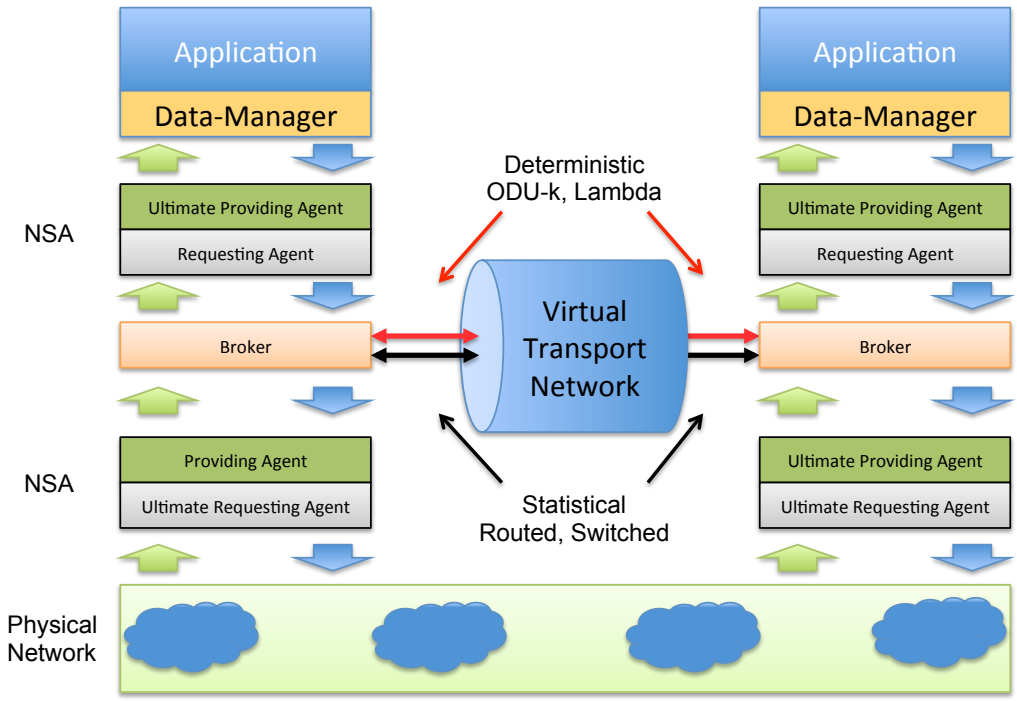


Figure 5: Deterministic and statistically multiplexed transport

Table III: SERVICE INTERFACE PRIMITIVES

Primitive	*Description*
RESERVE	The **requesting agent** (RA) requests the **providing agent** (PA) to reserve network resources
PROVISION	The RA requests the PA to provision network resources according to the previous reserve request. Depending on actually available resources the provision request may differ from the reserve request.
RELEASE	The RA requests the PA to de-provision resources without removing the reservation
ACTIVATE	The RA requests the PA to activate provisioned resources
TERMINATE	The RA request the PA to release provisioned resources and terminate the reservation
FORCED END	PA notifies RA that a reservation has been terminated
QUERY	Can be used as a status polling mechanism between RA and PA

have to be defined, which enable communication in both directions. To achieve compatibility, open interface standards are preferred. The following sections describe general requirements for service and network interfaces.

1) Open Service Interface (OSI): The northbound interface of the control layer communicates with the application layer, the southbound interface with the network layer. Since there is a multitude of network domains, horizontal communication is mandatory to enable federated network services based on a virtual multi domain network. Therefore, both application and control layer, implement embedded Network Service Agents (NSA) which are connected by a service interface. The application NSA is called requesting, the control layer NSA providing agent. Multiple services can be handled by a single NSA, in fact, as many as there are available on the end to end infrastructure. The requesting agent communicates only with the local NOS, information from other network domains is gathered and provided by the remote home domain NOS.

Because the NSA has no authority about local or remote resources, any kind of resource management is realized by the NOS in conjunction with the controller. Flexibility regarding to the introduction of new network services is enabled by the modularity of the OSI and NSA concept.

The OSI connection protocol communicates requirements to the providing agent and consists of 6 primitives, listed in Table III. These requirements have to be mapped on the corresponding QoS properties – sustained bandwidth, latency and maximum latency variation. Furthermore, the dedicated instance of time a certain transmission should start is communicated. The providing agent either answers with a complete confirmation or starts negotiating with the requesting agent. Once a service is confirmed there will be no further negotiations or limitations.

2) Open Network Interface (ONI): Current network elements implement control and forwarding plane on the same closed platform. Decoupling this control functionalities from the infrastructure, requires a protocol to exchange information between these two layers. This section describes the functions, required to implement the features described in Section V-A2.

An efficient placement of data flows requires a global view on the underlaying infrastructure. Therefore, the position of all network elements within a domain and their connection between themselves has to be announced to the control layer.

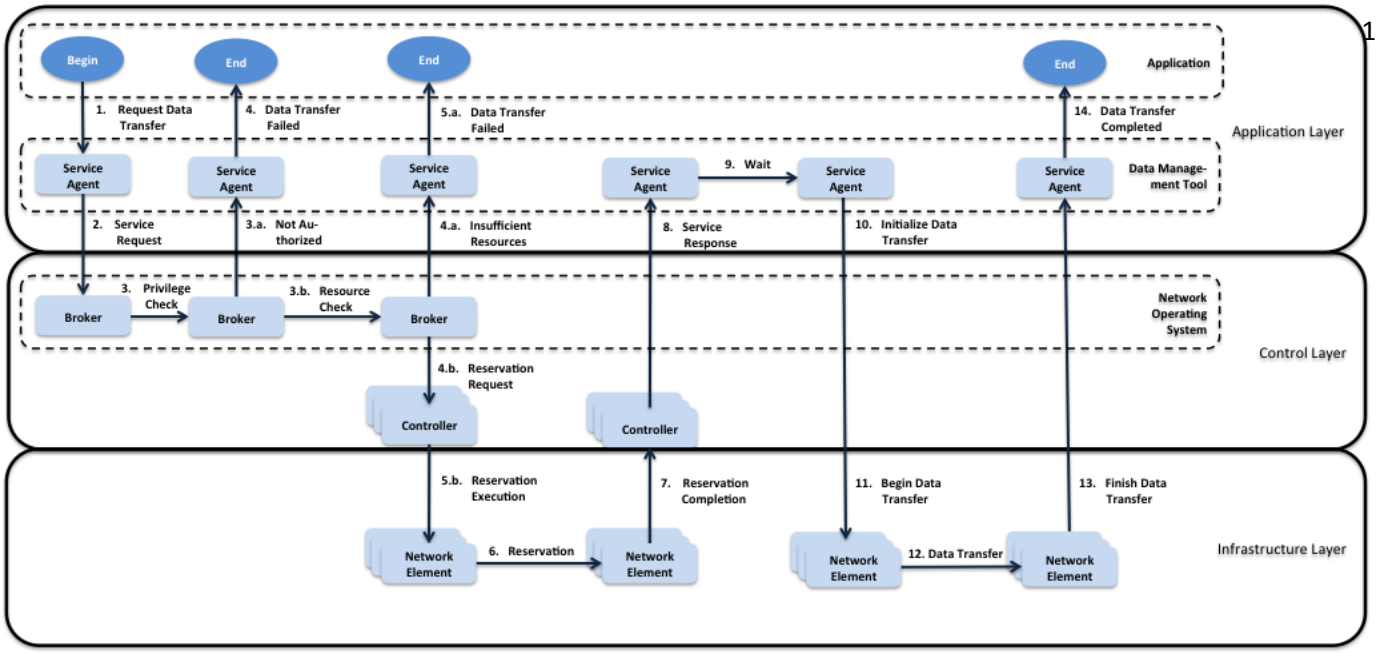


Figure 6: Integrated SDN communication process negotiating for network resources

This can be implemented by an initialization message during the startup of a network element and a link discovery protocol like LLDP [21]. Once a network element and its connections is known, state changes are noticed implicitly as soon as a data flow can not be routed anymore. In this case, the link is removed from the topology graph until the node is back and sends the initialization message.

Once the underlaying network topology is identified, link characteristics like bandwidth, latency or cost have to be communicated to the control layer during the initialization phase. These mostly static properties are stored together with source and destination of a link and are used to build a weighted graph for data transfers if requested.

Next to these properties, there are more varying link state informations like utilization, message rate or number of flows. Updating these values on every change would cause an immense overhead. Therefore, these informations are requested periodically by the Controller and only reported to the NOS as soon as values exceed predefined thresholds. The controller can request these informations explicitly by a message, or implicitly as soon as a data transfer is completed.

Additionally, to the upward directed information flow the ONI has to implement the reservation requests from the Control Layer to the network elements. These reservation requests can be combined with a period of time during which they are valid. If the reservation observance can be handled by the network elements only, the requests have to be transmitted. If not, the control layer has to add the reservation at the beginning and remove it at the end. This causes more overhead, but leaves the control function within the dedicated layer.

As described, the main objective of the ONI is to provide information about the infrastructure for the control layer to enable efficient traffic engineering. To reduce the emerging overhead, information should be updated implicitly on occurring events which already require a communication. Additionally, the executive commands instructed by the control layer have to be transmitted to the network elements by the ONI.

C. Communication process

Specified requirements for data transfers can not be satisfied in any case, e.g., if the request exceeds available capacities. To ensure a data transfer anyway and independent from the current utilization, we recommend to partition the available capacity. One part for best-effort transfers, the other one for optimized SDN communications. This way, rejected data transfer requests can use conventional communication protocols. Also for small data sets the best-effort transfer might be the better path. Since the conventional best-effort communication is known, we confine the description in this section to the optimized SDN communication.

Figure 6 depicts the chronological sequence of a demand-oriented communication within the introduced SDN architecture. Thereby, the application communicates its requirements to the data management tool first. The integrated service agent determines the network services which are required to satisfy the request. Subsequently, the agent can apply for these services by forwarding the request to the Broker of the Control Layer.

As described in Section V-A2, the Broker verifies incoming requests. If the requestor is not authorized to use these services or if there are not enough resources to fulfill the request, the transfer fails and the application is informed by the service agent. At this point, a new request with different requirements

can be initiated. This process can be repeated until both sides accept the conditions. The negotiation phase can also be implemented transparent to applications within the data management tool. This way the application defines tolerable ranges for the requested network parameters instead of single values. If both sides can not agree on a parameter set, the application has to transfer the data by using the conventional best-effort path.

If the request is valid the Broker initializes the reservation process and instructs all required Controllers to distribute the reservation to all participating network elements. Once all reservation confirmations arrived at the Controller, the Service Agent can be informed about the conditions of the requested transfer. At the communicated start point the data transfer can be initialized and accomplished. From the application's point of view, the following transfer does not differ from the conventional communication process, except that the infrastructure behaves like negotiated in the initialization phase.

As Figure 6 and the description of the communication process show, the overhead increases due to the initialization phase. Therefore, the optimized data path is only recommended for elephant flows, where the transfer time is much higher than the startup time. In this case, the overhead to define an optimized environment is worthwhile. However, small flows may still use the conventional data path.

VI. Migration from Existing Architectures

Migration towards an application driven network configuration has to be done in a stepwise approach. In a first step, the network elements have to be enabled to support a common controller language. For network elements in use today, there will have to be a translation overlay. Controller-input at that stage will not be automated and it will be per domain or even per network element group.

In a second step, the Network Operating System (NOS) has to be defined for providing input to the controllers. First, only single domain interworking and bidirectional communication between domain NOS and domain controller will be supported. Based this horizontal integration of involved NOS can be implemented for ensuring interoperability between multiple domains.

The next milestone is to enable applications to communicate with the respective NOS. Thereby, the user has to know about the requirements and communicates them directly to the NOS. Later, a fully automated negotiation process can be initiated by the application.

In the final step, the NOS will be enabled to request and integrate required functional building blocks, additional to the requested network resources.

VII. Bandwidth-On-Demand Prototype

Global data traffic increases steadily by developments in Cloud Computing, Social Media and Big Data applications. According to projections, 2015 the core infrastructure of the Internet has to handle four times as much data than 2010 [22]. Therefore, extremely high bandwidth data networks are required, which was the reason for the two testbeds in Germany, described in the following.

That federated applications and services can profit from increasing the bandwidth was already proven within the 100-Gigabit-Testbed on a 60 km Wide Area Network between the *Center for Information Services and High Performance Computing* (ZIH) of the Technical University Dresden and the computing center of the *Technical University Bergakademie Freiberg*, started in 2010 [23].

A follow-up testbed which also introduced our first SDN prototype was presented on the ISC'13 [24], based on a 400-Gigabit/s-Demonstrator between the *Center for Information Services and High Performance Computing* (ZIH) in Dresden and the *Rechenzentrum Garching* (RZG). For the next generation 400-Gigabit/s-Ethernet technology not only the bandwidth but also the distance was increased. Therefore, the HPC centers in Dresden and Garching (Munich) were connected by a 640 km dark fiber, provided by Deutsche Telekom, combined with access and transport technology from Alcatel-Lucent. Cluster systems from Bull in Dresden and IBM in Garching were used to set up a General Parallel File System (GPFS) to give applications a consolidated view on distributed data. To achieve the necessary I/O throughput the cluster nodes contained high-speed PCIe RealSSD flash cards from EMC[2] with 3.2 GByte/s sequential read and 1.9 GByte/s sequential write performance. With three of these cards per server we achieved a theoretical peak read/write performance of 921.6/547.2 Gbit/s in total, which was sufficient to saturate the 400-Gigabit/s-link. The bandwidth to the WAN was ensured by 40-Gigabit/s-Ethernet cards from Mellanox, which were directly connected to the service router from Alcatel-Lucent. Additionally, the HPC clusters in Dresden and Garching had to be integrated into the testbed. Therefore, FDR InfiniBand cards from Mellanox were deployed. So the clusters on both sides were only used to direct the traffic from the location where the data was stored, to the computing nodes in the HPC centers. The entire architecture of the 400-Gigabit/s-Testbed is also shown in Figure 7.

Within the 400-Gbit/s-Testbed we have demonstrated the impact of different applications utilizing the same infrastructure, with and without interoperability between application and network infrastructure. On one hand, the turbine simulation, described in Section III, requires a parallel file system and distributed calculation. Therefore, low latency and high bandwidth elasticity are required to achieve a high performance application layer. On the other hand, the climate application scenario, also described in Section III, requires the transfer of huge geographically dispersed datasets for intercomparison. Consequently, a very high sustained bandwidth is required, to guarantee a reliable and predictable data transfer. To specify the different demands of these applications, we implemented a web-frontend, which forwarded incoming reservation requests to a centralized management system. This system evaluated incoming requests and instructed the controllers in Dresden and Garching to configure all participating network elements. Because there was no mechanism available to remove the

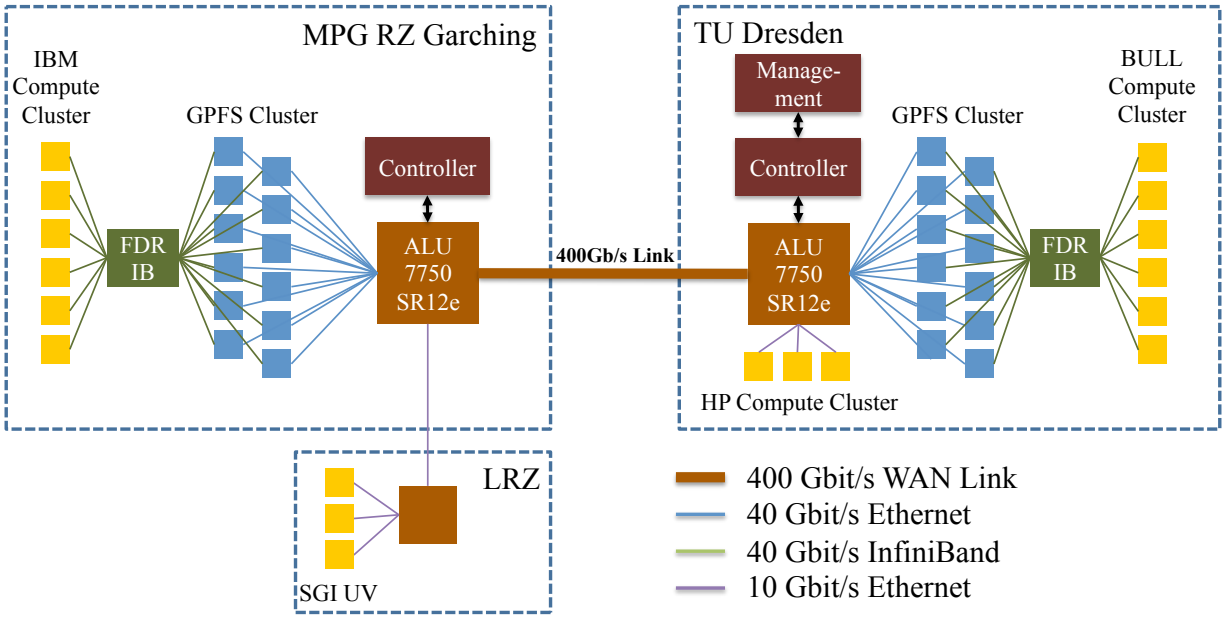


Figure 7: Overview 400-Gigabit/s-Testbed architecture

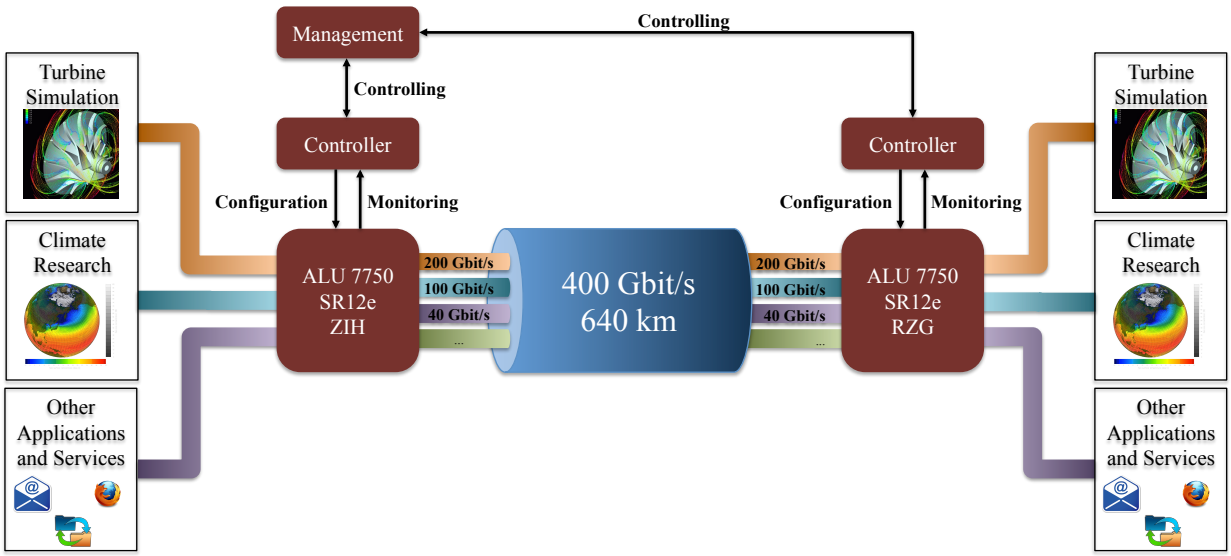


Figure 8: Setup of the Bandwidth-on-Demand Demonstrator

reservations automatically after expiration, the centralized management system also included a time table and scheduler for all reservations. Based on the entries in this time table, reservations were declined if conflicts occurred and removed after expiration. Figure 8 shows the entire setup of the Bandwidth-on-Demand demonstrator.

In this testbed, we were able to demonstrate that all applications can benefit from this new control layer, providing an interface between application and infrastructure layer. Especially the predictability of long term data transfers was increased tremendously, independent of the concurrent traffic on the link. Also, providing dedicated bandwidth for the turbine simulation increased the performance. Unfortunately, the latency could not be influenced, due to topology limitations. So, there is still potential for more optimization.

VIII. Conclusion

Our integrated SDN architecture enables concurrent applications competing for network resources, to define virtual networks that satisfy their respective requirements providing efficient network usage and reliable data transfers. We introduced the elements necessary for an end-to-end negotiation of network resources between multiple domains and without any limitation to specific protocols.

On the top the application layer represents the users view on this network architecture. A southbound Network Service

Agent (NSA) requesting resources from the underlying control layer. Communication between the NSAs is realized by the Open Service Interface (OSI). The providing NSA in turn is handing over the request to the Network Operating System (NOS), which links the network layer of its own domain with NOS's of other domains. The NOS has a centralized view on the network resources available and outstanding requests from applications, so it is able to arbitrate between them. Scalability and compatibility is enabled by using different Controllers, depending on the work load and the underlaying infrastructure. Thereby, our architecture supports end-to-end negotiation of network resources between multiple domains and without limitation to a specific protocol.

Additional to the architecture description we show up a feasible approach to migrate from existing traditional network architectures to an application driven network architecture. Furthermore, we were able to demonstrate the benefits of our approach for applications within a 400-Gigabit/s-demonstrator, connecting two High Performance Computing centers in Germany, by a prototypical implementation.

REFERENCES

[1] A. Georgi, R. Budich, Y. Meeres, R. Sperber, and H. Hérenger, "An Integrated SDN Architecture for Applications Relying on Huge, Geographically Dispersed Datasets," ser. INFOCOMP. IARIA, 2013, pp. 129–134.

[2] Alcatel-Lucent, "Alcatel-lucent upgrades cable system linking japan and california," January 21 2013, retrieved April 26th, 2013, from http://www3.alcatel-lucent.com.

[3] Energy Sciences Network ESnet, "ESnet Partners with North American, European Research Networks in Pilot to Create First 100 Gbps Research Link Across Atlantic," April 24 2013, retrieved April 26th, 2013, from http://esnetupdates.wordpress.com.

[4] "CLIVAR Exchanges - Special Issue: WCRP Coupled Model Intercomparison Project - Phase 5 - CMIP5," Project Report, May 2011. [Online]. Available: http://eprints.soton.ac.uk/194679/

[5] "CMIP5 Coupled Model Intercomparison Project," retrieved May 26th, 2014, from http://cmip-pcmdi.llnl.gov/cmip5.

[6] Top500, "Top 500 Supercomputer Sites," http://www.top500.org/, 2013.

[7] N. Hemsoth, "20 Lessons Enterprise CIOs Can Learn from Supercomputing," *Datanami*, November 2012, http://www.datanami.com/datanami/2012-11-12/20_lessons_enterprise_big_data_buffs_can_learn_from_supercomputing.html.

[8] "Cmip5Status," retrieved May 26th, 2014, from https://github.com/ESGF/esgf.github.io/wiki/Cmip5Status.

[9] "Modeling Groups and their Terms of Use," retrieved May 26th, 2014, from http://cmip-pcmdi.llnl.gov/cmip5/docs/CMIP5_modeling_groups.pdf.

[10] C. Grimme and A. Papaspyrou, "Cooperative Negotiation and Scheduling of Scientific Workflows in the Collaborative Climate Community Data and Processing Grid," *Future Generation Computer Systems*, vol. 25, pp. 301–307, 2009, publication status: Published.

[11] Energy Sciences Network ESnet, "BER Science Network Requirements; Report of the Biological and Environmental Research," Network Requirements Workshop, LBNL report LBNL-4089E, April 29-30 2010.

[12] J. Raciazek, "Synchro-data script bundle," Technical documentation, retrieved May 26th, 2014, from http://dods.ipsl.jussieu.fr/jripsl/synchro_data.

[13] "ITU-T Recommendation G.805: Generic functional architecture of transport networks," International Telecommunication Union, Tech. Rep., Mar. 2000. [Online]. Available: http://www.itu.int/rec/T-REC-G.805/en

[14] "ITU-T Recommendation G.809: Functional architecture of connectionless layer networks," Mar. 2003. [Online]. Available: http://www.itu.int/rec/T-REC-G.809/en

[15] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945 (Proposed Standard), Internet Engineering Task Force, October 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3945.txt

[16] P. Kaufmann, *Gesamtdarstellung des VIOLA-Projektes (Vertically integrated optical testbed for large applications in DFN)*. DFN-Verein, 2007. [Online]. Available: http://books.google.de/books?id=9ZelPgAACAAJ

[17] F. Dijkstra, B. Andree, K. Koymans, J. van der Ham, P. Grosso, and C. de Laat, "A Multi-layer Network Model Based on ITU-T G.805," *Comput. Netw.*, vol. 52, no. 10, pp. 1927–1937, Jul. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2008.02.013

[18] M. Labedzki, C. Mazurek, A. Patil, and M. Wolski, "Common Network Information Service - modelling and interacting with a real life network," 2009.

[19] D. Schwerdel, D. Günther, M. R. Khondoker, B. Reuther, and P. Müller, "A building block interaction model for flexible future internet architectures," in *NGI*. IEEE, 2011, pp. 1–8.

[20] P. Grosso, A. Brown, A. Cedeyn, F. Dijkstra, J. van der Ham, A. Patil, P. Primet, M. Swany, and J. Zurawski, "Network topology descriptions in hybrid networks," March 2010.

[21] "IEEE Standard for Local and Metropolitan Area Networks– Station and Media Access Control Connectivity Discovery," *IEEE Std 802.1AB-2009 (Revision of IEEE Std 802.1AB-2005)*, pp. 1–204, 2005.

[22] C. Inc., "Cisco Visual Networking Index: Forecast and Methodology, 2010–2015," Cisco Inc., Tech. Rep., June 2011.

[23] M. Kluge, S. C. Simms, T. William, R. Henschel, A. Georgi, C. Meyer, M. S. Müller, C. A. Stewart, W. Wünsch, and W. E. Nagel, "Performance and quality of service of data and video movement over a 100 gbps testbed," *Future Generation Comp. Syst.*, vol. 29, no. 1, pp. 230–240, 2013.

[24] R. Budich, A. Georgi, J. Müller and R. Sperber, "International Supercomputing Conference 2013," Leipzig, Jun. 2013.

Formal Synthesis of Real-Time System Models in a MDE Approach

Cédrick Lelionnais,
Jérôme Delatour,
and Matthias Brun

ESEO-TRAME
Angers, FRANCE
Email: cedrick.lelionnais@eseo.fr
Email: jerome.delatour@eseo.fr
Email: matthias.brun@eseo.fr

Olivier H. Roux
and Charlotte Seidner

IRCCyN - Université de Nantes
École Centrale de Nantes
Nantes, FRANCE
Email: olivier-h.roux@irccyn.ec-nantes.fr
Email: charlotte.seidner@irccyn.ec-nantes.fr

***Abstract*—The development of real-time embedded systems is quite complex because of the wide range of execution platforms and of the importance of non-functional requirements. Furthermore, Model Driven Engineering is particularly suitable for handling the diversity of implementation targets. Therefore, several real-time embedded systems development suites leverage Model Driven Engineering by automatically generating platform-specific code from high-level design models. Such tools may also take non-functional requirements into account by integrating verification activities. These activities typically rely on the generation of formal models from the same high-level design descriptions used for code generation. However, few tool suites support both code and formal model generation. Furthermore, among these, most overlook real-time operating systems mechanisms. Therefore, both code and formal models generated by these tool suites may not behave as specified in the high-level design descriptions. The present work extends the SExPIsTools code generator tool suite with a support for the generation of formal models. The proposed strategy relies on the composition of formal model fragments described using an extension of the classical Time Petri Nets. This paper presents a formalization of this composition that generically considers the behavior of platforms. As an illustration, we then give the formal model describing the behavior of an application on two different platforms (OSEK/VDX and VxWorks) and check a safety property on both models.**

***Keywords*—Real-time operating systems, Model Driven Engineering, Time Petri Nets, Multi-platform deployment, Formal model.**

I. Introduction

Real-Time Embedded Systems (RTES) increasingly surround us in various domains (aircrafts, cars, cell phones, robotics, etc.). RTES engineers are confronted with the challenge of developing more complex, higher quality systems, with shorter development cycles at lower costs. Model Driven Engineering (MDE) helps engineers to develop tool suites that partially automate the development of RTES. Using model transformations, these tool suites mainly produce either executable code or formal models from high-level design descriptions of RTES.

Some of these tool suites have both code and formal models generation processes. However, the mechanisms of Real-Time Operating Systems (i.e., executable software platforms supporting real-time applications, RTOS) are often ignored by these generation processes. As a result, generated code and generated formal models may not behave as specified in the high-level design description. Consequently, verification and validation activities applied on the RTES development could provide erroneous results. For instance, the detection of malfunctions (e.g., wrong treatments of critical data, or bad scheduling of real-time multitasking applications) could be compromised.

Nevertheless, among these tool suites, some consider real-time aspects in their generation processes. They thus take the deployment of real-time applications on RTOS (i.e., mapping of application concepts to execution platform services in order to execute them) into account. However, none of them satisfies the four criteria given below:

- **Portability** of real-time applications to adapt to the RTOS heterogeneity;
- **Reusability** of generation processes for a rapid migration of these applications in a multi-platform deployment case;
- **Maintainability** of RTES to help all stakeholders in their interventions;
- **Correctness** of generation processes in ordrer to have confidence in RTES development.

This work is part of an overall strategy of RTES development using a MDE approach. This strategy is supported by a tool suite called SExPIsTools (for Software Execution Platform Inside Tools). In order to satisfy the criteria previously given, SExPIsTools relies on the following approach:

- **considering any RTOS** as parameter of generation processes to achieve multi-platform deployment;
- **writing more generic transformation rules** to be independent from the considered RTOS;
- **separating domain concerns** (i.e., application deployment choice, RTOS consideration, transformation rules and verification and validation activities) to clarify interventions of each domain specialist;
- **formalizing transformation rules** to increase the correctness of generation processes.

In the present paper, we focus on the latter point. We need to construct RTES formal models, i.e., models of the whole system including the RTOS. For this purpose, our approach relies on a single transformation that does not depend on any specific RTOS. This transformation composes multiple formal model fragments independently of the target RTOS. Each of these fragments represents a part of the formal model that captures the behavior of the RTES.

As a basis of the construction, we use roles to generically identify connection points. These roles are used as a glue between the formal fragments. As a consequence, a new definition is presented to represent these formal fragments based on roles. The class of models we use is Time Petri Nets (TPN). The generic construction is then formalized as a basis of the transformation rules. Finally, an application example is proposed to illustrate a multi-platform deployment case, where two RTOS are considered. As a result of this experimentation, a scheduling safety property is verified on both resulting models.

This paper extends our previous work [1] in which the fundamental rules of composition have been presented. An extension of these rules is given here to both 1) compose formal models of multitasking deployed applications with priority policy, and 2) provide a first validation of the generic construction.

Section II presents multi-platform deployment related works within a MDE approach [2]. A description of SEx-PIsTools is then given in Section III. Section IV gives the formal definition of the TPN fragments composition operator, which is based on roles. The application of this operator to the construction of whole RTES formal models is then described in Section V. Application examples are presented in Section VI. The benefits and limits of this approach are discussed in Section VII. Finally, we conclude in Section VIII.

II. Related Works

The following sub-sections present existing tool suites related to the multi-platform deployment problem.

Firstly, some code generators are introduced. Formal model generation tool suites are then presented, some of which are also capable of generating code. Finally, we will take a stand on the adopted approach.

A. Code generator tool suites

In order to promote the reuse of deployment tools within a single code generator, the genericity of processes has been at the heart of concerns. For instance, TransPI [3] relies on a two-step approach to generate specific code. A first phase considers a generic behavioral representation of the RTOS API (Application Programming Interface) in accordance with POSIX standard. In a second phase, the deployment is refined by configuring the process rules with the API of the targeted RTOS. However, the configuration of new rules does not fully satisfy the reuse of such a process since the tool must be modified.

A similar experimentation [4] improves reuse by specifying the RTOS concerned by the deployment without modification of the process. This orientation has been thought with the aim of porting real-time applications. This strategy relies on the transcription of code snippets by configuration of functions with RTOS information. Those information come from components of the targeted RTOS whose architecture was previously modeled by the generic modeling language AADL, which is dedicated to the real-time domain. The flexibility of this process meets the heterogeneous requirements of platforms.

Another approach [5] also contributes to the multi-platform deployment problem. This fills the behavioral gap in the SRM package of the MARTE UML profile [6]. Before dealing with the RTOS behavior, a transformation process was developped [7] to generate a deployed application model by considering the targeted RTOS structure described with SRM. Descriptions of executable concepts (i.e., resources and services of the API) of the targeted platform required by the application are instantiated through the deployment process. This instantiation is completed by a refinement of descriptions depending on both application data and location of elements playing a generic role (e.g., a task priority or a counting semaphore capacity) within the considered platform. The integration of the behavioral aspect is also based on this notion of role. Code snippets are assigned to execution services (e.g., a task creation or a semaphore taking) in accordance with Java or C++/POSIX implementations.

The main interest of the two last contributions is the independence of specialists during their interventions regarding the tool suite. This criterion guarantees the quality of maintainability, which is added to the already mentioned reusability and portability. Despite this, these contributions present a major drawback that is inherent to all code generators. The formalization is indeed absent from the addressed processes. This weakness prevents specialists from applying verification activities on deployed applications.

B. Formal model generation tool suites

The tool suites presented in this section encourage the behavioral formalization of RTOS.

1) Without code generator: An approach [8] has recently launched a formal synthesis for composing behavioral models of RTES. In order to achieve this, both application and platform are modeled with adequate modeling language. With the help of an Algebra of Communicating and Shared Resources (ACSR), behavior of the targeted platform is formalized. Behavior of the application is described by a Timed and Resource-oriented Statechart (TRoS) including both time annotations and resource constraints. A model of the deployed application is then composed with the obtained formal models and used for analysis. This synthesis provides a detailed design of RTES by formalizing their implementation with a complementary manner. Unfortunately, this process is complex to use, which forces stakeholders to have a good knowledge of formalization tools. Moreover, the composition requires a strong dependency of the application with respect to the platform.

Metropolis [9] supports both design and analysis of heterogeneous embedded systems on the basis of the platform-based methodology. The behavioral representation is illustrated by entities such as concurrent and communication activities. In addition, this environment offers the possibility to use formal languages in accordance with the LTL logic for verifying both

functional and non-functional properties once the deployment reached by mapping of the system components with the platform entities is described. Similarly, GME [10] includes use constraints on the representation of executable concepts thanks to the Platform Modeling Language (PML). This consideration mixed with the integration of formal languages to describe the behavior of platforms provides a deeper design of embedded systems. However, the genericity of modeling languages used for describing the platforms does not facilitate the treatment of particular domains such as real-time. Furthermore, transformation rules are not entirely clear. This leads to a less meaningful separation of domain skills and consequently to a maintainability decrease.

2) With code generator: More specifically to RTES, Ptolemy project [11] is further adapted for describing execution models. The definition of those models relies on the actor-oriented design and revolves around mechanisms of concurrency and communication implemented between RTES components. The behavior represented within those models is translatable into several execution semantics. This benefit offers a code-formal model duality of deployed applications. This approach thus achieves a wide coverage of software execution platforms. Nevertheless, the concepts of structural representation are only intended for the modeling of applications with this approach, and not for RTES themselves. Mechanisms such as RTES components synchronization must therefore be simulated with other verification tools. In spite of a very high completeness, the RTES maintainability with Ptolemy is once again called into question.

C. Positioning

In order to highlight both advantages and drawbacks of the works presented above, Table I classifies the tool suites according to their uses. Depending on whether a given tool suite addresses only code generation, only formal models generation, or both it will be in the first, second, or third column respectively. The first row is for tool suites that are not adapted to RTOS, while the second row is for tool suites that are adapted to RTOS.

TABLE I: Tool suites comparison

	Code	Formal	Code & Formal
Not adapted to RTOS		Metropolis [9] GME [10]	
Adapted to RTOS	TransPI[1] [3] snippets+AADL [4] SRM [5]	ACSR+TRoS[2] [8]	Ptolemy[3] [11]

[1] Less suitable for both reusability and maintainability
[2] Less suitable for reusability
[3] Less suitable for maintainability

Ptolemy seems to be the most versatile. Nevertheless, stakeholders distinction does not appear clearly, which does not facilitate maintainability.

Within MDE, alternative approaches were compared [12] to meet these requirements. The adopted strategy offers the possibility to capitalize most RTOS descriptions for multi-platform deployment in a generic way.

In conjunction with this objective, SExPIsTools integrates the Real-Time Embedded Platform Modeling Language (RTEPML) [13]. RTEPML was developed with the aim of representing executable platform concepts dedicated to the real-time domain. To further detail their representation, RTEPML [14] has been enriched to describe their behavior in TPN. However, the generation process used to take into account these behavioral descriptions needs to be formalized, which was started in [1], and is continued in the work presented in this paper (see Sections IV and V).

III. SExPIsTools

This section presents SExPIsTools. Firstly, the modeling language RTEPML used to describe RTOS mechanisms is presented. Then, both deployment and formal models generation processes integrated in SExPIsTools are described. The role notion used as a generic basis of transformation rules is highlighted.

A. Modeling with RTEPML

RTEPML distinguishes the RTOS structural modeling from the behavioral RTOS modeling.

1) Structural description: RTEPML is born from SRM package [7] mentioned in the previous section. SRM allows the description of a large number of RTOS [15] and had identified all concepts and their mechanisms present in RTOS. In SRM, these concepts are called resources (e.g., task, semaphore, etc.). RTEPML keeps the same taxonomy. In Figure 1, a small part of OSEK/VDX RTOS [16] and VxWorks RTOS [17] descriptions in RTEPML are given. The task concept, called schedulable resource in RTEPML, is described for both RTOS.

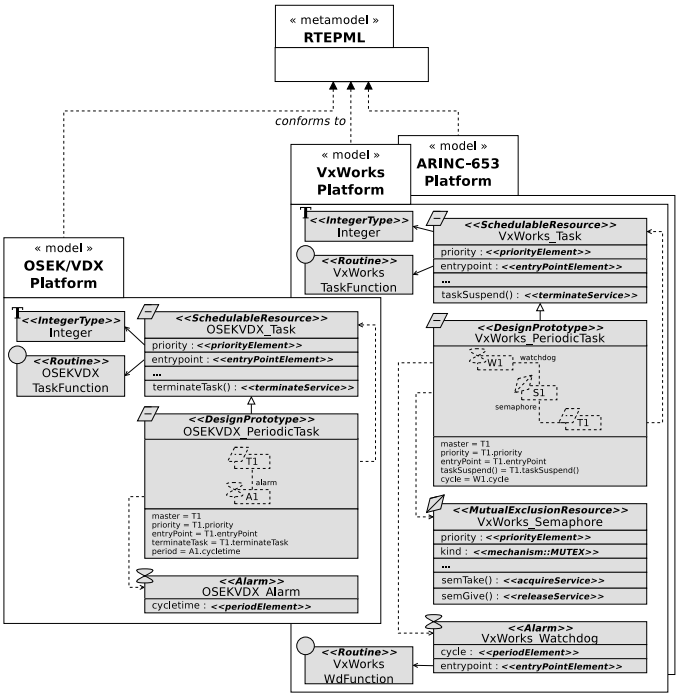


Fig. 1: Structural representation of OSEK/VDX platform

As depicted in Figure 1, thanks to the notion of roles (represented in bold between french quotation marks), we

could specify the priority (an integer) of OSEKVDX_Task or of VxWorks_Task. This priority plays the role of priority element (<<***priorityElement***>>) for both tasks. Roles are thus used to identify both structures and features of each RTOS resource. As another example, on OSEK/VDX model, the <<***terminateService***>> role characterizes the terminateService of OSEKVDX_Task in Figure 1.

Sometimes, certain concepts do inherently not exist on RTOS. As an example, the periodic task concept is missing on both OSEK/VDX and VxWorks platforms. With RTEPML, sets of concepts (i.e., identified with the *DesignPrototype* role in Figure 1) can be composed to translate this kind of concept. Thus, a PeriodicTask concept could be viewed at a composition of a Task and an Alarm for OSEK/VDX. As regards VxWorks, the PeriodicTask concept is differently composed of a Task and a Watchdog synchronizing with a Semaphore.

2) Behavioral description: The behavioral description allows to represent the life cycle of RTOS concepts. Figure 2 extends the representation of OSEK/VDX concepts, given in Figure 1. The <<***behavioralPrototype***>> role leads to the assignation of a behavioral description to each concept, including services.

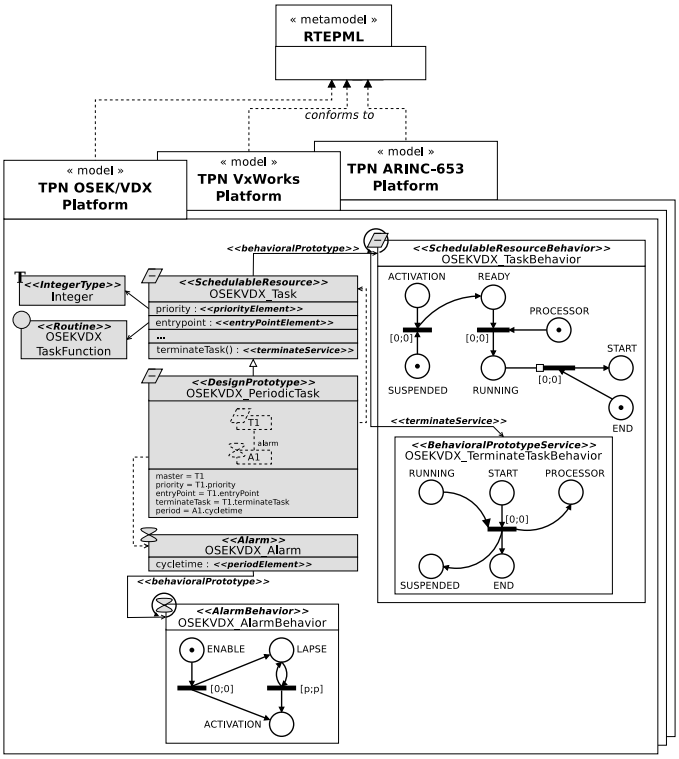


Fig. 2: Behavioral representation of OSEK/VDX platform

Each behavioral description is translated into a Time Petri Net (TPN) [18] [19] whose definition is given Section IV. This class of model is used to describe both synchronism and parallelism, as well as time evolution. Therefore, TPN are well adapted to our concerns.

In the example given in Figure 2, the TPN describing OSEKVDX_AlarmBehavior represents the periodic activation of OSEKVDX_Alarm. Informally, once a token is present in the ENABLE place, making it *marked*, one token is periodically distributed in the ACTIVATION place. Distribution of tokens is initially achieved once the left transition, represented as a black rectangle, is triggered (i.e., when ENABLE is marked and the transition clock has reached 0 time unit). The periodicity is then guaranteed by the right transition triggering (i.e., when LAPSE is marked and the transition clock has reached p time units). With clocks on transitions, we can consequently add as many time constraints as necessary, e.g., on the OSEKVDX_TaskBehavior TPN, a delay between the READY and RUNNING places could represent the required time to start the task execution.

B. Model generation processes within SExPIsTools

SExPIsTools is designed for multi-platform deployment. Both code and formal model generations are performed in two steps. Figure 3 depicts these two steps.

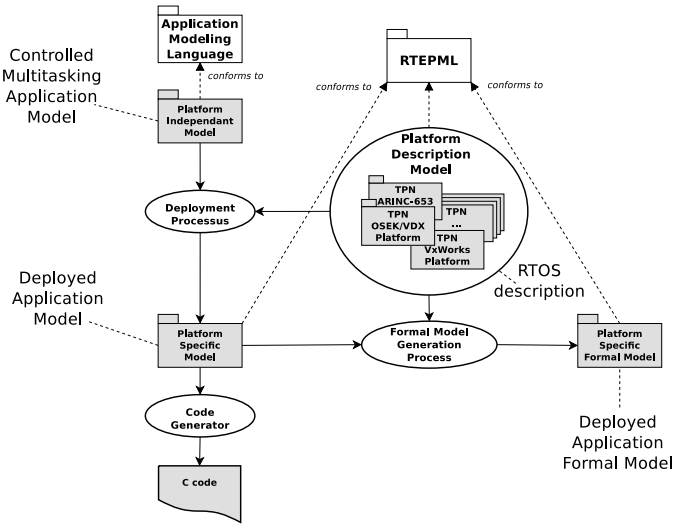


Fig. 3: Multi-platform deployment process within SExPIsTools

The first one concerns the deployment. This is common to both generations, which avoids to deploy twice. The application is deployed on a specific RTOS [12] [13]. The considered RTOS is given as a parameter of the deployment process. Transformation rules of the process are defined independently from the targeted RTOS. This independence is possible thanks to roles previously highlighted. The deployment is performed by mapping each application concept with its execution on the targeted RTOS. The mapping consists in locating the role of the executable concept corresponding to each application concept through the RTOS model. Once a correspondance is established, the structure of the located executable concept is *instanciated*, i.e., duplicated. Each instance is afterwards enriched by specifying its features with the help of the appropriate roles. Features specification emanates from application concepts information. All these specified instances finally constitute the model of the deployed application (i.e., the platform specific model of Figure 3).

The following step concerns either the code generation or the formal model generation. These both processes take in input the same generated deployed application model.

The code generator being not our focus in this paper, we only present the formalization of the deployed application model. Similarly to the deployment, the generation process of deployed application formal models instanciates TPN behavioral descriptions [14]. The location of each TPN is carried out from the structural instances of the deployed application model. Indeed, knowing the source executable concept of each structural instance, the corresponding TPN fragment is located with the <<***behavioralPrototype***>> role. Each corresponding TPN is thus duplicated giving a TPN behavioral instance (or each TPN fragment).

The generation of these TPN fragments engages their composition to constitute a global formal model of the deployed application. The elements serving as connection points must be located through the set of these TPN fragments. Similarly to the structural part, these elements are also located with roles. As instantiation rules, composition rules are based on these roles. In the interest of consistency, we have formalized the sequence of composition rules. This sequence is ordered to avoid any ambiguity in the formal model construction. The TPN fragments are therefore categorized according to RTOS concepts generalized in RTEPML:

- **Concurrent resources:** tasks, interruptions, alarms, etc.
- **Interaction resources:** semaphores, message queues, shared data, events, etc.
- **Routines:** application treatment including services called within the application. This treatment is only represented by the execution time.

The following Algorithm 1 informally describes the sequence of composition rules. We admit here that TPN fragments were already instanciated.

Each composition rule is labelled from a) to d) in comments through this algorithm. Firstly, a) each routine instance is composed of all its called service instances. Then, each concurrent resource must be composed with its execution routine. The execution routine is called the entry point of the concurrent resource. Each entry point is located with the <<***entryPointElement***>> role (see Figure 2). As a consequence, b) each concurrent resource instance is composed with its entry point. The next composition c) concerns all concurrent resource instances in order to put them in concurrency. As a final step, d) interaction resource instances are composed with the set of composed concurrent resource instances so that these latter interact with some of them.

This order will be respected in Section V in which these rules will be formalized. Next, in Section IV, the composition operator of TPN based on roles is defined to formally express these rules afterwards.

Algorithm 1 *Composition rules*

Input:
- $I_S = \{{I_S}^{R_1}, {I_S}^{R_2}, \ldots, {I_S}^{R_l}\}$; // The service calls behavioral instances with $\forall j \in [1, l]$, ${I_S}^{R_j} \subseteq I_S$ and l the number of routines to compose
- $I_C = \{i_{C1}, i_{C2}, \ldots, i_{Cm}\}$; // m concurrent resources behavioral instances
- $I_I = \{i_{I1}, i_{I2}, \ldots, i_{In}\}$; // n interaction resources behavioral instances

Output:
- M // The composed deployed application behavioral model

1: **for** $j = 1$ **to** l **do**
2: // a) Each routine behavioral instance is composed
3: $i_{Rj} \leftarrow ruleComposeRoutine({I_S}^{R_j})$
4: **end for**
5: **for** $k = 1$ **to** m **do**
6: **for all** j such that $1 \leq j \leq l$ **do**
7: **if** $\exists i_{Rj}$ such that i_{Rj} is the entrypoint of i_{Ck} **then**
8: // b) Each entry point is composed
9: $i_{EPk} \leftarrow ruleComposeEntryPoint(i_{Ck}, i_{Rj})$
10: **else**
11: $i_{EPk} \leftarrow i_{Ck}$
12: **end if**
13: **end for**
14: **end for**
15: **for all** k such that $1 \leq k \leq m$ **do**
16: $I_{EP} \leftarrow \{i_{EP1}\} \cup \cdots \cup \{i_{EPk}\} \cup \cdots \cup \{i_{EPm}\}$
17: **end for**
18: // c) All concurrent resources are composed
19: $i_{CR} \leftarrow ruleComposeConcurrentResources(I_{EP})$
20: // d) All interaction resources are composed
21: $i_{IR} \leftarrow ruleComposeInteractionResources(i_{CR}, I_I)$
22: $M \leftarrow i_{IR}$

IV. TPN Composition based on Roles

In order to define the composition of TPN fragments, roles are added to the TPN modeling. These roles are therefore assigned to places. The interest of such a method is to merge places [20] [21], which are the connection points of the deployed system that must be generated in TPN.

In this section, TPN with roles are firstly defined. The definition of the instantiation of TPN with roles is then given. Finally, the composition of TPN is highlighted through a synchronization formalism based on roles.

A. Formal definition of TPN with roles

TPN are a timed extension of classical Petri nets [22] in which an implicit *clock* and an explicit *time interval* are associated with each transition of the net. Informally, the clock measures the time since the transition has been (continuously) enabled, whereas the interval is interpreted as a *firing* condition: the transition, once enabled, may be fired only if the value (or *valuation*) of its clock belongs to the time interval.

In the following, $\mathbb{N}$ denotes the set of natural numbers, $\mathbb{R}_{\geq 0}$ the set of non-negative real numbers, $\emptyset$ is the empty set and $\mathbf{0}$ is the null vector.

Definition 1 (TPN): A TPN $\mathcal{T}$ is a tuple $\langle P, T, \mathsf{Pre}, \mathsf{Post}, m_0, I_s \rangle$ where:

- P is a finite, non-empty set of *places*;
- T is a finite, non-empty set of *transitions*;
- $\mathsf{Pre} : P \times T \rightarrow \mathbb{N}$ is the *backward incidence function*;

- $\mathsf{Post} : P \times T \to \mathbb{N}$ is the *forward incidence function*;
- m_0 is the *initial marking* of the net;
- $I_s : T \to \mathbb{N} \times (\mathbb{N} \cup \{+\infty\})$ assigns a static *time interval* to each transition.

A *marking* of the net $\mathcal{T}$ is an application from P to $\mathbb{N}$ giving for each place of the net the number of tokens it contains. A transition $t \in T$ is *enabled* by a marking m, which is denoted by $t \in enabled(m)$, if all of its input places contain "enough" tokens; more formally, $enabled(m) = \{\, t \in T \mid \forall p \in P,\ m(p) \geq \mathsf{Pre}(p,t) \,\}$. A transition $t \in T$ is *newly enabled* by the firing of transition t_f from the marking m, which is denoted by $t \in \uparrow enabled(m, t_f)$, if it is enabled by the final marking m_f defined by $\forall p \in P,\ m_f(p) = m(p) - \mathsf{Pre}(p,t_f) + \mathsf{Post}(p,t_f)$ but not by the intermediate marking m_i defined by $\forall p \in P,\ m_i(p) = m(p) - \mathsf{Pre}(p,t_f)$. More formally, $\uparrow enabled(m,t_f) = enabled(m_f) \bigcap ((T \setminus enabled(m_i)) \cup \{t_f\})$.

Finally, for any interval I_s, we denote by $I_s{}^{\downarrow}$ the smallest left-closed interval with lower bound 0 that contains I_s. For each transition tr there is an associated clock x_{tr}. We consider valuations on the set of clocks $\{\, x_{tr} \mid tr \in T \,\}$ and we will slightly abuse the notations by writing $v(tr)$ instead of $v(x_{tr})$ to denote the valuation of the clock associated with transition tr.

The operational semantics of a TPN can be formally described as a time transition system; as it is a special case of the semantics of TPN with read and inhibitor arcs (given in Def. 3, we will omit it here for the sake of clarity.

In order to model such behaviors as conditional executions and preemption mechanisms, TPN have been extended with *read arcs* (represented in the following with a white square instead of a regular arrow) and *inhibitor arcs* (represented with a white circle). It should be noted that these arcs only impact the enabling rules of the net but not the marking obtained by firing a transition: read arcs test the presence of tokens in places without consuming them, whereas an inhibitor arc is used to stop the elapsing of time on a transition as long as there is a certain number of tokens in the place.

Definition 2 (TPN with read/inhibitor arcs): A TPN with read and inhibitor arcs (RI_TPN) is a tuple $\mathcal{T}_{RI} = \langle \mathcal{T}, \mathsf{Read}, \mathsf{Inh} \rangle$ where:

- $\mathcal{T} = \langle P, T, \mathsf{Pre}, \mathsf{Post}, m_0, I_s \rangle$ is a TPN,
- $\mathsf{Read} : P \times T \to \mathbb{N}$ is the *read function*;
- $\mathsf{Inh} : P \times T \to \mathbb{N} \cup \{+\infty\}$ is the *inhibition function*[1].

Informally, a transition is enabled if there are "enough tokens" in the places linked by either input arcs or read arcs *and* if there are "not too many tokens" in the places linked by inhibitor arcs. More formally, the definition of the set of transitions enabled by a marking m is updated as follows:

$$enabled(m) = \{t \in T \mid \forall p \in P, \mathsf{Inh}(p,t) > m(p) \geq \max(\mathsf{Pre}(p,t), \mathsf{Read}(p,t))\}$$

[1]If no inhibitor arcs links a transition t to a place p, then $\mathsf{Inh}(p,t) = +\infty$.

The definition of the set of transitions newly enabled from a marking m by the firing of a transition t_f is similarly updated.

Definition 3 (Semantics of the RI_TPN): The operational semantics of the RI_TPN with read and inhibitor arcs $\mathcal{T}_{RI}$ defined above is given by the time transition system $\mathcal{S} = (Q, q_0 \to)$ such that:

- $Q = \mathbb{N}^P \times \mathbb{R}^T_{\geq 0}$;
- $q_0 = (m_0, \mathbf{0})$;
- $\to \in Q \times (T \cup \mathbb{R}_{\geq 0}) \times Q$ is the *transition relation* and is composed of:
  - the *discrete transition transition*, defined $\forall t_f \in T$ by $(m,v) \xrightarrow{t_f} (m', v')$ iff:
    - $(t_f \in enabled(m)$;
    - $v(t_f) \in I_s(t_f)$;
    - $\forall p \in P,\ m'(p) = m(p) - \mathsf{Pre}(p,t_f) + \mathsf{Post}(p,t_f)$;
    - $\forall t \in T,\ v'(t) = \begin{cases} 0 \textbf{ if } t \in \uparrow enabled(m,t_f) \\ v(t) \textbf{ otherwise} \end{cases}$;
  - the *discrete transition transition*, defined $\forall d \in \mathbb{R}_{\geq 0}$ by $(m,v) \xrightarrow{d} (m,v')$ iff $\forall t \in enabled(m)$, $\forall \delta \in]0;d]$, $(v(t) + \delta) \in I_s{}^{\downarrow}(t)$.

Definition 4 (RI_TPN with roles): A RI_TPN with roles is a tuple $\mathcal{N} = \langle \mathcal{T}_{\mathcal{RI}}, R, \lambda \rangle$ where:

- $\mathcal{T}_{\mathcal{RI}}$ is a RI_TPN,
- R is a finite set of roles,
- $\lambda : P \to R \cup \{\perp\}$ is the function assigning a role to a place and $\perp$ denoting that no role is assigned to a place. Hereafter, some notations and properties of this function are enumerated:
  1) $P_\lambda = \{p \in P \mid \lambda(p) \neq \perp\}$ is the set of places with role.
  2) $\lambda_{\setminus P_\lambda} : P_\lambda \to R$ is an injective function;
  3) $\lambda^{-1} : R \cup \{\perp\} \to P \cup \{\emptyset\}$ such that $\begin{cases} \forall r \in R,\ \lambda^{-1}(r) = \begin{cases} p \text{ if } \lambda(p) = r \\ \emptyset \text{ otherwise} \end{cases} \\ \lambda^{-1}(\perp) = \emptyset \end{cases}$

The operational semantics of the RI_TPN with roles $\mathcal{N} = \langle \mathcal{T}_{\mathcal{RI}}, R, \lambda \rangle$ is the same as that of RI_TPN. Indeed, the use of roles within the definition of RI_TPN does not impact its semantics.

B. Instantiation of RI_TPN with roles

As seen previously, all RI_TPN fragments are instantiated before being composed. In order to distinguish the fragments to compose, atomic elements such as roles, places and transitions must be identified according to the instances names, but also according to referenced instances names.

Indeed, referenced instances emerge when instances are service calls. Each service call refers to a resource instance. As an example, a task activation service refers to a task. The two concepts are distinguished because this has an impact during the composition between a service call instance and

its referenced resource instance. For this reason, the renaming of a role and a renaming of places and transitions are distinctly separated. This distinction is made with the following instantiation operator.

Let $\mathcal{N} = \langle P, T, \mathsf{Pre}, \mathsf{Post}, m_0, I_s, \mathsf{Read}, \mathsf{Inh}, R, \lambda \rangle$ be the RI_TPN with roles to instantiate. The following labels ins and ref respectively gives the names of the instance and the referenced instance. If the instance is a resource, there is no referenced instance with $ref = ins$. The global renaming function $\rightharpoonup$ is a bijective function from Set to Set' where $Set \in \{P, T, R\}$.

Definition 5 (Instantiation of RI_TPN with roles): The instantiation of $\mathcal{N}$ denoted by $\mathcal{N}_{ins} = Ins(\mathcal{N}, ins, ref) = \langle P_{ins}, T_{ins}, \mathsf{Pre}_{ins}, \mathsf{Post}_{ins}, m_{0-ins}, I_{s-ins}, \mathsf{Read}_{ins}, \mathsf{Inh}_{ins}, R_{ref}, \lambda_{ins} \rangle$ is defined by:

$$
\begin{array}{rcl}
\mathcal{N}_{ins} & = & Ins(\mathcal{N}, ins, ref) \\
& = & \mathcal{N} \left| \begin{array}{l}
P_{ins} = \{p_{ins} \text{ s.t. } p \in P \text{ and } p \rightharpoonup p_{ins}\}, \\
T_{ins} = \{t_{ins} \text{ s.t. } t \in T \text{ and } t \in T, t \rightharpoonup t_{ins}\}, \\
R_{ref} = \{r_{ref} \text{ s.t. } r \in R \text{ and } r \rightharpoonup r_{ref}\}, \\
\forall p \in P, \forall t \in T, \forall r \in R, \text{ s.t. } p \rightharpoonup p_{ins}, t \rightharpoonup t_{ins}, \\
\text{ and } r \rightharpoonup r_{ref} \text{ we have:} \\
\mathsf{Pre}_{ins}(p_{ins}, t_{ins}) = \mathsf{Pre}(p, t), \\
\mathsf{Post}_{ins}(p_{ins}, t_{ins}) = \mathsf{Post}(p, t), \\
\mathsf{Read}_{ins}(p, t) = \mathsf{Read}_{ins}(p_{ins}, t_{ins}), \\
\mathsf{Inh}_{ins}(p, t) = \mathsf{Inh}_{ins}(p_{ins}, t_{ins}), \\
\lambda_{ins}(p) = r \text{ iff } \lambda_{ins}(p_{ins}) = r_{ref} \\
\lambda_{ins}(p) = \perp \text{ iff } \lambda_{ins}(p_{ins}) = \perp
\end{array} \right.
\end{array}
$$

C. Specific extension of instantiated RI_TPN with roles

Once RI_TPN are instantiated, we have sometimes been faced with the need to extend them according to the application to deploy. For intance, in a real-time system based on a cooperative multitasking application with priorities, eligible low priority tasks are inhibited by eligible high priority tasks when allocating the processor.

We have focused on this case through this paper in order to enrich our previous work [1] in which all tasks had the same priorities. The cooperative multitasking case with priorities is depicted in Figure 4. The RI_TPN $\mathcal{N}_{T1} = Ins(\mathcal{N}, ins, ref)$ is an instance of concurrent resource such that a periodic task where $ref = ins = T1$. In bold, some places with inhibitor arcs, represented with circles, are connected to the $resume_{T1}$ transition to inhibit the state change from $READY_{T1}$ to $RUNNING_{T1}$. The marking of one of the places set $\{READY_{T2}, READY_{T3}, \ldots, READY_{Tx}\}$ carries out this inhibition action and ensures the cooperative scheduling of tasks.

This action being a scheduling specific case, we defined a dedicated operator for adapting instantiated RI_TPN of concurrent resources such that $\mathcal{N}_{T1}$ to a cooperative scheduling context.

Let $\mathcal{N}_{ins} = \langle P_{ins}, T_{ins}, \ldots, \lambda_{ins} \rangle$ be a RI_TPN with roles of a concurrent resource firstly instantiated as previously seen. $\mathcal{N}^{cs}_{ins} = \langle P^{cs}_{ins}, T^{cs}_{ins}, \ldots, \lambda^{cs}_{ins} \rangle$ represents the same instance extended according to a set of n concurrent resources identified by $INS = \{ins_1, ins_2, \ldots, ins_n\}$ with upper priorities.

Definition 6 (Extension of RI_TPN with roles): Cooperative scheduling of concurrent resources: The

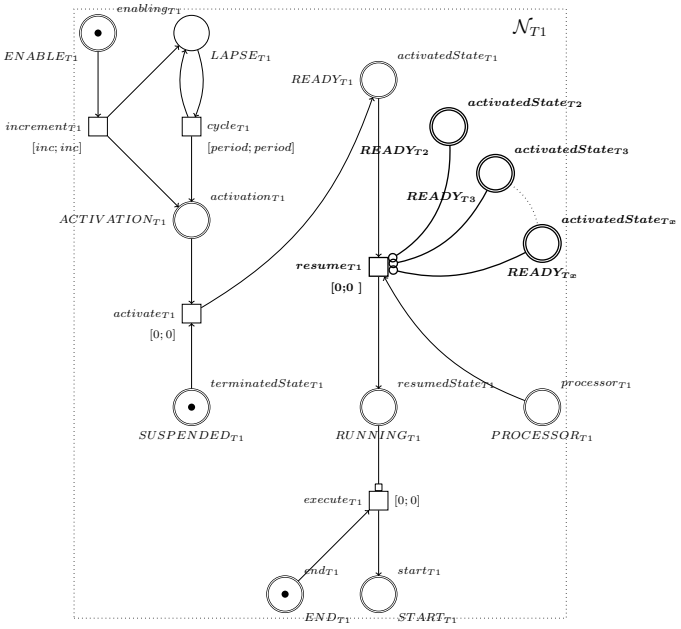


Fig. 4: Specific extension of periodic task in RI_TPN for cooperative multitasking

extension of $\mathcal{N}_{ins}$ in concurrence with n instances adapted to a cooperative scheduling is denoted by:

$$\mathcal{N}^{cs}_{ins} = CoopSched(\mathcal{N}_{ins}, INS)$$

with $\forall t \in T_{ins}$, $\exists \mathsf{Pre}(\lambda^{-1}(activatedState_{ins}), t) \in \mathsf{Pre}_{ins}$ and $\exists \mathsf{Post}(\lambda^{-1}(resumedState_{ins}), t) \in \mathsf{Post}_{ins}$

Formally, this definition gives:

- $R^{cs}_{ins} = R_{ins} \cup R_{INS}$ with $R_{INS} = \bigcup_{\forall i \in [1,n]} \{r_{ins_i}\}$;
- $P^{cs}_{ins} = P_{ins} \cup P_{INS}$ with $P_{INS} = \bigcup_{\forall i \in [1,n]} \{p_{ins_i}\}$;
- $T^{cs}_{ins} = T_{ins}$;
- $\lambda^{cs}_{ins} : P^{cs}_{ins} \rightarrow R^{cs}_{ins}$ is defined by:
  - $\forall p \in P^{cs}_{ins} \setminus P_{INS}$, $\lambda^{cs}_{ins}(p) = \lambda_{ins}(p)$
  - $\forall p \in P_{INS}$ and $\forall i \in [1, n]$, $\lambda^{cs}_{ins}(p) = r_{ins_i}$ with $r_{ins_i} \in R_{INS}$
- $\mathsf{Pre}^{cs}_{ins} : P^{cs}_{ins} \times T^{cs}_{ins} \rightarrow \mathbb{N}$ is defined $\forall p \in P^{cs}_{ins}$ and $\forall t \in T^{cs}_{ins}$ by $\mathsf{Pre}^{cs}_{ins}(p, t) = \mathsf{Pre}_{ins}(p, t)$;
- $\mathsf{Post}^{cs}_{ins} : P^{cs}_{ins} \times T^{cs}_{ins} \rightarrow \mathbb{N}$ is defined $\forall p \in P^{cs}_{ins}$ and $\forall t \in T^{cs}_{ins}$ by $\mathsf{Post}^{cs}_{ins}(p, t) = \mathsf{Post}_{ins}(p, t)$;
- $m_{0^{cs}_{ins}} : P^{cs}_{ins} \rightarrow \mathbb{N}$ is defined $\forall p \in P^{cs}_{ins}$ by $m_{0^{cs}_{ins}}(p) = \begin{cases} m_{0ins}(p) \text{ if } p \in P^{cs}_{ins} \setminus P_{INS} \\ m_{0INS}(p) \text{ if } p \in P_{INS} \end{cases}$ with m_{0INS} is defined by $m_{0INS} : P_{INS} \rightarrow \mathbb{N}$;
- $I_{s^{cs}_{ins}} : T^{cs}_{ins} \rightarrow \mathcal{I}$ is defined $\forall t \in T^{cs}_{ins}$ by $I_{s^{cs}_{ins}}(t) = I_{s_{ins}}(t)$;
- $\mathsf{Read}^{cs}_{ins} : P^{cs}_{ins} \times T^{cs}_{ins} \rightarrow \mathbb{N}$ is defined $\forall p \in P^{cs}_{ins}$ and $\forall t \in T^{cs}_{ins}$ by $\mathsf{Read}^{cs}_{ins}(p, t) = \mathsf{Read}_{ins}(p, t)$;

- $\mathsf{Inh}^{cs}_{ins} : P^{cs}_{ins} \times T^{cs}_{ins} \rightarrow \mathbb{N}$ is defined $\forall p \in P^{cs}_{ins}$ and $\forall t \in T^{cs}_{ins}$ by $\mathsf{Inh}^{cs}_{ins}(p,t) = \begin{cases} \mathsf{Inh}_{ins}(p,t) \text{ if } p \in P^{cs}_{ins} \setminus P_{INS} \\ \\ 1 \text{ if } \begin{cases} p \in P_{INS} \\ t \in T^{cs}_{ins}\ , \\ \exists \mathsf{Pre}(\lambda^{-1}(activatedState_{ins}),t) \in \mathsf{Pre}_{ins} \\ \text{ and} \\ \exists \mathsf{Post}(\lambda^{-1}(resumedState_{ins}),t) \in \mathsf{Post}_{ins} \end{cases} \end{cases}$

D. RI_TPN Synchronization based on roles

In order to synchronize some RI_TPN, we must clarify the definition of the composition of RI_TPN, which will be based on roles assigned to places. Let $\mathcal{N}_1, \ldots, \mathcal{N}_n$ be n RI_TPN $\mathcal{N}_i = \langle P_i, T_i, \mathsf{Pre}_i, \mathsf{Post}_i, m_{0_i}, I_{s_i}, Read_i, Inh_i, R_i, \lambda_i \rangle$ with roles such that $\forall k \neq k' \in [1,n] \implies T_k \cap T_{k'} = \emptyset$ and $P_k \cap P_{k'} = \emptyset$. The composition $\mathcal{N} = \langle P, T, \mathsf{Pre}, \mathsf{Post}, m_0, I_s, R, \lambda \rangle$ of the previous RI_TPN with roles will be denoted by $\mathcal{N} = \mathcal{N}_1 || \mathcal{N}_2 || \ldots || \mathcal{N}_n$. Linked to this composition, we define a function leading to the merging of places whose assigned roles will be taken into account in parameters.

The merging function $\hookrightarrow$ is a partial function from $(R_1 \cup \{\bullet\}) \times (R_2 \cup \{\bullet\}) \times \cdots \times (R_n \cup \{\bullet\}) \rightarrow P \times R$ where $\bullet$ is a special symbol used when a RI_TPN is not involved in a particular merge of the global system. We then extend the definition of the assigning inverse function with $\lambda^{-1}(\bullet) = \emptyset$

The composition of n RI_TPN with m merging is denoted by

$$\Big(\mathcal{N}_1 || \ldots || \mathcal{N}_n\Big)\Big|_{\substack{(r^1_1, \ldots, r^1_n) \hookrightarrow (p^1, r^1) \\ \ldots \\ (r^m_1, \ldots, r^m_n) \hookrightarrow (p^m, r^m)}}$$

with $\forall i \in [1,n]$, $\forall j \in [1,m]$, $r^j_i \in R_i$, $r^j \in R$ and $p^j \in P$, and $\forall k \in [1,m]$, $k \neq j \Rightarrow r^k_i \neq r^j_i$

We will subsequently use the following notations:

- Let $P^{merged}_i \subseteq P_i$ be the set of places of the net $\mathcal{N}_i$ merged by the composition. Formally $P^{merged}_i = \bigcup\limits_{\forall j \in [1,m]} \{\lambda^{-1}_i(r^j_i)\}$
- Let $P^{\hookrightarrow} \subseteq P$ be the set of places of the net $\mathcal{N}$ obtained by the merging. Formally $P^{\hookrightarrow} = \bigcup\limits_{\forall j \in [1,m]} \{p^j\}$

Definition 7 (Composition of RI_TPN with roles): The composition of the n RI_TPN $\mathcal{N}_i$ with the merging $\hookrightarrow$ denoted by:

$$\mathcal{N} = \Big(\mathcal{N}_1 || \ldots || \mathcal{N}_n\Big)\Big|_{\substack{(r^1_1, \ldots, r^1_n) \hookrightarrow (p^1, r^1) \\ \ldots \\ (r^m_1, \ldots, r^m_n) \hookrightarrow (p^m, r^m)}}$$

is defined by:

- $R = \left(\bigcup\limits_{\forall i \in [1,n]} \left(R_i \setminus \bigcup\limits_{\forall j \in [1,m]} \{r^j_i\}\right) \right) \cup \left(\bigcup\limits_{\forall j \in [1,m]} \{r^j\} \right)$;
- $P = \left(\bigcup\limits_{\forall i \in [1,n]} P_i \setminus P^{merged}_i \right) \cup P^{\hookrightarrow}$;
- $T = \bigcup\limits_{\forall i \in [1,n]} T_i$;
- $\lambda : P \rightarrow R$ is defined by:
  - $\forall p \in P \setminus P^{\hookrightarrow}$ meaning that $\exists i$ such that $p \in P_i$ then $\lambda(p) = \lambda_i(p)$
  - $\forall p^j \in P^{\hookrightarrow}$, meaning that p is the result of a merging, $\lambda(p^j) = r^j$
- $\mathsf{Pre} : P \times T \rightarrow \mathbb{N}$ is defined $\forall p \in P$ and $\forall t \in T_i \subseteq T$ by $\mathsf{Pre}(p,t) = \begin{cases} \mathsf{Pre}_i(p,t) \text{ if } p \in P \setminus P^{\hookrightarrow} \text{ and } p \in P_i \\ \\ \mathsf{Pre}_i(p',t), \text{ if } \begin{cases} p \in P^{\hookrightarrow} \text{ and } p' \in P_i \\ (\ldots, r^k_i, \ldots) \hookrightarrow (p, \lambda(p)) \\ \lambda_i(p') = r^k_i \end{cases} \\ \\ 0 \text{ otherwise.} \end{cases}$
- $\mathsf{Post} : P \times T \rightarrow \mathbb{N}$ is defined $\forall p \in P$ and $\forall t \in T_i \subseteq T$ by $\mathsf{Post}(p,t) = \begin{cases} \mathsf{Post}_i(p,t) \text{ if } p \in P \setminus P^{\hookrightarrow} \text{ and } p \in P_i \\ \\ \mathsf{Post}_i(p',t), \text{ if } \begin{cases} p \in P^{\hookrightarrow} \text{ and } p' \in P_i \\ (\ldots, r^k_i, \ldots) \hookrightarrow (p, \lambda(p)) \\ \lambda_i(p') = r^k_i \end{cases} \\ \\ 0 \text{ otherwise.} \end{cases}$
- $m_0 : P \rightarrow \mathbb{N}$ is defined $\forall p \in P$ by: $m_0(p) = \begin{cases} m_{0i}(p) \text{ if } p \in P \setminus P^{\hookrightarrow} \text{ and } p \in P_i \\ \sum\limits_{i=1}^{n} m_{0i}\big(\lambda^{-1}(r^k_i)\big) \text{ if } \begin{cases} p \in P^{\hookrightarrow} \\ (r^k_1, \ldots, r^k_n) \hookrightarrow (p, \lambda(p)) \end{cases} \end{cases}$
- $I_s : T \rightarrow \mathcal{I}$ is defined $\forall t \in T$ by: $I_s(t) = I_{s_i}(t)$ **if** $t \in T_i$;
- $\mathsf{Read} : P \times T \rightarrow \mathbb{N}$ is defined $\forall p \in P$ and $\forall t \in T_i \subseteq T$ as $\mathsf{Pre}(p,t)$;
- $\mathsf{Inh} : P \times T \rightarrow \mathbb{N}$ is defined $\forall p \in P$ and $\forall t \in T_i \subseteq T$ as $\mathsf{Pre}(p,t)$

As an example, $\mathcal{N} = \Big(\mathcal{N}_1 || \mathcal{N}_2 || \mathcal{N}_3\Big)\Big|_{(r_1, r_2, \bullet) \hookrightarrow (p,r)}$ is the parallel composition of the 3 TPN, i.e., $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$, where the place $p_1 \in P_1$ such that $\lambda_1(p_1) = r_1$ and the place $p_2 \in P_2$ such that $\lambda_2(p_2) = r_2$ are merged. The name of the place obtained by this merging in $\mathcal{N}$ is $p \in P$ and its role is $\lambda(p) = r \in R$.

Property 1 (Associativity): The composition of TPN with roles is associative in the following sense:

$$\Big(\mathcal{N}_1 || \mathcal{N}_2 || \mathcal{N}_3\Big)\Big|_{(r_1, r_2, r_3) \hookrightarrow (p,r)} = \Big(\Big(\mathcal{N}_1 || \mathcal{N}_2\Big)\Big|_{(r_1, r_2) \hookrightarrow (p_{12}, r_{12})} || \mathcal{N}_3\Big)\Big|_{(r_{12}, r_3) \hookrightarrow (p,r)} = \Big(\mathcal{N}_1 || \Big(\mathcal{N}_2 || \mathcal{N}_3\Big)\Big|_{(r_2, r_3) \hookrightarrow (p_{23}, r_{23})}\Big)\Big|_{(r_1, r_{23}) \hookrightarrow (p,r)}$$

Property 2 (Commutativity): The composition of TPN with roles is commutative:

$$\Big(\mathcal{N}_1 || \mathcal{N}_2\Big)\Big|_{\substack{(r^1_1, r^1_2) \hookrightarrow (p^1, r^1) \\ \ldots \\ (r^k_1, r^k_2) \hookrightarrow (p^k, r^k)}} = \Big(\mathcal{N}_2 || \mathcal{N}_1\Big)\Big|_{\substack{(r^1_2, r^1_1) \hookrightarrow (p^1, r^1) \\ \ldots \\ (r^k_2, r^k_1) \hookrightarrow (p^k, r^k)}}$$

V. Construction and illustration

The definitions presented above will help with the formal construction of behavioral models expressed as RI_TPN. This

construction will serve as a basis for the transformation process within the SExPIsTools framework (Figure 3). As described in Algorithm 1, the process consists of four successive composition rules, detailled in the paragraphs below and defined by equations (1) to (4).

A construction example in RI_TPN is provided to illustrate the method. Figure 5 presents some RI_TPN with roles, one per box, instantiated and ready for construction. Every operation details the fragments involved in the composition. The mergeable places are represented in double circle and those ready to be merged are connected by a hook-dotted arc with a letter corresponding to the construction step, i.e., the sequence of rules in Algorithm 1. Finally, roles are indicated above and to the right of places.

The whole model is describing a monoprocessor application $Proc$ with two periodic tasks $T1$ and $T2$ sharing the same semaphore S. A cooperative multitasking is established between $T1$ and $T2$ with a non-preemptive context. $T2$ has a higher priority than $T1$. Each task points to an execution routine composed of three services called in the following order: $Get_k(S); Release_k(S); Terminate_k(Tk)$ with $k \in [1, 2]$.

a) ruleComposeRoutine: The list of services considered in RTEPML is not exhaustive at the moment. The instructions described in RI_TPN are currently activation and termination of task, acquisition and release of semaphore and waiting, notification and inhibition of event.

Let n be the number of call services described following: $\{\mathcal{N}_{S1}, \mathcal{N}_{S2}, \ldots, \mathcal{N}_{Sn}\}$ such that $\forall i \in [1, n], \mathcal{N}_{S_i} = Ins(\mathcal{N}_S, S_i, ref\_S_i)$ with $\mathcal{N}_S$ the RI_TPN describing a service, S_i the instance name and ref_S_i the referenced instance name. The routine construction then implies $n - 1$ compositions, each one having m_j mergings of places with $j \in [1, n-1]$. The construction of a routine instance $\mathcal{N}_R$ is given by (1).

Illustration 1 (see Figure 5): By applying $\mathcal{N}_R$ from (1), $\forall k \in [1, 2]$, $\mathcal{N}_{TkBody}$ is built from RI_TPN $\{\mathcal{N}_{Get_k(S)}, \mathcal{N}_{Release_k(S)}, \mathcal{N}_{Terminate_k(Tk)}\}$. This sequence describes in the order, an acquisition of S, a release of S and a termination of Tk.

b) ruleComposeEntryPoint: Each resource points to a routine described by $\mathcal{N}_R$ previously formed. Consequently, $\mathcal{N}_R$ is composed with $\mathcal{N}_{C\tau} = Ins(\mathcal{N}_C, C_\tau, C_\tau)$ where $\mathcal{N}_C$ is the RI_TPN describing a concurrent resource and C_τ is the label indexed to identify each instance. The construction $\mathcal{N}_{EP}$ of a concurrent resource instance with its executable body is given by (2) for m mergings (we admit here that specific extensions of $\mathcal{N}_{C\tau}$ have already been applied for the needs of the application in this equation).

Illustration 2 (see Figure 5): By applying $\mathcal{N}_{EP}$ from (2), $\forall \phi \in [1, 2]$, $\mathcal{N}_{Task\phi\_withBody}$ is built composing $\mathcal{N}_{T\phi}$ with its entry point $\mathcal{N}_{T\phi Body}$. Prior to each composition, $\mathcal{N}_{T1}$ has been extended since this task has the lowest priority. This extension has thus been achieved by the $CoopSched(\mathcal{N}_{T1}, \{T2\})$ operation.

c) ruleComposeConcurrentResources: At this stage, concurrent resources must be linked together with the aim of being scheduled by the same processor.

Let q_C be the number of concurrent resources with their composed executable bodies such that $\forall i_C \in [1, q_C]$, each resource is described by $\mathcal{N}_{EP_{i_C}}$ in accordance with $\mathcal{N}_{EP}$ previously formed. The construction then implies $q_C - 1$ compositions, each one having m_{j_C} mergings with $j_C \in [1, q_C - 1]$. The construction of $\mathcal{N}_{CR}$ is given by (3).

Illustration 3 (see Figure 5): By applying $\mathcal{N}_{CR}$ from (3), $\mathcal{N}_{DeployedApplication_{CR}}$ is firstly composed of $\mathcal{N}_{T1\_withBody}$ and $\mathcal{N}_{T2\_withBody}$.

d) ruleComposeInteractionResources: Note that the processor is also a shared resource. It will therefore be considered as an interaction resource.

Let q_I be the number of interaction resources considered such that $\forall i_I \in [1, q_I]$, each resource is described by $\mathcal{N}_{I_{i_I}} = Ins(\mathcal{N}_I, I_{i_I}, I_{i_I})$ with $\mathcal{N}_I$ the TPN describing an interaction resource. Each interaction resource is composed with $\mathcal{N}_{CR}$ previously formed. The global construction then implies q_I compositions, each one having m_{j_I} mergings with $j_I \in [1, q_I]$. The global composition $\mathcal{N}_{IR}$ is given by (4).

Illustration 4 (see Figure 5): By applying $\mathcal{N}_{IR}$ from (4), $\mathcal{N}_{DeployedApplication}$ is finalized by composing $\mathcal{N}_{DeployedApplication_{CR}}$, $\mathcal{N}_S$ and $\mathcal{N}_{Proc}$.

VI. Experimentation

We illustrate the use of the formal model generation process on a case study. This case study is adapted from a schedulability case [23] in the context of cooperative multitasking.

A. Case study description

We consider an application with three concurrent real-time activities implemented as three real-time schedulable tasks $T1$, $T2$ and $T3$. The concurrency of these tasks emanates from a cooperative multitasking scheduler (based on a non-preemptive priority policy). Here are their characteristics:

- $T1$ is periodic with period $P1 = a$ with $a \in [0, +\infty[$ and has an execution time $C1 \in [10, 20]$.
- $T2$ is sporadic with only a minimal delay of $P2 = 2a$ time units between two activations. The execution time of $T2$ is $C2 \in [18, 28]$.
- Finally, $T3$ is periodic with period $P3 = 3a$ time units and has an execution time $C3 \in [20, 28]$.

These three tasks are defined with the following priority order: $T1 > T2 > T3$. Period a of $T1$ is a parameter determining the limit condition of schedulability of the tasks.

B. Purpose

The formal model generation process will be applied for two different RTOS. The two chosen RTOS are those used to present RTEPML in Section III: OSEK/VDX [16] and VxWorks [17]. Both are used in the industrial sector, have different API and behave differently. Roméo [24], the model-checking tool developed within our team is used to check the generated formal models.

$$\mathcal{N}_R = \left(\left(\left(\mathcal{N}_{S_1}||\mathcal{N}_{S_2}\right)\middle|\begin{array}{l}(end_{ref\_S_1}, start_{ref\_S_2}) \hookrightarrow (S_{S_1\to S_2}, \perp)\\ (r^2_{S_1}, r^2_{S_2}) \hookrightarrow (p^2_{S_2}, r^2_{S_2})\\ \dots\\ (r^{m_1}_{S_1}, r^{m_1}_{S_2}) \hookrightarrow (p^{m_1}_{S_2}, r^{m_1}_{S_2})\end{array}\ ||\mathcal{N}_{S_3}\right)\middle|\begin{array}{l}(end_{ref\_S_2}, start_{ref\_S_3}) \hookrightarrow (S_{S_1S_2\to S_3}, \perp)\\ (r^2_{S_1S_2}, r^2_{S_3}) \hookrightarrow (p^2_{S_3}, r^2_{S_3})\\ \dots\\ (r^{m_2}_{S_1S_2}, r^{m_2}_{S_3}) \hookrightarrow (p^{m_2}_{S_3}, r^{m_2}_{S_3})\end{array}$$

$$\dots||\mathcal{N}_{S_n}\Big)\Big|\begin{array}{l}(end_{ref\_S_{n-1}}, start_{ref\_S_n}) \hookrightarrow (S_{S_1S_2\dots S_{n-1}\to S_n}, \perp)\\ (r^2_{S_1S_2\dots S_{n-1}}, r^2_{S_n}) \hookrightarrow (p^2_{S_n}, r^2_{S_n})\\ \dots\\ (r^{m_{n-1}}_{S_1S_2\dots S_{n-1}}, r^{m_{n-1}}_{S_n}) \hookrightarrow (p^{m_{n-1}}_{S_n}, r^{m_{n-1}}_{S_n})\end{array} \quad (1)$$

with $\forall k \in [1, m_j]$ and $n \geq 2$ if $k \geq 2$ then $r^k_{S_1\dots S_j} = r^k_{S_{j+1}}$

$$\mathcal{N}_{EP} = \left(\mathcal{N}_{C_\tau}||\mathcal{N}_R\right)\Big|\begin{array}{l}(start_{C_\tau}, start_{ref\_S_1}) \hookrightarrow (S, \perp)\\ (end_{C_\tau}, end_{ref\_S_n}) \hookrightarrow (E, \perp)\\ (r^3_{C_\tau}, r^3_R) \hookrightarrow (p^3_{C_\tau}, r^3_R)\\ \dots\\ (r^m_{C_\tau}, r^m_R) \hookrightarrow (p^m_{C_\tau}, r^m_R)\end{array} \quad (2)$$

with $\forall k \in [1, m]$ if $k \geq 3$ then $r^k_{C_\tau} = r^k_R$

$$\mathcal{N}_{CR} = \left(\left(\mathcal{N}_{EP_1}||\mathcal{N}_{EP_2}\right)\middle|\begin{array}{l}(processor_{EP_1}, processor_{EP_2}) \hookrightarrow (P_{EP_1\to EP_2}, processor_{Proc})\\ (r^2_{EP_1}, r^2_{EP_2}) \hookrightarrow (p^2_{EP_2}, r^2_{EP_2})\\ \dots\\ (r^{m_1}_{EP_1}, r^{m_1}_{EP_2}) \hookrightarrow (p^{m_1}_{EP_2}, r^{m_1}_{EP_2})\end{array}\right.$$

$$\left.\dots||\mathcal{N}_{EP_{q_C}}\right)\Big|\begin{array}{l}(processor_{Proc}, processor_{EP_{q_C}}) \hookrightarrow (P_{EP_1\dots EP_{q_C-1}\to EP_{q_C}}, processor_{Proc})\\ (r^2_{EP_1\dots EP_{q_C-1}}, r^2_{EP_{q_C}}) \hookrightarrow (p^2_{EP_{q_C}}, r^2_{EP_{q_C}})\\ \dots\\ (r^{m_{q_C-1}}_{EP_1\dots EP_{q_C-1}}, r^{m_{q_C-1}}_{EP_{q_C}}) \hookrightarrow (p^{m_{q_C-1}}_{EP_{q_C}}, r^{m_{q_C-1}}_{EP_{q_C}})\end{array} \quad (3)$$

with $\forall k_C \in [1, m_{j_C}]$ and $q_C \geq 2$ if $k_C \geq 2$ then $r^{k_C}_{EP_1\dots EP_{j_C}} = r^{k_C}_{EP_{j_C+1}}$

$$\mathcal{N}_{IR} = \left(\left(\mathcal{N}_{CR}||\mathcal{N}_{I_1}\right)\middle|\begin{array}{l}(r^1_P, r^1_{I_1}) \hookrightarrow (p^1_{I_1}, r^1_{I_1})\\ \dots\\ (r^{m_1}_P, r^{m_1}_{I_1}) \hookrightarrow (p^{m_1}_{I_1}, r^{m_1}_{I_1})\end{array}\quad \dots||\mathcal{N}_{I_{q_I}}\right)\Big|\begin{array}{l}(r^1_{PI_1\dots I_{q_I-1}}, r^1_{I_{q_I}}) \hookrightarrow (p^1_{I_{q_I}}, r^1_{I_{q_I}})\\ \dots\\ (r^{m_{q_I}}_{PI_1\dots I_{q_I-1}}, r^{m_{q_I}}_{I_{q_I}}) \hookrightarrow (p^{m_{q_I}}_{I_{q_I}}, r^{m_{q_I}}_{I_{q_I}})\end{array} \quad (4)$$

with $\forall k_I \in [1, m_{j_I}]$ and $q_I \geq 1$, $r^{k_I}_{PI_{j_I-1}} = r^{k_I}_{I_{j_I}}$

The aim is to verify the limits of the schedulability and the valid values of parameter a (i.e., the period of $T1$). The application is schedulable if each activity always has at most one running instance.

The sufficient condition ensuring that the system is schedulable with a non-preemptive priority policy requires a processor load U such that:

$$U = \sum_{i=1}^{n}(C_i/P_i) \leq 1 \quad (5)$$

with n representing the number of tasks, C_i indicating the worst execution time of each task, and P_i being the period (resp. minimal delay) of each periodic (resp. sporadic) task T_i.

The theoretical expected values (calculated without taking into account the RTOS mechanism) for the a parameter are $a \geq 44$ [25]. We expect that our formal verification on the two deployed application on VxWorks and OSEK/VDX leads to the same result.

C. Formal composition fragment

For the sake of clarity, Figure 6 only shows the behavioral arrangement of task $T3$ ($\mathcal{N}_{T3}$) considering the OSEK/VDX norm (Figure 6(a)) on the left side, and the VxWorks platform (Figure 6(b)) on the right side.

$T3$ has been chosen as an illustration instead of other tasks because it has the lowest priority. Consequently, it presents the most complex case. We can indeed note the presence of inhibition arcs since tasks are scheduled in accordance with a cooperative multitasking non-preemptive priority policy. The $\mathcal{N}_{T3}$ instance has consequently been extended by applying $CoopSched(\mathcal{N}_{T3}, \{T1, T2\})$, for each targeted RTOS.

The body of $T3$ is simplified and contains only one service call to *suspend* (in OSEK/VDX variant) or *pend* (in VxWorks variant).

On both Figure 6(a) and Figure 6(b), roles appear in bold to highlight connection points useful for the composition through the RI_TPN. In a similar manner, the mergeable places connected by a hook-dotted arc are the ones located to compose $T3$ and its body according to equation (2).

The same reasoning is obviously applied to $T1$ and $T2$

before composing them with $T3$ in compliance with our formalization through equations (3) and (4).

D. Application verification

Once the models are composed, they have subsequently been checked using Roméo in order to determine the limit value of a so that the RTES application is schedulable. Given the structure of both nets, the systems are schedulable if, at any time, there is at most one token in each place (the nets are then said to be *safe*). Additionally, Roméo provides the set of values of parameter a for which the property is true. The outcome is given hereafter:

```
———-
Checking property AG[0,inf]bounded(1) on TPN:
"/home/clelionnais/TPN/OSEKVDX_NonPreemptiveApplication.xml"
Waiting for response...
Result:
{a >= 44
}
———-
Checking property AG[0,inf]bounded(1) on TPN:
"/home/clelionnais/TPN/VxWorks_NonPreemptiveApplication.xml"
Waiting for response...
Result:
{a >= 44
}
```

Both results match the theoretical value mentionned earlier. We can thus observe that taking into account RTOS mechanisms does not change the theoretical result in this case. Expected constraints are therefore satisfied.

Other properties could be verified, for which taking RTOS mechanisms into account could have an impact. However, this is beyond the scope of this paper. Alternatively, the same property could also be verified starting from a different design model. For instance, we could attempt to model periodicity with a delay instead of an alarm. In such a case, we would be able to verify that the expected properties are not preserved.

However, the purpose of our present case study is simply to illustrate that we can support different platforms (OSEK/VDX and VxWorks) without changing our formalization rules.

VII. Benefits and Limits

One of the major advantage of SExPIsTools is the multi-platform deployment process. The possibility of capitalizing a large number of RTOS models as a parameter of the process, satisfies both reusability and portability criteria. The role notion presented in this paper encourages us in this way to provide more genericity to our transformation rules.

This role notion also fulfills the maintainability requirements. Composition rules have been written independently of the RTOS modeling. In addition, the formalization of these rules could have been done without dealing with other stakeholders concerns. Our Algorithm 1 has been strengthened, detecting errors (i.e., TPN fragments composition ambiguity within rules). As a result, the correctness of the generation process has been improved.

Furthermore, a deployment on two RTOS with different mechanisms has been achieved to show our strategy. The same safety property of schedulability has been verified on both deployments. This illustrates both genericity and correctness of deployed application model construction in TPN.

Another important point is the behavioral modeling in TPN. This results in the possibility to apply verification activities. Moreover, the verification of time properties such as RTES time constraints is possible.

However, to date, this synthesis is an ongoing sketch of proof. The purpose of such a work is to demonstrate the feasability to develop a versatile tool suite. This experimentation must deal with other aspects by considering:

- **more complex RTOS mechanisms** such as preemption, priority ceiling protocol or special queues of message box;
- **other RTOS descriptions** such as ARINC-653 [26], which presents other concepts (e.g., memory partition);
- **other verifications** such as time constraints;
- **more precisely the application**, so that it is not seen as just an ordered sequence of called services.

VIII. Conclusion

In this paper, we have presented a first formalization of the formal model generation process of our SExPIsTools tool suite. As its name suggests, this process generates, from high-level design descriptions, a formal model of the deployed application on a specific RTOS.

The presented formalization focuses on both instantiation and composition rules of the generation process. Indeed, several formal model fragments describing parts of the RTOS and RTES behaviors need to be instantiated and composed. This results in a verifiable global model of the deployed application. The composition rules are independent of a specific RTOS thanks to the notion of role. This notion is an essential point of our strategy and represents a major benefit compared to other existing approaches.

This formalization leads to the definition of a new class of Petri Net, the Time Petri Net with roles and read/inhibitor arcs. A new operator, compared to our previous work [1], has been defined. It allows to model the cooperative scheduling of non-preemptive tasks. This comes to strengthen the instantiation of RI_TPN behavioral fragments according to a priority policy before composing them.

An example of a composition of an application with two RTOS (OSEK/VDX and VxWorks), taking into account the different behavior of the platform, has been given.

Future prospects are scheduled in order to meet the needs identified in Section VII. We are exploring the possibility of extending the formalization with other model classes such as Scheduling TPN [27], where both cooperative and preemptive scheduling are considered.

REFERENCES

[1] C. Lelionnais, M. Brun, J. Delatour, O. H. Roux, and C. Seidner, "Formal Composition Based on Roles within a Model Driven Engineering Approach," in Advances in System Testing and Validation. Venice, Italy: IARIA, Nov. 2013, pp. 27–32. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00941024 and www.thinkmind.org

[2] J. Miller and J. Mukerji, "Model Driven Architecture (MDA) Guide, version 1.0.1." Tech. Rep., June 2003.

[3] J. C. Maeng, D. Na, Y. Lee, and M. Ryu, "Model-Driven Development of RTOS-Based Embedded Software," in Proceedings of the 21st International Conference on Computer and Information Sciences, ser. ISCIS'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 687–696.

[4] B. Kim, I. Lee, L. T. X. Phan, and O. Sokolsky, "Platform dependent code generation of real-time embedded software," in Proceedings of the 4^{th} ACM/IEEE International Conference on Cyber-Physical Systems, ser. ICCPS '13. New York, NY, USA: ACM, 2013, pp. 246–246.

[5] W. El Hajj Chehade, "Contribution to Multiplatform Deployment of Multitasking Applications by High-Level Execution Services Behavioral Modeling," Ph.D. dissertation, Laboratoire d'Ingénierie Dirigée par les Modèles des Systèmes Temps Réels Embarqués (LISE) - CEA Saclay, 2011.

[6] Object Management Group (OMG), "UML Profile for Modeling and Analysis of Real Time and Embbeded Systems (MARTE), version 1.1." Tech. Rep., June 2011.

[7] F. Thomas, S. Gérard, J. Delatour, and F. Terrier, "Software Real-Time Resource Modeling," in Embedded Systems Specification and Design Languages. Springer, 2008, pp. 169–182.

[8] J. Kim, I. Kang, J.-Y. Choi, I. Lee, and S. Kang, "Formal synthesis of application and platform behaviors of embedded software systems," Software & Systems Modeling, 2013, pp. 1–21.

[9] A. Pinto, "Metropolis Design Guidelines," University of California, Berkeley, USA, Tech. Rep., Nov. 2004.

[10] J. Davis, "GME: The Generic Modeling Environment," in Companion of the 18^{th} Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications, ser. OOPSLA '03. New York, NY, USA: ACM, 2003, pp. 82–83.

[11] E. A. Lee, "Overview of the Ptolemy Project," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M03/25, July 2003.

[12] M. Brun, "Contribution to the Software Execution Platform Integration during an Application Deployment Process," Ph.D. dissertation, École Centrale de Nantes, Nantes, France, Oct. 2010.

[13] M. Brun and J. Delatour, "Contribution on the Software Execution Platform Integration During an Application Deployment Process," in First Topcased Day, Toulouse, France, Feb. 2011.

[14] C. Lelionnais, M. Brun, J. Delatour, O. H. Roux, and C. Seidner, "Formal Behavioral Modeling of Real-Time Operating Systems," in 14^{th} Int. Conf. Ent. Information Systems - Model Driven Development for Information Systems (MDDIS 2012), Wroclaw, Poland, June 2012.

[15] F. Thomas, J. Delatour, F. Terrier, and S. Gerard, "Towards a Framework for Explicit Platform-Based Transformations," in 11^{th} IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), May 2008, pp. 211–218.

[16] OSEK/VDX Group, "OSEK/VDX Operating System Specification, version 2.2.3," Tech. Rep., Feb. 2005, http://www.osek-vdx.org/.

[17] WindRiver, "VxWORKS Programmer's Guide, version 6.9." Tech. Rep., Feb. 2011.

[18] P. M. Merlin, "A Study of the Recoverability of Computing Systems," Ph.D. dissertation, 1974, aAI7511026.

[19] M. Boyer and O. H. Roux, "On the Compared Expressiveness of Arc, Place and Transition Time Petri Nets," Fundamenta Informaticae, vol. 88, no. 3, 2008, pp. 225–249.

[20] F. Taïani, M. Paludetto, and J. Delatour, "Composing Real-Time Objects: A Case for Petri Nets and Girard's Linear Logic," in Proceedings of the 4^{th} IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, ISORC-2001. IEEE, 2001, pp. 298–305.

[21] F. Peres, B. Berthomieu, and F. Vernadat, "On the Composition of Time Petri Nets," Discrete Event Dynamic Systems, vol. 21, no. 3, Sept. 2011, pp. 395–424.

[22] C. A. Petri, "Kommunikation mit Automaten," Ph.D. dissertation, Institut für Instrumentelle Mathematik, Bonn, 1962.

[23] G. Bucci, A. Fedeli, L. Sassoli, and E. Vicario, "Timed State Space Analysis of Real-Time Preemptive Systems," IEEE Transactions on Software Engineering, vol. 30, no. 2, Feb. 2004, pp. 97–111.

[24] D. Lime, O. H. Roux, C. Seidner, and L. M. Traonouez, "Roméo: A Parametric Model-Checker for Petri Nets with Stopwatches," in 15^{th} International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2009), ser. Lecture Notes in Computer Science, S. Kowalewski and A. Philippou, Eds., vol. 5505. York, United Kingdom: Springer, Mar. 2009, pp. 54–57.

[25] A. Jovanović, D. Lime, and O. H. Roux, "Integer Parameter Synthesis for Timed Automata," in 19^{th} International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2013), ser. Lecture Notes in Computer Science, N. Piterman and S. Smolka, Eds., vol. 7795. Rome, Italy: Springer, Mar. 2013, pp. 401–415.

[26] Airlines Electronic Engineering Committee, "Avionics Application Software Standard Interface, ARINC Specification 653-1," Tech. Rep., Oct. 2003, aeronautical radio INC., Annapolis, Maryland, USA.

[27] D. Lime and O. H. Roux, "Formal Verification of Real-Time Systems with Preemptive Scheduling," Journal of Real-Time Systems, vol. 41, no. 2, 2009, pp. 118–151, copyright Springer.

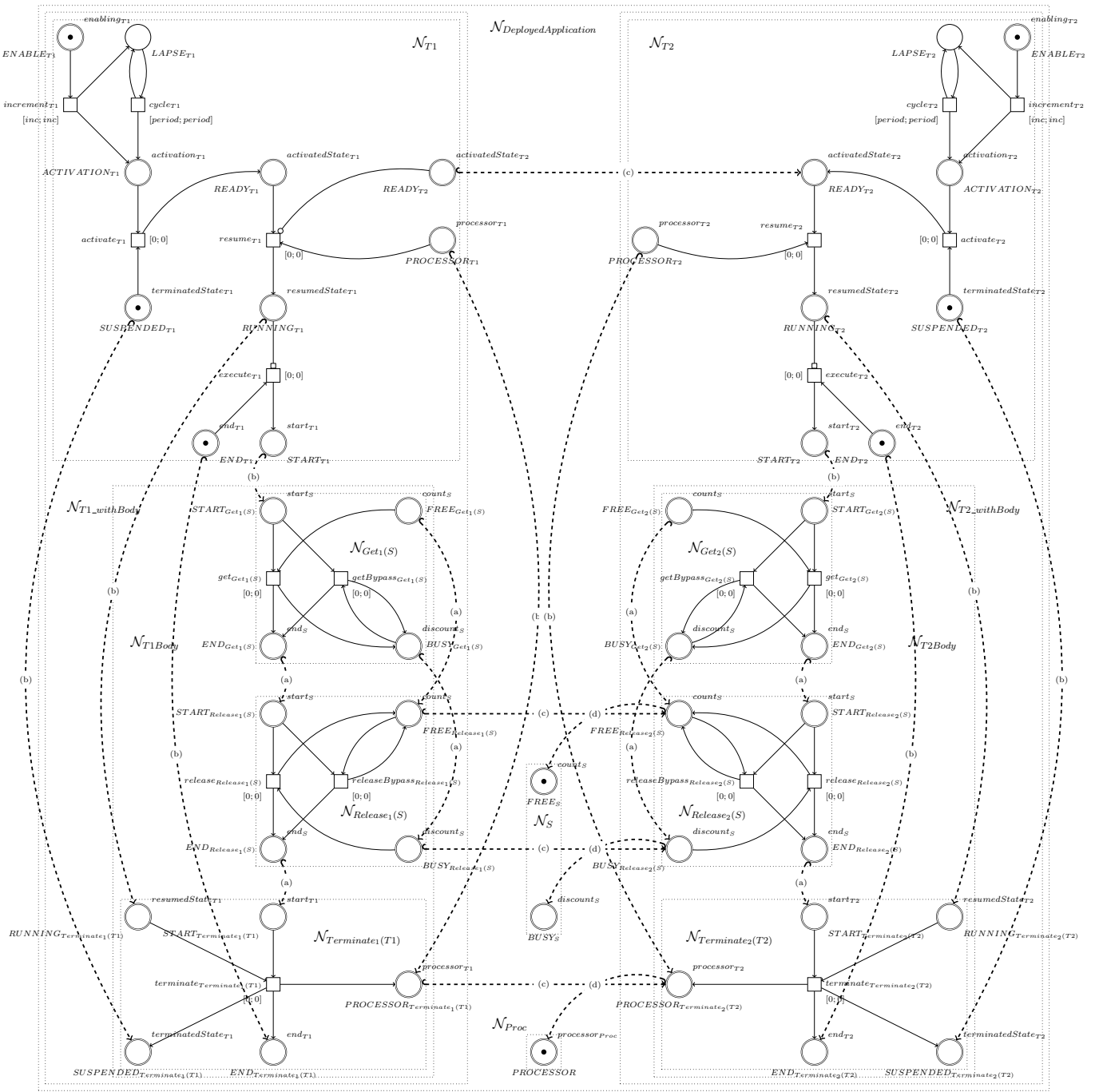
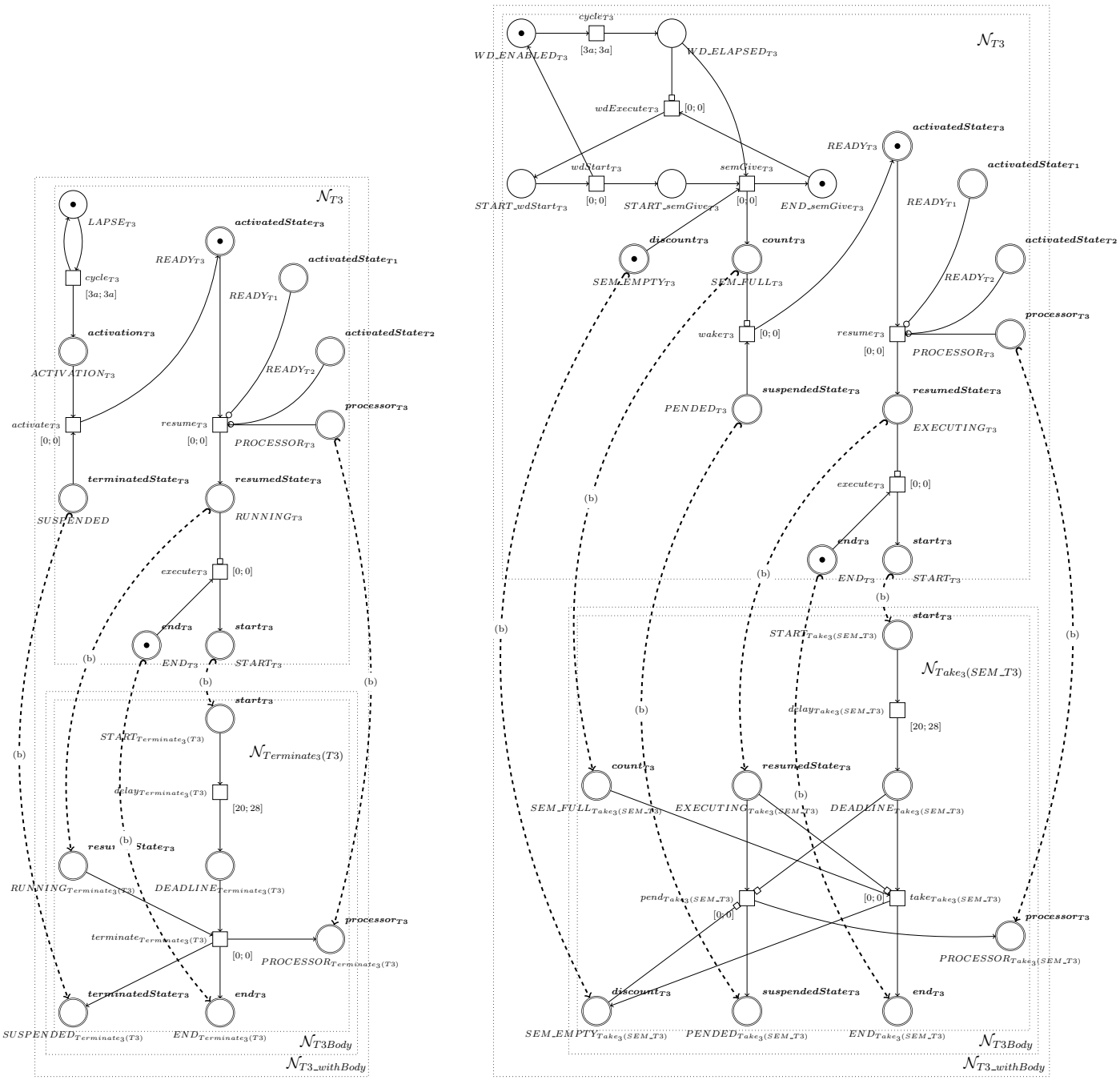


Fig. 5: Deployed application of semaphore sharing composed in RI_TPN



(a) OSEK/VDX (b) VxWorks

Fig. 6: RI_TPN of periodic task implemented on two different RTOS

Towards an Integrated Methodology for the Development and Testing of Complex Systems - with Example

Philipp Helle and Wladimir Schamai
Airbus Group Innovations
Hamburg, Germany
Email: {philipp.helle,wladimir.schamai}@eads.net

***Abstract*—This article reports on a framework for the development and testing of complex systems. The framework provides a meta-model for the description of systems at different levels of abstraction, which is used as a basis for the combination of model-based testing (MBT) techniques for automated test case generation with executable requirement monitors that continuously observe the status of the System under Test (SuT) during test execution. The overall goal is to reduce the total development and testing effort for complex systems. This is accomplished by enabling a high degree of automation and reuse of engineering artefacts throughout the systems engineering lifecycle. The framework is illustrated using an example from the aircraft systems domain: the door locking system.**

***Keywords*—Model-based Systems Engineering, Model-based Testing, Monitor-based Testing, SysML.**

I. Introduction

This article is a revised and extended version of the article [1], which was originally presented at the The Fifth International Conference on Advances in System Testing and Validation Lifecycle (VALID 2013).

The ever-increasing complexity of products has a strong impact on time to market, cost and quality. Products are becoming increasingly complex due to rapid technological innovations, especially with the increase in electronics and software even inside traditionally mechanical products. This is especially true for complex, high value-added systems in the aerospace and automotive domain - the methodology was developed and is therefore embedded in an aeronautic context but generally is independent of a specific domain - that are characterized by a heterogeneous combination of mechanical and electronic components. System development and integration with sufficient maturity at entry into service is a competitive challenge in the aerospace sector. Major achievements can be realized through efficient system testing methods.

”Testing aims at showing that the intended and actual behaviours of a system differ, or at gaining confidence that they do not. The goal of testing is failure detection: observable differences between the behaviours of implementation and what is expected on the basis of the specification”[2].

The typical testing process is a human-intensive activity and as such it is usually unproductive and often inadequately done. It requires human test engineers to manually write test cases. A test case contains a series of test inputs and expected results. Nowadays, the test execution especially on lower levels of testing is largely automated. Nevertheless, this process is cumbersome and costly. Therefore, testing is one of the weakest points of current development practices. According to the study in [3] 50% of embedded systems development projects are months behind schedule and only 44% of designs meet 20% of functionality and performance expectations. This happens despite the fact that approximately 50% of total development effort is spent on testing [3], [4]. This shows the importance and desirability of reducing test effort by advances in the testing methodologies.

Testing needs to be applied as early as possible in the lifecycle to keep the relative cost of repair for fixing a discovered problem to a minimum. This means that testing should be integrated into the lifecycle model so that each phase in the development contributes to the verification of the product as Figure 1 shows. Laycock claims that ”the effort needed to produce test cases during each phase will be less than the effort needed to produce one huge set of test cases of equal effectiveness on a separate lifecycle phase just for testing”[5].

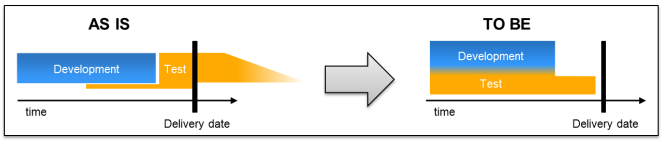


Fig. 1: Envisaged process change

This paper reports on a framework to further automate the system testing process. It is a continuation of the work earlier reported in [6]. The framework provides a meta-model for the description of systems on different layers of abstraction and combines model-based testing (MBT) techniques for automated test case generation based on a whitebox SysML model of the system with executable requirement monitors that continuously observe the status of the System under Test (SuT) during test execution. The overall goal is to achieve a high degree of automation and reuse of engineering artefacts throughout the systems engineering lifecycle.

Paper structure: First, we present background information on SysML, MBT and monitor-based testing (Section II) before we will explain the methodology in detail (Section III). Next, the methodology will be illustrated using an example from the aeronautic domain (Section IV). Finally, we propose a number of ideas for future research (Section V) and close with a summary of the current status (Section VI).

II. Background

This section provides background information on SysML, Model-based testing, Monitor-based testing and related work.

A. SysML

The Unified Modeling Language (UML) [7] is a standardized general-purpose modelling language in the field of software engineering and the Systems Modeling Language (SysML) [8] is an adaptation of the UML aimed at systems engineering applications. Both are open standards, managed and created by the Object Management Group (OMG), a consortium focused on modelling and model-based standards.

SysML is not a methodology, i.e., it does not define what steps need to be performed in what order and which diagrams should be used for which step. Estefan [9] provides an overview of existing methodologies used in industry, some of which use UML-based languages. SysML is a graphical modelling language, i.e., diagrams are used to create and view model data. However, the graphical representation is decoupled from the actual model data. The model data and its graphical representation are typically stored in different files in UML/SysML tools.

Neither UML nor SysML define complete model execution semantics in their core specification. This is different from modelling and simulation languages, such as Modelica [10], which specify the syntax (textual notation) as well as the execution semantics. However, work is underway to resolve that [11], [12], [13]. In the mean time, SysML tool suppliers often provide their own execution semantics [14], so it is possible to include action code into models, generate code from the models and then execute them.

B. Model-based testing

The term MBT is widely used today with slightly different meanings. Surveys on different MBT approaches can be found in [2], [15], [16]. One of them is that ”Model-based testing (MBT) relates to a process of test generation from an SuT model by application of a number of sophisticated methods”[17].

Model-based testing is a variant of testing that relies on explicit behaviour models that encode the intended behaviour and expected failure states of a system and possibly the behaviour of its environment. The use of explicit models is motivated by the observation that traditionally, the process of deriving tests tends to be unstructured, barely motivated in the details, not reproducible, not documented, and bound to the creativity and expertise of single engineers. The idea is that the existence of an artefact that explicitly encodes the intended behaviour can help mitigate the implications of these problems [2].

Intensive research on MBT and analysis has been conducted in recent years, and the feasibility of the approach has been successfully demonstrated, e.g., in [18], [17]. Yet, Boberg [19] shows that most studies apply model-based testing at the component level, or to a limited part of the system while only few studies focus on the application of the technique at the system or even aircraft level. The main difference being that the goal of modelling at system level aims at generating a specification whereas modelling at component level aims at generating code that runs on target. Giese [20] explains that this slow adoption is not only due to scalability reasons but he also claims that ”to benefit from formal verification and early simulation, a model must be precise and detailed with respect to all aspects that are the subject of verifiation. This can usually be carried out in the detailed design phase at the earliest”[20].

A major distinction between the different available MBT approaches can be made by looking at the source of the generated test cases [20]. Some approaches rely on separate explicit test models that are disjunct from the system or specification model, as depicted by Figure 2 while other approaches do not make that distinction and generate test cases from the defined system behaviour as shown by Figure 3.

The usage of explicit test models reflects the different objective (validation vs. solution) and point of view (tester vs. implementer) in creating a test model rather than a specification model [21]. A test model is a model representing all possible stimulations of input of the system interacting in various usage contexts and normally also includes verification points stating what is a correct response from the system to an input and what not. It thereby follows a tester’s view who also has to think of how to combine the possible input stimuli of a system to achieve a high confidence in its correctness.

The main benefit of this approach is the degree of independence it naturally entails between the generated test cases and the system. The generated test cases can thus be used directly to test any form of the SuT, either a model or the implementation. Additionally, as the test model is not a part of the design it can be optimised for validation and verification purposes thereby increasing the chance to uncover defects that are outside the focus of the design artefacts [20]. A drawback of the approach is that there are two models that have to be kept consistent with the requirements at all time, which requires further effort.

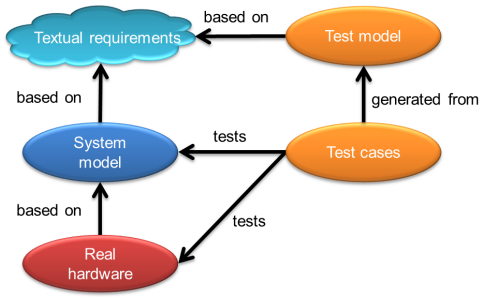


Fig. 2: Model-based testing using explicit test models

One example for an approach that does not rely on explicit models is the work from Lettrari [22] that is the basis for the commercially available IBM Rational Rhapsody Automatic Test Generator (ATG) tool. Test cases are generated from a behaviour model of the SuT using model coverage as test selection criteria. Automated test case generation uses constraint based symbolic execution of the model and search algorithms.

The main benefit is that the approach does not require the creation and maintenance of a separate test model. On the other hand, since the test case generation is not guided by a test engineer it cannot distinguish between ”good” and ”bad” test cases. The only goal for the generator is to achieve a high

degree of model and/or code coverage by generating stimuli that force the executable system model to visit all states and transitions and call all functions of the system's components. Furthermore, there is no independence between the generated test cases and the system model. This means that the test cases cannot be used to test the model they were generated from if the test success criteria is that the observed behaviour and the test case behaviour are the same.

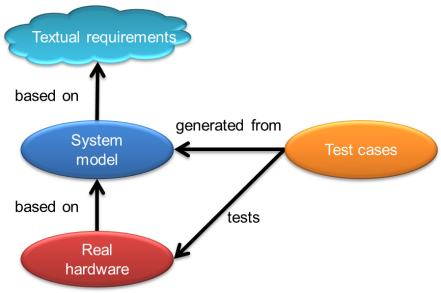


Fig. 3: Model-based testing using design/specification models

C. Monitor-based testing

The idea for formalizing a natural language requirement statement into a requirement monitor is similar to the monitor concept used in runtime verification [23], [24]. A more formal definition states, that "Runtime verification is the discipline of computer science that deals with the study, development, and application of those verification techniques that allow checking whether a run of a system under scrutiny satisfies or violates a given correctness property"[23].

Runtime verification itself deals with the detection of violations of correctness properties. Thus, whenever a violation is observed, it typically does not influence or change the programs execution, say for trying to repair the observed violation. Checking whether an execution meets a correctness property is typically performed using a monitor. In its simplest form, a monitor decides whether the current execution satises a given correctness property by outputting either yes/true or no/false [23].

D. Related Work

In [25], Artho et. al. propose a method for combining test case generation and runtime verification for software systems. In their framework they combine automated test case generation, which is based on a systematic exploration of the input domain of the tested software system using a model checker that is extended with symbolic execution capabilities with runtime verification techniques, that monitor execution traces and verify them against properties expressed in a temporal logic notation. They include further capabilities for the analysis of concurrency errors, such as deadlocks and data races. The paper also provides a description of the application of the method using a NASA rover controller.

Our work differs from the work by Artho et. al. in some major points. Firstly, the test oracles are written as temporal logic formulas whereas we use SysML for both the modelling of the system as well as the requirement monitors. Secondly, the test scenarios are generated based on a definition of all possible inputs using a model checker, whereas we generate the test scenarios from a whitebox model of the system under test.

Drusinsky calls the usage of statecharts for the automated verification of models *execution-based model checking* and compares it to classical model checking, i.e., static analysis, as follows: "[execution-based model checking] seldom yields 100% test coverage, whereas classical model checking consists of a mathematical proof that does yield 100% coverage. The truth, however, is that both classes of techniques require compromises. Execution-based model checking compromises in the achieved test coverage; classical compromises in the size and type of programs that can be verified, and in the kinds of assertions that can be verified to begin with"[26]. In our work, we follow a concept that is similar to what Drusinky calls *execution-based model checking* but embed the idea in an overall framework for the development and testing of systems.

III. Methodology for Development and Testing of Complex Aircraft Systems

This section provides a description of our methodology in terms of the overall concept, the underlying metamodel and the envisaged process.

A. Concept

Our methodology combines monitor- and model-based testing to test the system model and the resulting system. Our aim is to achieve a high degree of reuse of artefacts from early development stages at later development stages and a high degree of automation throughout the process. Since we consider multiple levels of abstraction in our metamodel it is necessary to provide means, which can verify a model at any abstraction level or the final product without the need for redeveloping the verification artefacts for each verification stage. To this end, we use executable requirement monitors, which can be built as soon as the first requirements are defined. The formalized reuquirement monintors can be reused and adapted easily for verifying the models or the product. Also, these monitors can be reused for testing different variants and/or design alternatives.

Figure 4 provides an overview of the main artefacts involved and their relations.

A requirement monitor is an exectuable model representing one requirement that, at any point in time, indicates the requirement violation status. The status should be enumerated with at least the following values [27]:

- Not evaluated (default value), to indicate that the requirement has not been evaluated yet. Typically, this means that a necessary precondition has not been met yet or that the monitor is currently evaluating but could not make a verdict yet.
- Not violated, to indicate that no violation has happened and implying that the requirement has been evaluated.
- Violated, to indicate a violation of the requirement and implying that the requirement has been evaluated.

This enumeration is referred to as ”three-valued semantics” in [23] with the literals ”inconclusive”, ”false” and ”true” respectively.

The monitor status can be obtained from a monitor at any point in time and can change between not evaluated, not violated and violated in any possible way. Following this approach, the status of the individual reuqirement monitors that are instantiated during one test can be used in aggregation to derive the test verdict. Removing the test verdict from the test cases will enable the reuse of test cases, that we now call *test scenarios*, for the verification against several requirements.

The task of converting the natural language statement into a formal language will require a correct interpretation of the requirement statement and the ability to translate the meaning into a model that expresses exactly the same. The general systematic way for deriving a monitor from natural language requirement is as follows:

1) Read the requirement statement
2) Identify properties that can be quantified either by explicit numbers or by logical conditions
3) Identify pre-conditions (if any), which must be satisfied before the requirement can be evaluated
4) Express when the requirement is violated and when not

Neither a particular design of the system nor scenarios are needed for formalizing a requirement. The resulting monitor can be used for the verification of any design alternative of the system using any scenario. Generally, the task of formalizing a requirement into a requirement monitors can be accomplished in many different ways using different formalisms. We decided to use SysML for the task because using the same notation for design and testing artefacts enables integrated development and testing without the need for additional tools or data converters.

We drive the tests using scenarios that we generate from the system models using MBT technology. Since we derive the test verdict from the requirement monitors independently from the system model we can use the scenarios derived from the system model to actually verify the system model as well as the final product.

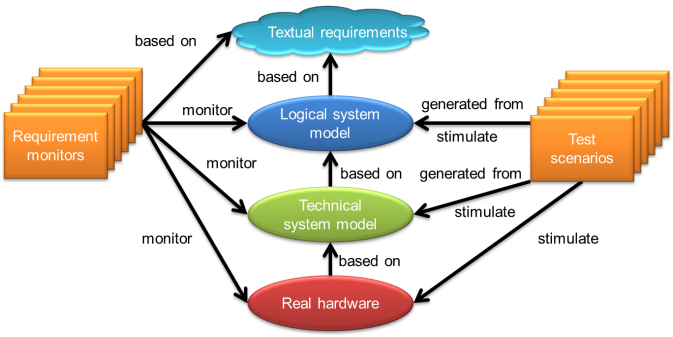


Fig. 4: Model-based testing using monitors

B. Meta-model

For our purpose, we extended the already established meta model for functional and systems architecture modeling [28] to allow a distinction between the functional, logical and the technical architecture of the system as depicted by Figure 5.

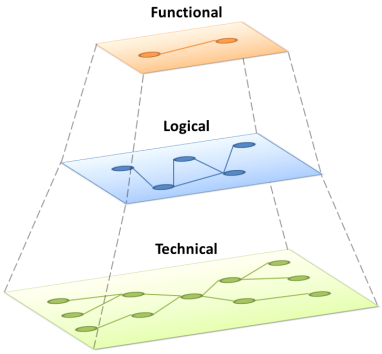


Fig. 5: Levels of abstraction

The main rationale for the distinction between these different layers is reusability. Between different aircraft programmes the functional architecture of a system is quite stable whereas the implementation can differ drastically. For a given aircraft programme the logical architecture is fixed quite early but different technical implementations might be considered and compared in trade studies. Ideally, we can now reuse the same functional architecture that is mature and proven and derive different logical and even more possible technical implementations that satisfy these functional needs.

The functional architecture, consisting of functions and data exchanges via functional dependencies is mapped to a logical system architecture, consisting of logical components that are instances of logical component classes and logical links between these components. This logical architecture can then in turn be mapped to the technical architecture of the system, which contains technical components, i.e., devices, and technical connectors, i.e., cables that connect the components. As can be seen from Figure 6, the relations between the elements in the different modelling layers allow a full traceability. This is crucial especially for maintaining the consistency of the models after changes.

While the modelling of the functional architecture in our approach is purely descriptive, the logical and the technical system architecture models are fully executable. Typically, the complexity of the models increases from the functional over the logical to the technical model. This is mainly due to two reasons: Firstly, when following this top down approach for systems modelling the level of abstraction decreases, which in turn increases the level of detail and complexity. Secondly, most aircraft systems require a certain degree of redundancy to abide by the safety constraints. A fact, which is normally not considered during the functional analysis, only partly in the logical design but has the most impact on the technical architecture.

C. Process

The overall process underlying our methodology is straight forward and consists of the following steps:

1) Formalize requirements: create a violation monitor for each requirement

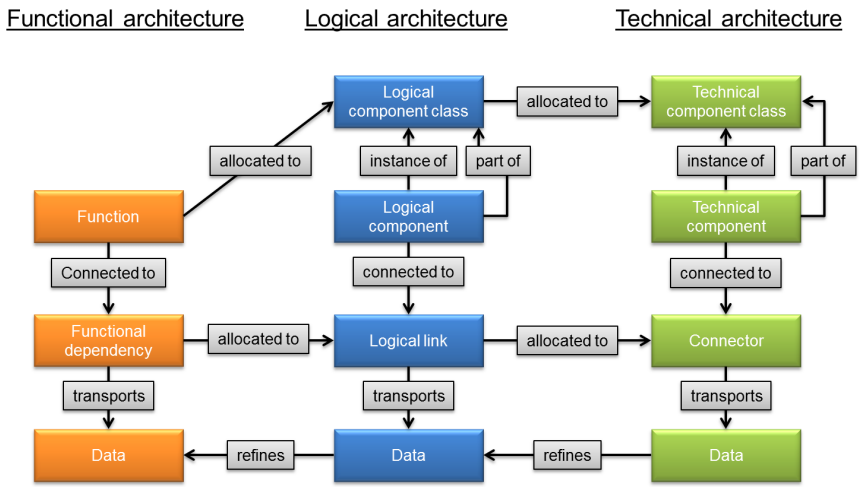


Fig. 6: Meta-model for current approach

2) Build system models
3) Generate test scenarios from system models using MBT
4) Prepare the test environment: instantiate the monitors of the requirements that can be tested using the available scenarios and connect them to the SuT (models or real hardware) appropriately
5) Execute tests: run all defined scenarios
6) Evaluate tests: aggregate the individual statuses of the requirement monitors that were active during a test to derive a test verdict
7) Analyze violations: Find root cause for violation in order to fix it

IV. Example - Door Locking System

We will illustrate steps 1 to 4 of our approach using a simple yet representative example from a passenger aircraft: the Door Locking System (DLS). The DLS controls the locks that fix the aircraft doors in a latched position and prevents unauthorised and unwanted door openings in flight and on ground in case of an existing pressure difference, so-called residual pressure, between the outside of the aircraft and the aircraft cabin. While the rationale for keeping the doors locked in flight at high altitude can be justified by common sense, incidents show that even on ground left-over residual pressure may cause harm [29]. Subsequently, the DLS is a safety-critical part of the aircraft and has to adhere to the according regulations, e.g., the DO-178 and DO-256[30]. The tool Rhapsody by IBM Rational is used for all modelling activities.

A. Initial requirements

The initial requirements of the DLS are provided in Table I. Please note, that this set of requirements serves as an example and is therefore not necessarily complete.

Without any information regarding the implementation of the DLS, a requirement analyst can start to formalize the requirements into requirement monitors. The subsequent sections provide the implementation of the requirement monitors

TABLE I: Description of initial requirements

Req.	Text
REQ-01	If the aircraft doors are unlocked, the DLS shall lock the aircraft doors when receiving the lock door command within 3 seconds.
REQ-02	The DLS shall calculate the residual pressure as the absolute difference between the cabin pressure and the outside pressure.
REQ-03	Once a door is locked, the DLS shall keep the door locked at all times, if the residual pressure exceeds 2.5 mbar.
REQ-04	If the aircraft doors are locked, the DLS shall unlock the aircraft doors when receiving the unlock door command within 3 seconds if the residual pressure is at or below 2.5 mbar.

for REQ-03 and REQ-04. The other requirements can be formalized in a similar fashion.

1) REQ-03: Following the steps described in Section III-A, reading the requirement yields the following properties that can be quantified either by explicit numbers or by logical conditions:

- isDoorLocked (bool, input): locked status of the door
- residualPressure (real, input): amount of residual pressure relevant for this door's control decision
- residualPressureThreshold (real, constant 2.5 mbar): threshold for the residual pressure above which the doors need to be kept locked

Using these identified properties, Figure 7 shows the statechart of the requirement monitor that is used for checking if the system adheres to REQ-03.

2) REQ-04: As before, reading the requirement leads to the identification of the following properties of the requirement that are needed to determine if the requirement is violated:

- isDoorLocked (bool, input): locked status of the door

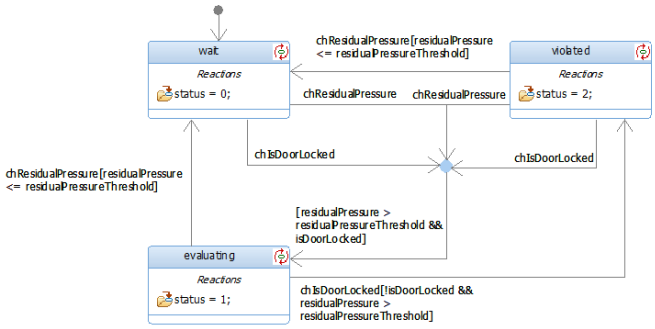


Fig. 7: Requirement monitor statechart for REQ-03

- evUnlockDoorCmd (event, input): unlock command has been send to the DLS
- maxWaitTime (int, constant 3000 ms): time available to open the door after a user sends an unlock command
- residualPressure (real, input): amount of residual pressure relevant for this door's control decision
- residualPressureThreshold (real, constant 2.5 mbar): threshold for the residual pressure above which a door needs to be kept locked

Figure 8 shows the statechart of the requirement monitor that is used for checking if the system adheres to REQ-04. Multiple unlock commands might be send to the DLS one after another and the requirement monitor needs to consider all of them. So, the monitor has an internal queue in which receptions of the *evUnlockDoorCmd* event are stored with a timestamp. The state *evaluating* now continuously (self-transition with timeout *pollTimeOut*) polls the queue and checks whether within 3 seconds (*maxWaitTime*) after the reception of a command by the monitor the door is unlocked or not.

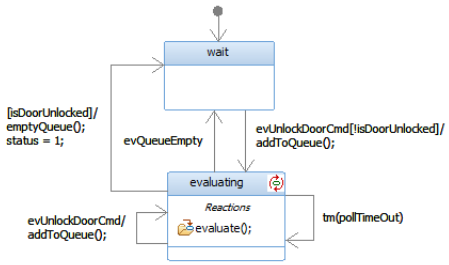


Fig. 8: Requirement monitor statechart for REQ-04

Figure 9 shows the algorithm, modelled as a flowchart, that is used to determine if requirement REQ-04 has been violated.

B. Functional model

Starting with the initial requirements the system engineer can create a functional model. The goal is to identify all functions that need to be performed by the system and the functional dependencies, i.e., data flows, between them. Table II provides a description of all the identified functions and Figure 10 shows the complete functional model. Note, that the functional model is not formal and not executable.

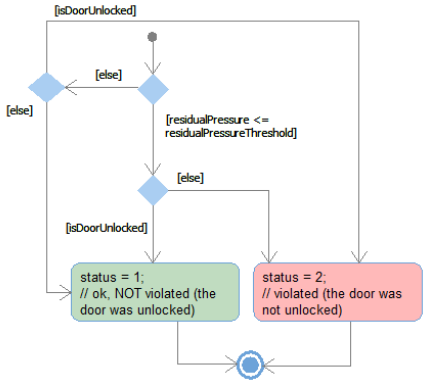


Fig. 9: Flowchart for status evaluation of REQ-04

TABLE II: Description of functions

Function	Description
Issue door commands	The issue door commands function is an interface function that allows the users of the system, i.e. the crew members, to issue commands to open or lock the aircraft doors.
Sense outside pressure	The sense outside pressure function measures the atmospheric pressure outside the aircraft.
Sense cabin pressure	The sense cabin pressure function measures the athmospheric pressure inside the aircraft cabin.
Control door locks	The control door locks function issues controls to the actuate door lock functions according to the user requests taking into account the athmospheric pressure outside and inside the aircraft.
Actuate door locks	The actuate door lock function moves the aircraft door locks between the locked and unlocked position and provides the status of the door locks to the control door locks function.

C. Logical model

The logical model is a much more sophisticated refinement of the functional model geared towards providing an actual specification while still keeping an appropriate level of abstraction.

D. Additional logical requirements

The additional complexity of the logical model compared to the purely descriptive functional model requires further design decisions and allows taking further external requirements into account. The additional requirements of the logical model that have not been taken into account in the functional model are provided by Table III. They are mostly motivated by the actual design of the aircraft that the system will be used in, while the segregation of the left and right side of the aircraft as postulated by REQ-05 is typically motivated by safety considerations and enforced by the airworthiness authorities.

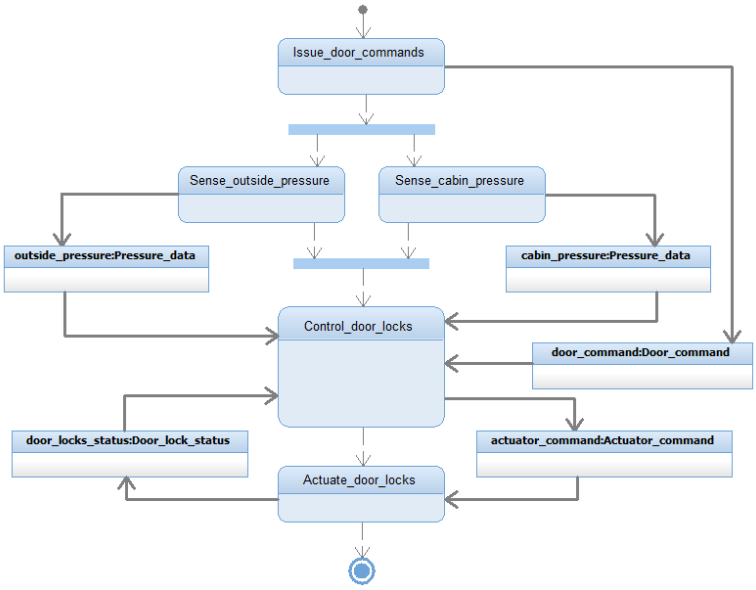


Fig. 10: Functional model

Note, that the functional model and the requirements from that level of abstraction remain valid for the logical model while the additional requirements from the logical level are not considered in the functional model.

TABLE III: Description of additional logical requirements

Req.	Text
REQ-04	The DLS shall control 2 doors on each side of the aircraft, one in the front and one in the back.
REQ-05	Doors on either side of the aircraft shall be controlled separately.
REQ-06	The cabin pressure shall be measured in the front of the cabin and at the back of the cabin.
REQ-07	For determining the residual pressure relevant for the operation of the front doors, the cabin pressure measured in the front of the aircraft shall be used primarily. If the pressure data from the front of the cabin is not available, then the data from the back of the cabin shall be used as backup.
REQ-08	For determining the residual pressure relevant for the operation of the back doors, the cabin pressure measured in the back of the aircraft shall be used primarily. If the pressure data from the back of the cabin is not available, then the data from the front of the cabin shall be used as backup.

Figure 11 shows the internal structure of the logical DLS. As can be seen the additional requirements from Table III have been taken into account, e.g., REQ-06 lead to the multiple instantiation of the *Sense cabin pressure Function* for measuring the pressure in the front as well as in the back of the cabin.

Figure 12 defines the mapping between the functions from the functional model to the logical components of the logical model. Note, that this mapping is at class level. The function *Actuate door locks*, which was responsible for moving all door locks in the aircraft from the functional model is mapped to the logical component class *Actuate door lock Function*, which moves a single door lock in the aircraft and therefore has to be instantiated multiple times in the logical DLS model, once for each door.

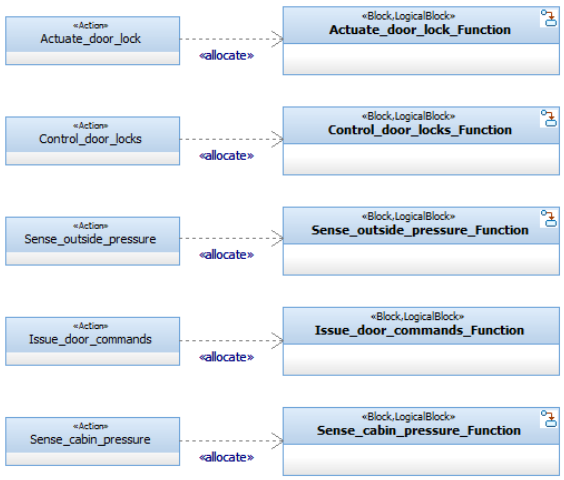


Fig. 12: Mapping between functional and logical model

Likewise, there exists a mapping of the functional dependencies from the functional model to the logical links of the logical model. This mapping can be one to one or one to many. For example, the functional dependency between the function *Sense outside pressure* and the function *Control door locks* that transports the *outside pressure* data is mapped to two logical links in the logical model: the link between *itsSense outside pressure Function* and *itsControl door locks Function Left* and the link between *itsSense outside pressure Function* and *itsControl door locks Function Right* as Figure 13 shows. Formally, this mapping is represented by the fact that the interfaces of the ports of the involved logical blocks contain the data that was previously associated with the functional link; in the example case the *outside pressure* data.

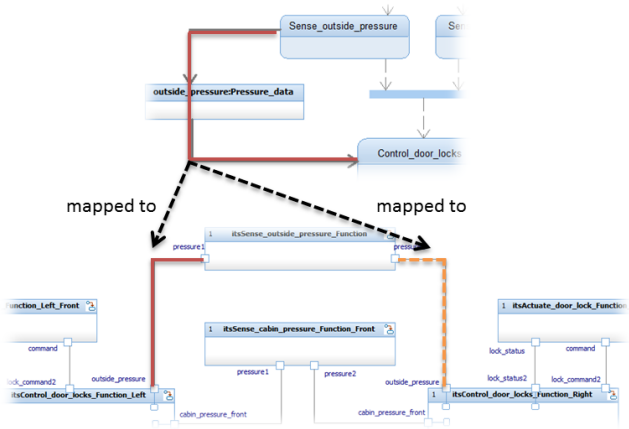


Fig. 13: Mapping between links in the functional and the logical model

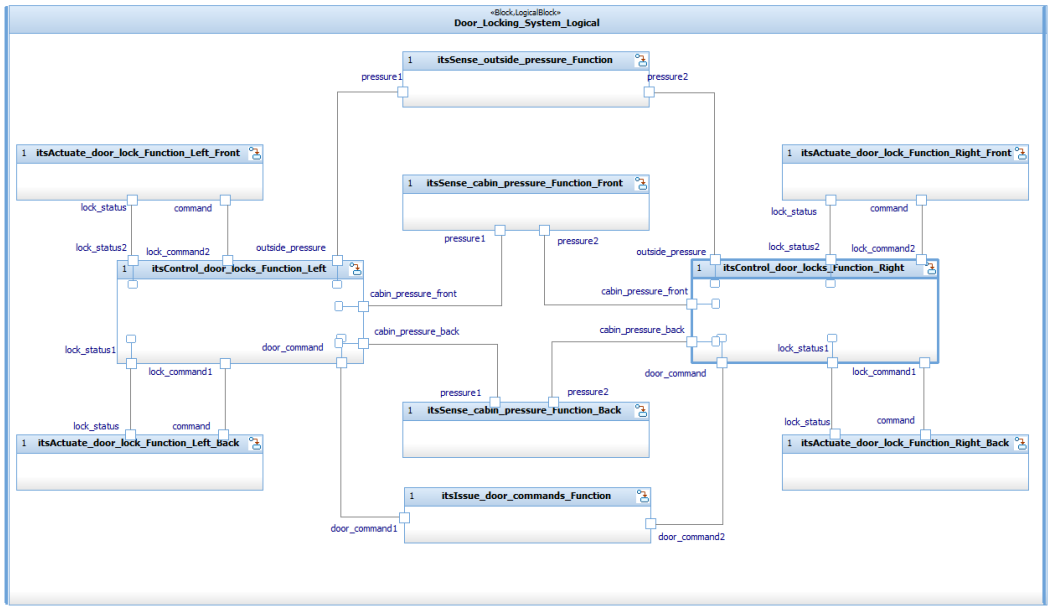


Fig. 11: Logical model

E. Technical model

The technical model is again a refinement of the logical model taking into account physical aspects of the system. Also, further design decisions have been made, i.e., it would be possible to create different versions of the technical model that satisfy the given requirements to represent different design alternatives.

F. Additional technical requirements

The additional requirements of the technical model that have not been taken into account in the functional or logical model are provided by Table IV. Most of them directly influence the choice of technical components, e.g., REQ-12 excludes all actuators that fail the given test. Not all of these requirements can be represented adequately in a SysML model, in our example this is probably true for the requirements 12 to 16. Others require extensions of the modelling profile and external tools for their evaluation, e.g., [28] provides an extension to SysML, the metamodel from Figure 6 and a tool for the evaluation of safety requirements like REQ-09.

Table V provides a description of all components of the technical model and Figure 14 shows the internal structure of the technical DLS. As can be seen, the technical model also includes technical components that are not motivated by the logical model and/or any external requirements: the remote data concentrators (RDC) and the switches. They have been added due to a design decision that the system will make use of the existing aircraft data network, which is based on the Ethernet standard and requires data concentrators to convert data between the network and discrete sensors and actuators.

Figure 15 defines the mapping between the logical components from the logical model to the technical components of the technical model. Keep in mind, that this mapping is again at class level. The mapping is not necessarily a one to one mapping. One logical component might require several technical components to implement the required behaviour. In the example, the logical component class *Actuate door lock Function* from the logical model is mapped to two components of the technical model: the Door Lock Actuator, responsible for moving the door lock, and the Door Lock Sensor, responsible for monitoring the status of the door lock. This is again due to a design decision. It would be perfectly possible to select an actuator that provides its status as an output without the need for an additional sensor.

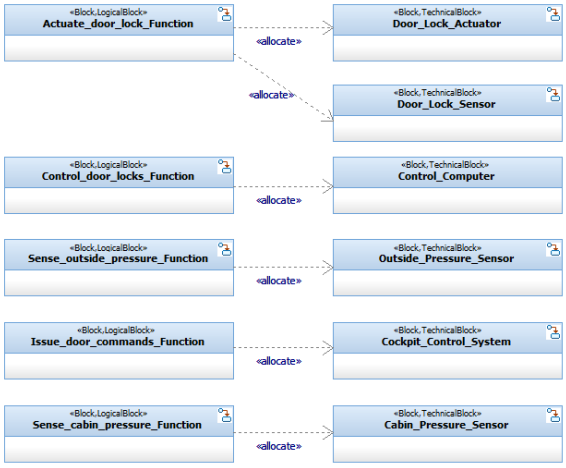


Fig. 15: Mapping between logical and technical model

Likewise, there exists a mapping of the logical links from the logical model to the physical connectors of the technical model. Picking up the example from earlier on, where the mapping of the functional link between the functions *Sense outside pressure* and *Control door locks* to the logical model has been shown, the logical link between *itsSense outside pressure Function* and *itsControl door locks Function Left* can now be mapped to a number of connectors in the technical

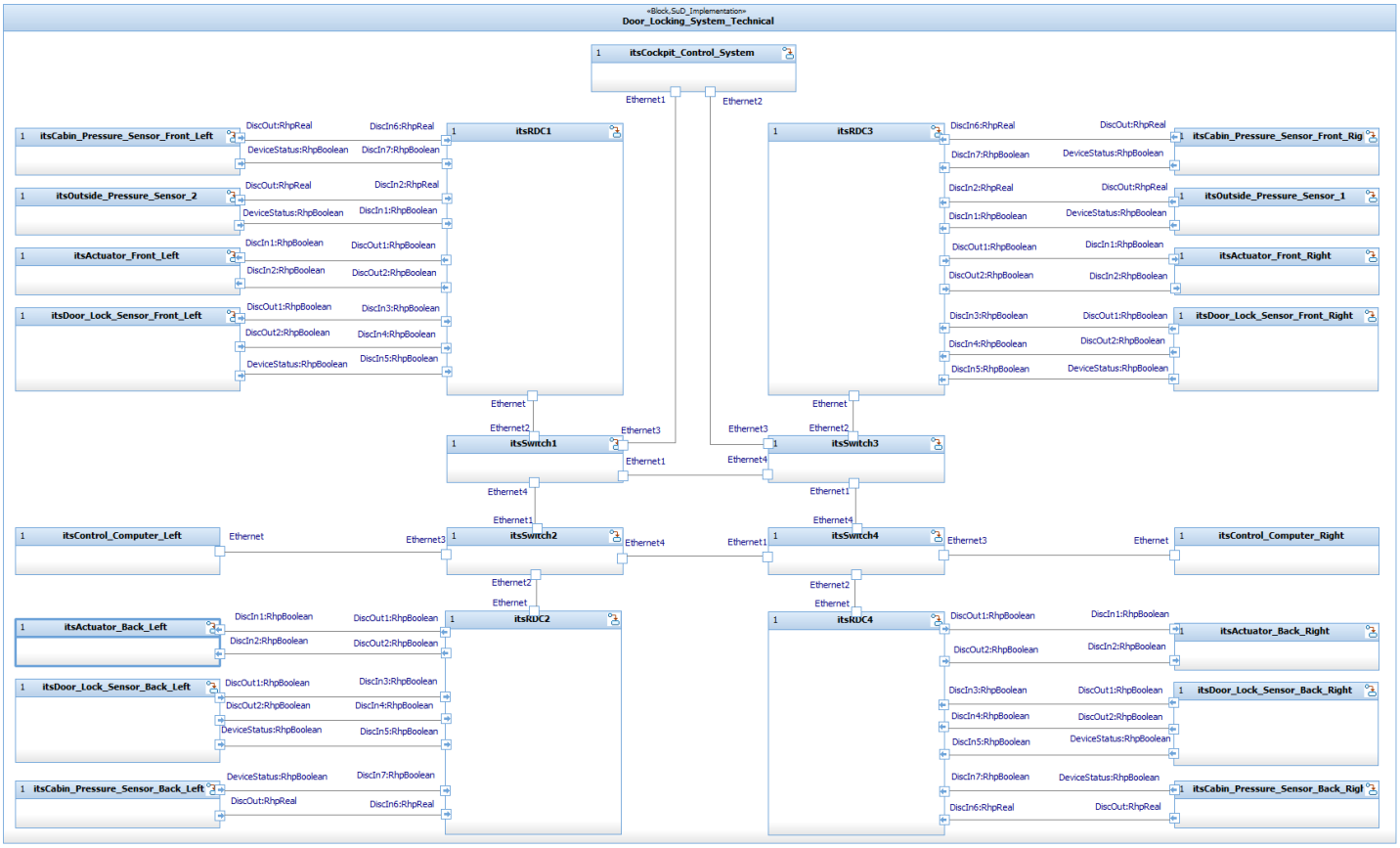


Fig. 14: Technical model

model as depicted by Figure 16.

The UML provides no natural construct to represent this mapping formally. So the information is stored by extending the logical links in the logical model with tags that contain the precise mapping of the logical link to all technical models in which it is implemented as XML data.

G. Testing the models

The logical and the technical model are executable, i.e., all blocks have a statechart that defines their behaviour and, using the execution framework that is provided by the modelling tool, it is possible to generate executable code, which when compiled simulates the model. This allows us to test both models using our defined requirement monitors.

There are two prerequisites for doing so: Firstly, the instantiation of the model under test (MuT) and the requirement monitors in a verification model. Note, that the requirement monitors may have to be instantiated several times, depending on the structure of the model that is to be tested. In the DLS case, the requirements and subsequently their requirement monitors are applicable for each door of the aircraft, hence they have to be instantiated four times in the verification model. Secondly, the connection of the instantiated requirement monitors and the model. The requirement monitors and the MuTs do not necessarily have matching interfaces so an additional entity is required, a so-called mediator [27]. This mediator pulls all the relevant data from the MuT, converts the data to the format that is expected from the requirement monitors and sends it to the requirement monitor instances. Given that the logical and the technical model, or even different implementations, i.e., design alternatives, of one of the models can have quite a different structure, the mediator allows reusing the same requirement monitors for testing all the models.

For the verification of the technical model, we connect a graphical user interface (shown by Figure 17) to the executable verification model. This interface allows stimulating the model and visualises various parameters of the models at runtime, i.e., it enables playing with the model. Since the requirement monitors are active all the time, this kind of testing may lead to uncovering errors that would not have been found with fixed test scenarios.

A more structured verification of a model can be achieved by defining fixed test scenarios. The UML Testing Profile (UTP) [31] is an extension to the UML that provides additional type definitions, such as *test case*, which can be used to manually define test scenarios and the implementation of the UTP in modelling tools allows the automatic execution of these scenarios to verify a model.

And of course, as described in Sections II-B and III-C, it is possible to derive the test scenarios directly from the logical or technical model using a white-box test case generator, such as the ATG for Rhapsody. This tool will systematically stimulate

TABLE IV: Description of additional technical requirements

Req.	Type	Text
REQ-09	Safety	The failure rate for wrong residual pressure determination in a controller shall be no greater than 1E-6/flight.
REQ-10	Weight	The weight of the DLS shall not exceed x kg.
REQ-11	Cost	The costs for purchase and installation of a DLS shall not exceed 0.5 Million Euro per aircraft in serial production.
REQ-12	Environmental	The door lock actuator shall be able to withstand the salt spray test as defined by the applicable standard DO-160E, Section 14, CAT. S
REQ-13	Maintenance	It shall be possible to replace the door lock sensors without removal door lock actuator from the aircraft and without recalibration.
REQ-14	Operational	The DLS shall be designed for a 10000 cycles life in normal operations.
REQ-15	Reliability	The guaranteed meantime between failure (MTBF) of all DLS components shall be at least 30000 flight hours.
REQ-16	Installation	The DLS shall be designed such that persons with a height of between 155 cm and 200 cm are able to install any component without the use of non-standard tools.

the MuT and find test scenarios that cover all states and transitions in the model.

V. Future Work

This section provides a couple of topics for current or future work for extending the approach described in this paper. Apart from extensions to the framework, we are also working on the application of the methodology for a concrete industrial-based use case to validate the framework.

A. Combination of model-based testing and model-based analysis

Dijkstra's famous aphorism holds that tests can only show the presence of errors not their absence [32]. Analysis techniques, e.g., model checking can be used to proof required characteristics of a system. Model-based analysis (MBA) and testing are complementary quality assurance techniques since static and dynamic analysis provide altogether different types of information: typically, static analysis provides general information about a model of the system while dynamic testing provides specific information about the system under test itself. Substantial quality and cost improvement can be obtained when they are systematically applied in combination.

TABLE V: Description of technical components

Technical Component	Description
Cabin Pressure Sensor	Measures the athmospheric pressure inside the aircraft cabin.
Cockpit Control System	External system that has a user interface that allows the users, i.e., the crew members, to issue commands to open or lock the aircraft doors.
Control Computer	Hosts the door lock control function.
Door Lock Actuator	Moves the door lock between locked and unlocked position.
Door Lock Sensor	Monitors the position of the door lock.
Outside Pressure Sensor	Measures the athmospheric pressure outside the aircraft.
RDC	Remote Data Concentrator converts between discrete and network data.
Switch	Ethernet switch for routing data in the aircraft data network.

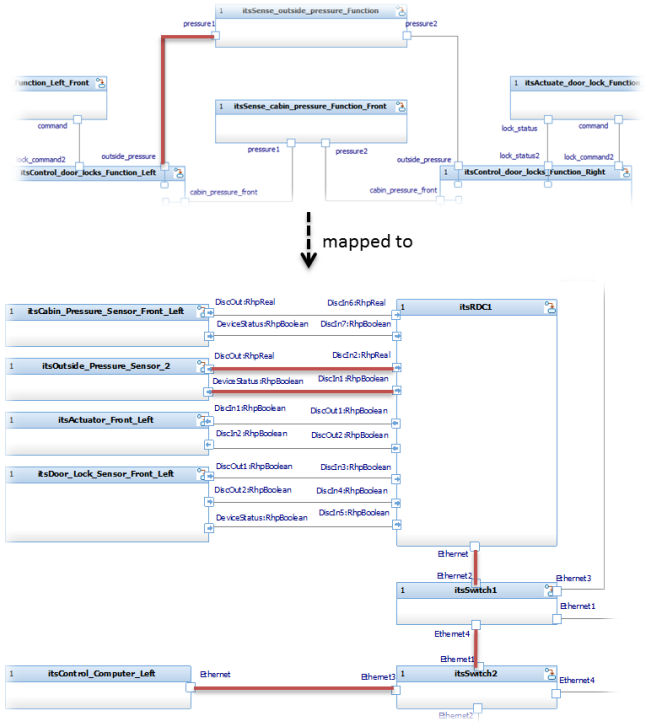


Fig. 16: Mapping between links in the logical and the technical model

One example for such a combination of MBT and MBA is the application of MBA in form of a model checker to improve the completeness of a test suite generated from a whitebox model using MBT as Figure 18 shows. The problem that is addressed by this method is that the automatic test scenario generator does not always achieve to generate a test suite with 100% coverage (coverage for this scenario means model/code coverage). At the moment, manual effort is required to complete a test suite to achieve 100% coverage. This manual effort can be replaced by the application of a model checker. If a test case generator manages to cover 95

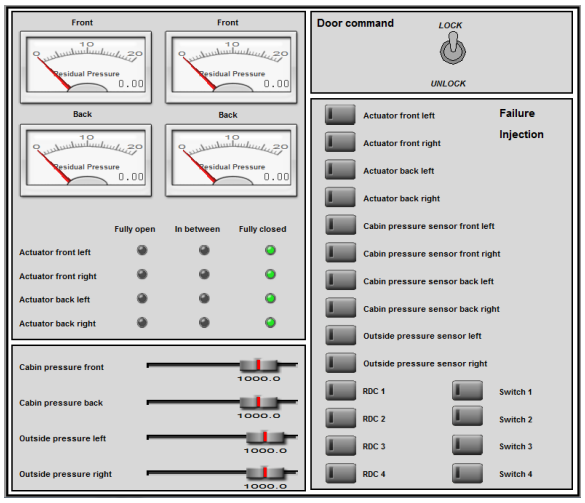


Fig. 17: User interface for simulation control

out of 100 states of a model using test scenarios then we can write properties that check the reachability of the remaining five states. If the model checker manages to reach a state then the proof trace provided by the model checker can be directly added to the test suite as a new test scenario. If the model checker cannot find a solution for reaching a state then the model needs to be adapted.

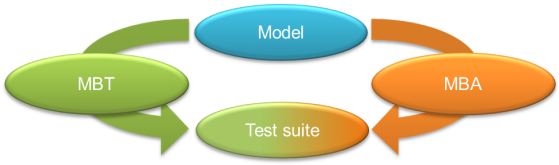


Fig. 18: Combination of model-based testing and model-based analysis

B. Combination of test scenarios obtained from different sources

Evluation of different MBT approaches and tools in the recent past, e.g., [33], [34] showed, that each tool has specific strengths and weaknesses and almost none of them can replace additional manual test scenario creation completely. If we use more than one test scenario generation approach and if we allow test scenario generation at different levels of abstraction as Figure 19 shows, then there is a high probability that the resulting test suite contains a high amount of redundant test scenarios. In order to test efficiently, especially when we are in the phase of hardware testing where a test run is much more expensive than a test run on a model, the redundancy in the test suite must be reduced to find an optimal test suite. Adaptation of previous work, e.g., [35], on that topic to our overall development and testing approach is currently being investigated.

C. Automated model-composition and results evaluation

The creation of verification models, i.e., models that integrate requirement monitors, a SuT system model and scenarios, i.e., the finding of suitable combinations of system model, scenarios and requirements, can be automated. Such a combined

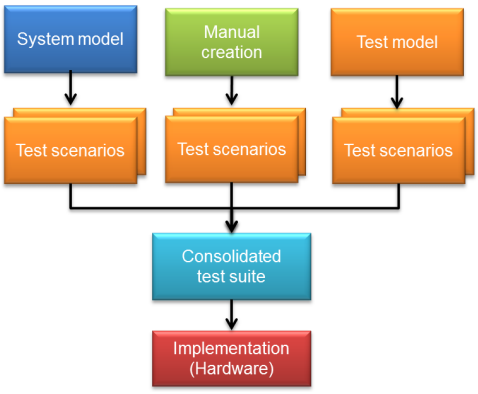


Fig. 19: Optimal test suite from different sources

verification model consists of one system model, which can be logical or technical, one scenario that can stimulate the design alternative, a set of requirements, which can be tested using the selected scenario and a mediator that ensures the compatibility between the involved models. To automate the process further information is needed to evaluate the suitability of a combination of a test scenario and a design model, a test scenario and a requirement or a requirement and a system model. An approach for encoding this information and thereby enabling the automated composition of such verification models is presented in [36]. Combining this approach with the one presented here is ongoing work. Additionally, running the tests, post-processing of the test results, and presenting the verification results appropriately can also be done in an automated fashion.

VI. Conclusion

We presented a framework for an integrated development and testing approach of complex systems and showed its application using an example from the aircraft system domain, the Door Locking System. The main driver behind this development was the need for more efficient testing. This was succesfully achieved by increasing the degree of reusability of engineering artefacts and automation of the testing process in the following way:

- Reusability
  - Explicit modelling of different architecture levels enables reuse of architectures.
  - Requirement monitors can be reused for testing different architecture levels as well as the real hardware product.
  - Removing verdicts from test cases allows using the same test scenario for testing multiple requirements. Additionally, testing a requirement in different test scenarios increases the confidence in the conclusions drawn from the test results.
- Automation
  - Executable requirement monitors allow automated test verdict derivation.
  - Generation of scenarios using MBT.
  - Automated test execution of formal test scenarios.

The approach requires, as most model-based approaches, a frontloading of effort, a personnel shift and a different education of the involved people compared to the current state of practice. While evidence suggests that, through the high degree of reuse and automation, the effort for model-based testing is only slighly higher than the one for traditional testing [37] the adoption of the presented approach in an industrial environment nevertheless requires a rethinking of traditional roles and process steps.

ACKNOWLEDGMENT

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement no. 269335 (ARTEMIS project MBAT) and from the German BMBF.

REFERENCES

[1] P. Helle and W. Schamai, "Towards an integrated methodology for the development and testing of complex systems," in VALID 2013, The Fifth International Conference on Advances in System Testing and Validation Lifecycle, 2013, pp. 55–60.

[2] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing approaches," Software Testing, Verification and Reliability, vol. 22, no. 5, pp. 297–312, Aug 2012.

[3] V. Encontre, "Testing embedded systems: Do you have the GuTs for it," IBM: http://www.ibm.com/developerworks/rational/library/459.html, Nov 2003, last visited on 5.5.2014.

[4] A. Helmerich et al., "Study of Worldwide Trends and R&D Programmes in Embedded Systems in View of Maximising the Impact of a Technology Platform in the Area," European Commission: http://www.artemis-austria.net/uploads/media/FAST_final-study-181105_en.pdf, Nov 2005, last visited on 5.5.2014.

[5] G. Laycock, The theory and practice of specification based testing. Department of Computer Science, University of Sheffield, 1993.

[6] P. Helle and W. Schamai, "Specification model-based testing in the avionic domain - Current status and future directions," in Proc. Sixth Workshop on Model-Based Testing (MBT 2010), ser. ENTCS, vol. 264, no. 3, 2010, pp. 85 – 99.

[7] Object Management Group, "OMG Unified Modeling Language (OMG UML), v2.3," 2010.

[8] ——, "OMG Systems Modeling Language (OMG SysML), v1.2," 2008.

[9] J. Estefan, "Survey of Model-Based Systems Engineering (MBSE) methodologies," INCOSE: https://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf, 2008, last visited on 5.5.2014.

[10] P. Fritzson and V. Engelson, "Modelica - a unified object-oriented language for system modeling and simulation," in Proc. European Conference on Object-Oriented Programming (ECOOP98). Springer, 1998, pp. 67–90.

[11] B. Selic, "The less well known UML," in Formal Methods for Model-Driven Engineering, ser. LNCS, M. Bernardo, V. Cortellessa, and A. Pierantonio, Eds. Springer, 2012, vol. 7320, pp. 1–20.

[12] Object Management Group, "Semantics Of A Foundational Subset For Executable UML Models (FUML, v.1.0)," 2011.

[13] ——, "Concrete Syntax For A UML Action Language: Action Language For Foundational UML (ALF), v1.0.1 beta," 2013.

[14] D. Harel and H. Kugler, "The Rhapsody Semantics of Statecharts (or, on the executable core of the UML)," in Integration of Software Specification Techniques for Applications in Engineering, ser. LNCS, H. Ehrig et al., Eds. Springer, 2004, vol. 3147, pp. 325–354.

[15] M. Utting and B. Legeard, Practical Model-Based Testing: A Tools Approach. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[16] M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. Pretschner, Model-Based Testing of Reactive Systems: Advanced Lectures (Lecture Notes in Computer Science). Springer, 2005.

[17] J. Zander-Nowicka, Model-based testing of real-time embedded systems in the automotive domain. Fraunhofer FOKUS, Berlin, 2009.

[18] T. Bauer, H. Eichler, A. Rennoch, and S. Wieczorek, Model-based Testing in Practice - Proc. 2nd Workshop on Model-based Testing in Practice (MoTiP 2009). University of Twente, The Netherlands, 2009.

[19] J. Boberg, "Early fault detection with model-based testing," in Proc. 7th ACM SIGPLAN workshop on ERLANG. New York, NY, USA: ACM, 2008, pp. 9–20.

[20] H. Giese, G. Karsai, E. A. Lee, B. Rumpe, and B. Schatz, Model-Based Engineering of Embedded Real-Time Systems, ser. LNCS. Springer, 2010, vol. 6100.

[21] H. Le Guen, F. Valle, and A. Faucogney, Model-Based Testing - Automatic Generation of Test Cases Using the Markov Chain Model. Wiley, 2012, pp. 29–81.

[22] M. Lettrari, "Using abstractions for heuristic state space exploration of reactive object-oriented systems," in FME 2003: Formal Methods. Springer, 2003, pp. 462–481.

[23] M. Leucker and C. Schallhart, "A brief account of runtime verification," Journal of Logic and Algebraic Programming, vol. 78, no. 5, pp. 293–303, 2009.

[24] J. Levy, H. Saïdi, and T. E. Uribe, "Combining monitors for runtime system verification," ENTCS, vol. 70, no. 4, pp. 112–127, 2002.

[25] C. Artho et al., "Combining test case generation and runtime verification," Theoretical Computer Science, vol. 336, no. 2, pp. 209–234, 2005.

[26] D. Drusinsky, Modeling and Verification Using UML Statecharts: A Working Guide to Reactive System Design, Runtime Monitoring and Execution-based Model Checking. Newnes, 2011.

[27] W. Schamai, "Model-based verification of dynamic system behavior against requirements : Method, language, and tool," Ph.D. dissertation, Linkoeping University, Department of Computer and Information Science, The Institute of Technology, 2013.

[28] P. Helle, "Automatic SysML-based safety analysis," in Proceedings of the 5th International Workshop on Model Based Architecting and Construction of Embedded Systems, ser. ACES-MB '12. New York, NY, USA: ACM, 2012, pp. 19–24.

[29] National Transportation Safety Board, "Safety Recommendation, A-00-23 through -27," August 2, 2002.

[30] V. Hilderman and T. Baghi, Avionics certification: a complete guide to DO-178 (software), DO-254 (hardware). Avionics Communications, 2007.

[31] Object Management Group, "UML Testing Profile (UTP), v1.2," 2013.

[32] E. W. Dijkstra, "The humble programmer," Communications of the ACM, vol. 15, no. 10, pp. 859–866, 1972.

[33] M. Shafique and Y. Labiche, "A systematic review of model based testing tool support, Technical Report SCE-10-04," Carleton University: http://people.scs.carleton.ca/ jeanpier/404F13/T7 - survey papers/SystematicReviewOfMBT-2010.pdf, 2010, last visited on 5.5.2014.

[34] A. C. Dias Neto, R. Subramanyan, M. Vieira, and G. H. Travassos, "A survey on model-based testing approaches: a systematic review," in Proc. 1st Workshop on Empirical Assessment of Software Engineering Languages and Technologies (WEASELTech'07). ACM, 2007, pp. 31–36.

[35] G. Fraser and F. Wotawa, "Redundancy based test-suite reduction," in Fundamental Approaches to Software Engineering, ser. Lecture Notes in Computer Science, M. Dwyer and A. Lopes, Eds. Springer, 2007, vol. 4422, pp. 291–305.

[36] W. Schamai, P. Fritzson, C. J. J. Paredis, and P. Helle, "ModelicaML value bindings for automated model composition," in Proc. 2012 Symposium on Theory of Modeling and Simulation - DEVS Integrative M&S Symposium, ser. TMS/DEVS '12. Society for Computer Simulation International, 2012, pp. 31:1–31:8.

[37] T. Bauer, F. Böhr, and R. Eschbach, "On MiL, HiL, statistical testing, reuse, and efforts," in 1st Workshop on Model-based Testing in Practice (MoTiP 2008). Fraunhofer, 2008, pp. 31–40.

An Integrated into FPGA System for Optical Link Testing and Parameters Tuning

Anton Kuzmin*, Dietmar Fey*, and Ulrich Lohmann†
*Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Computer Architecture
{anton.kuzmin,dietmar.fey}@informatik.uni-erlangen.de
†Chair of Micro- and Nanophotonics, Fern University in Hagen
ulrich.lohmann@fernuni-hagen.de

***Abstract*—Development, characterization and performance optimization of systems utilizing FPGAs with high-speed serial transceivers to implement optical links with 1 to 10 Gbps data rate is a complex task and it poses several challenges for design engineers. In this paper, an effective approach is presented designed to address these challenges based on the use of diagnostic features implemented in the transceivers and a soft-IP microcontroller system instantiated in the FPGA. The use of the soft-IP controller allows a single-point access to the control and diagnostic interfaces of all components forming the link. Combined with computational capabilities and a high-level programming language interpreter running on the soft-IP CPU inside the FPGA, it enables extensive optical link performance evaluation without relying on any additional test and measurement equipment and significantly shortens debugging and testing times. Two generations of the system including hardware, soft-IP microcontroller system and embedded software are presented. The implementation demonstrates the feasibility and effectiveness of the proposed approach to utilization of on-chip diagnostic capabilities.**

***Index Terms*—Optical fiber communication; Transceivers; FPGA; Microcontrollers; Embedded software**

I. Introduction

Modern applications including rich media content transport, on-the-fly image processing, high bandwidth data acquisition for experimental physics, and high performance computing, require ever increasing serial communication data rates. At the same time, latency requirements remain strict and significantly limit possibilities for error correction and therefore call for a lower number of acceptable errors in the communication channel. FPGA devices with integrated high-speed serial transceivers and optical interconnects provide a very efficient and flexible platform for implementing such demanding applications and can be found in an increasing number of systems. A prototype system combining FPGA with optical interconnect implemented by the authors was presented at the VALID 2012 [1]. This paper demonstrates an extension of the original test system to a platform for reconfigurable computing with parallel optical links. Various examples and applications of optical interconnects could be found in [2]–[6].

One of the major challenges is a parameter tuning of the various components forming an interconnect to achieve the lowest possible probability of bit errors. The problem is that accurate measurements at low error probabilities require very long times even at high data rates to accumulate statistics for a given confidence level while the parameter optimization space is relatively big. Additional complications arise from the fact that various components of the link have very different interfaces for setting parameters. In most cases, they are supported by proprietary tools with limited functionality for automatically tuning link parameters. The application of these tools often requires a connection of the system to external test and measurement equipment. The limitations associated with its usage become increasingly severe with a tighter integration between the FPGA and the optical transceiver blocks as recently proposed by Li et al. [7]. This level of integration makes electrical signals between the FPGA and optical transceiver practically inaccessible for external test equipment.

This paper presents an approach designed to address challenges associated with the testing, parameter tuning and performance monitoring of optical interconnects in FPGA-based systems. The approach is based on the use of a soft-IP controller embedded into the FPGA to perform two major tasks: link performance measurements and control of parameters of the different components forming the link.

The paper has the following structure. In the first section, an example of utilization of FPGA built-in transceiver diagnostic capabilities is presented and the key differences in the approach chosen by the authors are outlined. In the subsequent section, an overall inter-FPGA transceiver-based serial link structure is shown followed by brief description of its components and their respective configurable and tunable parameters. Then, a Bit Error Ratio (BER) [8] is introduced as an integral characteristic of link performance. An optimized algorithm for obtaining an accurate BER scan plot (bath-tub curve) is described. It can be used for indirect eye diagram width measurement by introducing a phase shift into a signal sampling point inside the receiver. The eye diagram width may serve as an indicator of the link performance and is used as a target function for the link parameter optimization.

Implementation aspects of the FPGA-based optical link test system are discussed in the next parts of the paper along with the obtained link performance measurement results. Comparison of the measured BER levels obtained on the prototype system for different optical modules confirms the validity of the implemented approach.

Limitations of the developed prototype system are discussed

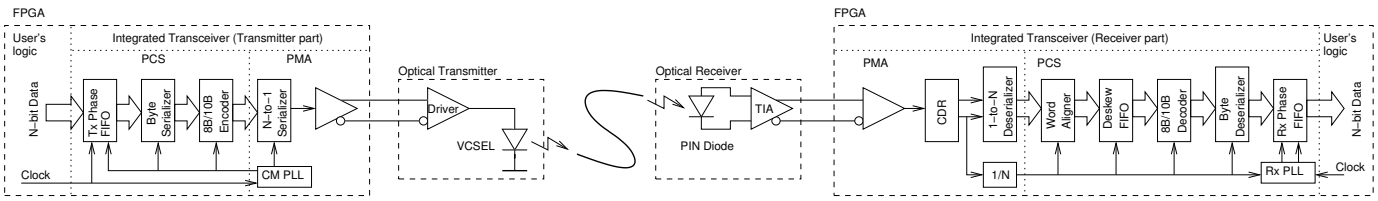


Fig. 1. Simplified inter-FPGA serial optical link structure.

in the next section. The discussion is followed by a presentation of changes made to the prototype during development of the second generation of the system. The changes to all system components from hardware to test software are shown along with the reasons behind particular design decisions. A novel optical cable connector compatible with the flat optical cables used in the system is presented in the next section. The connector provides an efficient mean to implement a full mesh connection between twelve boards.

An application of the developed hardware and system components to construct a reconfigurable computing platform utilizing optical interconnects between the FPGAs is discussed in the concluding section.

II. Related Work

Usage of FPGA for testing communication channels has been previously described. For instance, in [6] an implementation of the Bit Error Ratio Tester (BERT) based on the Altera's Stratix II GX transceivers is presented and compared with a commercial stand alone tester. It is shown that the results obtained with the FPGA implementation comply with the results of the stand alone tester. However, the implementation still utilizes external equipment to control the test system and to collect the measurement results.

This paper, while similar in overall approach to the one proposed by Xiang et al. [6], presents notable improvements in several areas. The most important of these is the implementation of a functionally complete test system inside the FPGA. Additionally, the flexibility of the implemented system allows extension of its hardware and software components to support interfaces for monitoring and controlling the parameters of the various components of the link without external equipment. Another improvement presented in this paper is an adaptation of eye-width as a link performance indicator instead of a raw BER. The eye-width can be measured significantly faster at low bit error probabilities with the aid of diagnostic circuitries integrated into the transceivers and therefore is more efficient as a target function for the parameter space exploration and link performance optimization.

III. Optical Link Structure

A block diagram of an optical digital communication link is shown in Figure 1. The link data path consists of a transmitter, an electro-optical converter (VCSEL with its driving circuits), an optical fiber, a photo detector (PIN diode and transimpedance amplifier) and a receiver. The transmitter and receiver are further divided into a Physical Coding Sublayer (PCS) and a Physical Medium Attachment (PMA) sublayer.

The PCS blocks are responsible for byte serialization/deserialization, byte ordering, rate matching, and 8B/10B encoding/decoding. All these functions are essential for the implementation of a reliable digital data channel. However, in this work, we concentrate on the physical layer performance measurements leaving the problems related to the coding sublayer out of the scope of the research.

The transmitter part of the transceivers integrated into the FPGA allows the tuning and run-time changes of several parameters. Among them are clock multiplication phase-locked loop (PLL) dividers and bandwidth, output driver common mode voltage, differential voltage output swing and preemphasis aimed at reducing the negative effects of inter-symbol interference. The receiver part, in turn, has the following tunable blocks and parameters: on-chip termination, adaptive equalization, decision feedback equalization, receiver input common mode voltage and gain. These blocks have a crucial impact on the signal quality on the input of the Clock and Data Recovery (CDR) circuitry, but their influence cannot be measured directly because the signal after these stages is not physically available outside the chip and cannot be connected to external measurement equipment. The CDR block provides a built-in diagnostic support circuitry to facilitate assessment of the signal quality on its input.

The hardware interfaces, which are necessary to change all the transceiver's parameters and to access the diagnostic circuits, are available to the logic programmed into the FPGA. Chip and design software vendors provide tools to access these interfaces, however, their use requires a connection between the development workstation with CAD software and the FPGA. The electro-optical components of the link have their own sets of tunable and monitoring parameters, such as driver and receiver power levels, VCSEL modulation and offset currents, temperatures and thermal compensation coefficients, signal power detected at the receiver input, etc. Access to these features is implemented through another set of vendor-specific interfaces and also requires a development workstation with a connection to the target system. Such connections may be not feasible in the embedded system while access to the interfaces is still highly desirable or even required. This problem may be addressed by integration of IP cores for all required management interfaces into the system instantiated in the FPGA.

The flexibility of a soft-IP microcontroller system inside

the FPGA allows the implementation of a single-point access to the management interfaces of all the components forming the link. Combined with built-in link diagnostic capabilities controlled by the same microcontroller system it results in a complete test system that enables link performance testing and parameter tuning without relying on any external equipment. Additionally, it is available not only during development and testing of the system but also after its deployment.

IV. Link Performance Indicators

Two link operation quality indicators are introduced in this section along with a description of an algorithm used by the authors to measure "eye-width" with the transceiver's built-in diagnostic circuits.

A. Bit Error Ratio

The integral quality of operation of a serial link is characterized by its Bit Error Ratio (BER): a ratio of the number of bits received with errors to the total number of bits transmitted through the link: $BER = N_{err}/N$. This ratio is used for both measured and actual values. A BER is usually measured with a special piece of test equipment, so called Bit Error Ratio Tester. It consists of a data pattern generator, a reference quality receiver, a digital comparator and counters for transmitted bits and errors. The flexibility of an FPGA allows to implement all blocks of a bit error ratio tester in programmable logic in the FPGA itself.

If single bit errors in a serial link may be viewed as independent events and conditions do not change over time, then the actual BER value is a probability of a single bit error (p_e) and the measured value approaches the actual BER in the limit: $\lim_{N\to\infty} N_{err}/N = p_e$. It is not possible for BER measurement to transmit an infinite number of bits since it would require an infinite measurement time and a way to measure the BER with a given accuracy is required. For practical application it is often enough to know that the BER is below some threshold with a given confidence while its actual value is irrelevant. As the literature shows (for instance, in [9]), if more than N_0 bits were transferred during the test with no errors detected, then with probability α the actual BER is less than p_e:

$$N \geq N_0 = \frac{1}{p_e} \ln \frac{1}{1-\alpha}$$

This number of bits (N_0) sets a lower limit on the test duration when no errors are observed. At a data rate of 5 Gbps it takes approximately 10 minutes to reach a 95% confidence that BER is lower than 10^{-12}, for the BER level of 10^{-15} it would require almost a week. The long runtime required makes it impractical to use the BER directly as a target function for the link parameters optimization. It would take enormous amount of time to find an optimum in the parameter space even if only a small fraction of all possible parameter combinations yielded a bit error ratio lower than 10^{-12}.

B. Eye-Width and its Measurement

The quality of a signal may be analyzed by evaluating its eye diagram: a picture on an oscilloscope display resulting from observing a transmission of a pseudo-random binary sequence with properties representative of the physical layer encoding used in the link. The width and height of an opening of the central part of the diagram ("eye") serve as indicators of the signal quality and may be used as target functions for the link parameter tuning. However, the signal on the input of the receiver CDR unit is not available for direct measurements. Therefore built-in diagnostic circuitries of the receiver should be utilized.

Serial transceivers integrated into the Altera Stratix IV GX FPGAs include special circuitry that facilitates measurements of the eye opening on the input of the CDR block [10]. The circuitry allows shifting of a sampling point of the signal from its optimal position in the center of the unit interval (UI) under external control. Then bit error ratio is measured for each phase offset. For sampling points close to the center of the eye opening, there will be no significant increase in the bit error ratio. For sampling points closer to the signal slopes the number of observed errors will gradually increase. Finally, in the area of the signal edge crossing widened by a jitter, a receiver will not be able to achieve synchronization with its input signal resulting in the observed bit error ratio of 0.5. From these measurements of the BER at signal sampling points distributed through the UI the eye opening and jitter characteristics of the signal may be deduced [9].

The key benefit of this approach is that the conclusion regarding the signal quality and, therefore, link parameters, may be reached by a number of BER measurements with different phase offsets through the UI instead of one at the optimal sampling point. However, each of these measurements needs to achieve a given confidence level at a much higher target BER and, therefore, requires significantly shorter runtime.

An algorithm implementing this approach can be further optimized to reduce the number of required BER measurements at the center of the eye opening, where the bit error ratio is low. These measurements take up most of the time and effectively provide no useful information. Several approaches to such optimization are described in [9].

Figure 2 illustrates the behavior of the modified algorithm implemented by the authors and shows an eye-diagram reconstructed from the measurements. As a first step (marked with 1 in the figure) an initial scan through the entire unit interval is performed with high target BER (10^{-7}). From these measurements, an approximate location of the eye boundaries is determined. At the second stage the BER is measured at the center of the eye opening to make sure that the target BER level (10^{-12}) is achievable at the close-to-optimal sampling point (2). Then, the BER is measured at sample points from the eye opening boundaries detected during the first scan towards the center to determine points where the target BER level is achieved (3). The distance between these points (eye-width) serves as a measure of the signal quality at the input of the

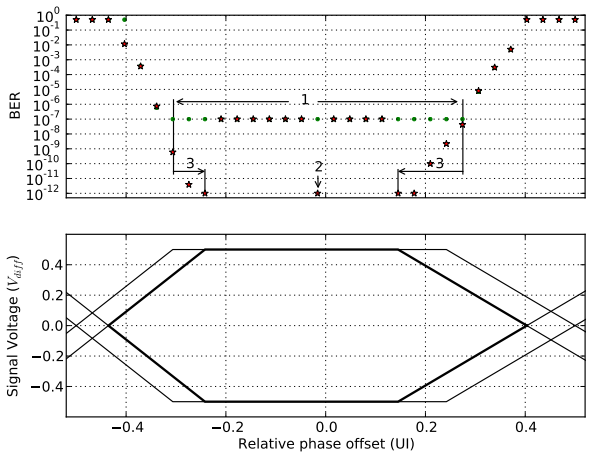


Fig. 2. "Bath-tub" curve scan algorithm and reconstructed eye diagram.

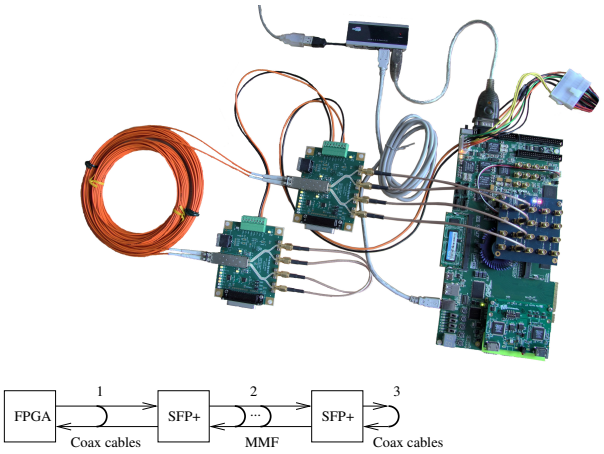


Fig. 3. Experimental system and loopback configurations.

receiver CDR unit and may be used as a target function for the link parameters' tuning.

The described algorithm for eye-width measurements reduces the number of BER samplings within the eye opening. For the diagram shown in Figure 2, it took only 55 minutes to collect all the data. An exhaustive UI scan under the same conditions takes 150 minutes but provides no additional information on the link operation.

V. Prototype System Implementation

To confirm the usefulness of the approach described to the optical link testing and parameter tuning and to create a base set of tools and building blocks to be used in future projects the authors implemented a prototype system. The system consists of hardware, a set of IP blocks, embedded software and development tools and facilitates debugging, testing and evaluation of the components. A photo of the assembled system hardware is shown in Figure 3 and components of the system are described in the following sections.

A. Hardware Platform

The system is based on the Altera Stratix IV GX FPGA (EP4SGX230KF40C2) installed on a TerasIC DE4 board. Through an adapter board with SMA connectors and a set of coaxial cables the DE4 board is connected to SFP+ evaluation boards hosting optical transceiver modules. Hot-pluggable SFP+ transceivers used in the system provide duplex LC-type optical connectors for the Multi-Mode Fiber. Management interface of the transceiver modules (I^2C) is accessible from the FPGA and is used for the monitoring of their parameters.

The highly modular construction of the hardware platform enables experimentation with different components and link configurations. During development and validation of the system several loopback configurations were used as shown in the diagram on Figure 3. The shortest possible one is an electrical loopback connecting the FPGA transmitter output signals directly to the input of the receiver (1). The second tested configuration uses a single optical transceiver with its input and output connected via a Multi-Mode Fiber (MMF) loopback (2). The length of the fiber loop used in the tests ranged from 15 cm to 15 meters. This loopback configuration is the closest to an actual optical link where the signal passes through one electro-optical and one opto-electrical conversion and a single fiber segment.

The most elaborate loopback configuration tested utilizes two transceiver modules and an electrical loopback on the "remote" side of a duplex fiber link (3). While this link exceeds configurations, which would be found in practical applications it is still interesting as it allows an easier separation of influence on the signal quality from different components of the link and serves as a model of a less favorable environment with longer links and a higher number of interconnects along the signal path.

The transceivers available in Stratix IV GX FPGA provide an on-die scope capable of $1/32$ unit interval resolution at data rates up to 6.5 Gbps [10]. Comparable technology is available in the transceivers integrated into the Xilinx Virtex-6 FPGA family. As an additional feature these transceivers are capable of a vertical scan of an eye-diagram [11], however, this functionality has not yet been explored by the authors so far.

B. System-on-Programmable Chip and IP Cores

The architecture of a soft-IP microcontroller system instantiated in the FPGA is shown in Figure 4. The system consists of the following main blocks: NIOS II CPU core with a small on-chip ROM containing boot code, a controller for external SRAM and FLASH, UART for communication with a control terminal, cores for the test pattern generator and checker, interfaces to access the transceiver configuration and diagnostic features, I^2C master cores for connection to the management interface of the SFP+ modules. The entire system utilizes only a small fraction of the available FPGA resources: the logic utilization is 3%, and available memory and DSP blocks are used for less than 1%.

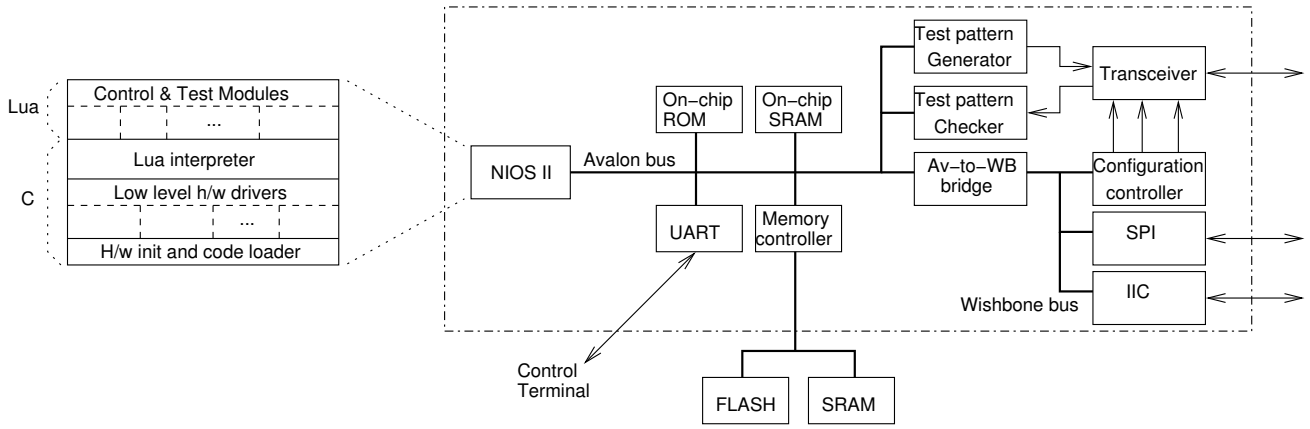


Fig. 4. Test System-on-Programmable Chip (SoPC) architecture.

The IP cores forming the system were taken from three sources. The first one is the library supplied by the FPGA vendor (Altera in this case). The cores are optimized for a specific FPGA architecture, but no source code is provided and the cores are not available on FPGAs from other vendors. The second source of IP cores for the system is a collection of free and open cores hosted on the OpenCores site [12]. These cores are provided under free licenses and their source code is available. This makes it possible to implement these cores in systems on different FPGA architectures. The price for such flexibility is the time and effort required for integration and adaptation, and the required time and effort is generally greater than for FPGA vendor supplied IP cores.

These two sources of IP cores, while covering most of the functionality, still do not provide several crucial interfaces required in order to access transceiver configuration and diagnostic interfaces. These missing parts were created by the authors by means of custom HDL development as the third source of IP blocks, and this required most effort.

Since the IP cores from different sources have different interfaces their integration into a working system is a technical problem in itself and required the development of "adapter" modules. The two primary on-chip interconnects used in the system are Avalon [13] and WISHBONE [14].

Overall, a combination of the readily available blocks (both proprietary and free) and those developed in-house proved to provide a reasonable and time efficient way of implementing the prototype system.

C. Embedded Software

The monitoring and control of all blocks forming the optical link, BER testing and processing of the test results are handled by an embedded software running on the NIOS II soft-IP CPU instantiated in the FPGA.

Low level software to access all hardware interfaces is implemented in the C programming language and its functionality is made available to the Lua interpreter. Lua, as is stated on its web-site [15], "is a powerful, fast, lightweight, embeddable scripting language". These properties make it very attractive for a wide range of applications including game development, mobile devices and embedded software [16]. A tight integration with C and an interactive interpreter facilitate an efficient development of diagnostic, testing and debugging software for embedded hardware systems.

Availability of an ANSI C compiler and a basic C run-time library are the only requirements to port Lua to a new platform and it was extremely easy to get an early prototype running on NIOS II. The efforts invested in the porting and support of Lua interpreter on the soft-IP microcontroller system in the FPGA were rewarded in the flexibility of the resulting system and increased development productivity.

Access to the interactive environment is very useful during embedded hardware development and debugging as it saves a lot of time in the edit-compile-load-run development cycle. Since the "hardware" itself is a soft-IP system instantiated in the FPGA this time saving becomes even more important: on the one hand, the system is malleable and experimental and includes design errors, on the other hand, traditional software development cycle is complicated by a separate FPGA design flow with longer iterations. With this additional complexity an availability of tools facilitating quick experiments and tests running directly on the target platform is a key factor for effective development. Our experience shows that Lua fits this role perfectly and allows rapid localization of the design errors both on the hardware and software levels. All the link configuration and BER measurement software in the system are implemented as a set of Lua modules.

VI. MEASUREMENT RESULTS

Measurements on the test system were performed for data rates in a range from 1 to 5 Gbps with various loopback configurations. The SFP+ module used in most experiments is the Avago AFBR-703SDDZ. The module is capable of data rates up to 10 Gbps and, as expected, performs excellently in the tested data rate range. Even with the most demanding loopback configuration the eye diagram opening for the 10^{-12} BER level is approximately 40% (80 ps) of the unit interval (200 ps at 5 Gbps).

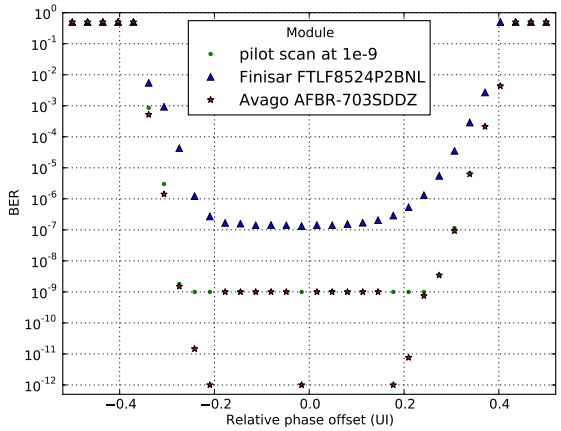


Fig. 5. Comparison of "bath-tub" curves for two SFP modules at 5 Gbps.

Several data patterns with different spectral characteristics were used in the experiments. Two test patterns that specifically check the link performance at the edges of its frequency band are the Low Frequency (LF) and High Frequency (HF) patterns. The other test patterns are Pseudo-Random Binary Sequences (PRBSx) generated by a linear feedback shift register with the length x. The lengths of 7, 15, 23, and 31 bit were used. The test results show slight dependency on the data pattern used, however, detailed analyses of this dependency have not yet been yet performed.

To validate the test system and confirm that the measurement results adequately represent link quality an SFP module with a lower maximum data rate has been used: Finisar FTLF8524P2BNL. According to its documentation the module is capable of data rates up to 4.25 Gbps. Experiments show that up to this limit it demonstrates BER $\leq 10^{-12}$, also the eye width is smaller than that with the Avago module. The bathtub scan results for both modules at 5 Gbps are shown in Figure 5. This data rate is outside of the specified range for the Finisar module and this is clearly visible from the diagram: even in the vicinity of the ideal sampling point BER does not achieve 10^{-7} level.

The results obtained allow the conclusion that the developed test system provides reliable data on the optical link performance and may be used to compare different link implementations and to tune parameters of the link. The comparison of the measurement results obtained with different data patterns may provide additional information that could be useful for optimizing link performance.

VII. Second Generation of the System

While the developed system proved the feasibility of the selected approach and served as a convenient platform for experiments with the optical links and IP cores, it also had severe limitations, which made further usage of the system and its components problematic. The following sections describe these limitations along with the changes implemented by the authors in the second generation of the system to overcome the discovered limitations.

A. Hardware Platform

The most obvious problems of the prototype platform at the hardware level were a limited number of the supported optical channels and a low integration level leading to a number of separate boards interconnected with a web of cables. While being beneficial at the early stage of the project enabling fast system setup and reconfiguration times, this approach does not scale well beyond simple desktop setup with one optical link. The most elaborate system configuration theoretically achievable with this approach would have eight optical links and require eleven boards for loopback configuration only. The situation would be even worse considering the interconnection of several FPGA boards. The other limitation of the prototype hardware platform is the unavailability of the optical transceivers' current consumption monitoring.

Recent advances in the optical transceiver technology enable higher level of integration and power efficiency than achievable with the SFP+ modules used in the prototype system [17]. To benefit from these improvements in the technology the authors designed an add-on card for the DE4 FPGA board. The add-on card mounted on top of the DE4 FPGA board with flat optical cables connected is shown in Figure 6. The card hosts Avago transmitter and receiver MiniPOD™ modules and connects them to the FPGA serial transceivers making use of all twelve channels available on the DE4 board extension connectors. The transmitter and receiver configuration and monitoring interfaces are connected to the FPGA as well. In addition to the management functionality integrated into the modules, the board implements circuitry to individually monitor power consumption of the modules on all power supply rails.

The developed add-on card makes twelve duplex optical links available for the FPGA without any additional boards or electrical cables connected. The transceiver and receiver modules make use of a flat ribbon optical cable to implement a high-density optical connection.

Two FPGA boards may be directly linked with ribbon optical cables to implement twelve duplex communication links. Extending the system to higher number of nodes would require more complex optical interconnection. One specifically important type of the interconnect is a full-mesh network where each node has a direct link to every other. A novel matrix optical connector presented in [18] implements a crossbar interconnect for twelve nodes.

The realization of an 4×4 crossbar interconnect is illustrated in Figure 7. A connector uses two fiber-matrix plates that are rotated by $90°$. Each 2D fiber-matrix plate combines four 1D fiber bundles. Because of the $90°$ rotation of the matrix plates, columns at the input side are connected to rows at the output side. Each output fiber bundle is connected to every input layer. Therefore, all the combinations of inputs and outputs are realized as required by the crossbar scheme.

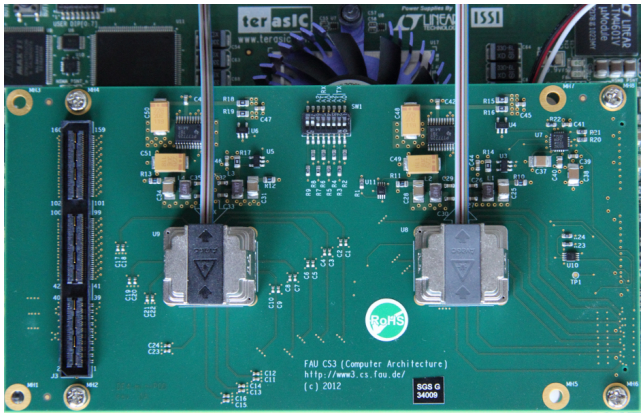


Fig. 6. MiniPOD extension board mounted on top of DE4.

Despite the simplicity of the interconnection scheme, the realization of the fiber-matrix poses significant challenges. In Figure 8 a 12×12 connector is shown. It includes a matrix with 12×12 multimode fibers, each with a core diameter of 50 μm. The distance between the single fibers is 250 μm and the total area of the matrix is 2.85×2.85 mm^2. The difficulty in making large connectors is given by the requirement to drill high number of holes with tight position tolerances. A detailed analysis of the fabrication procedure may be found in [19]. This crossbar optical connector with 144 channels in total is available as a so called CrossCon®-device from the company EUROMICRON.

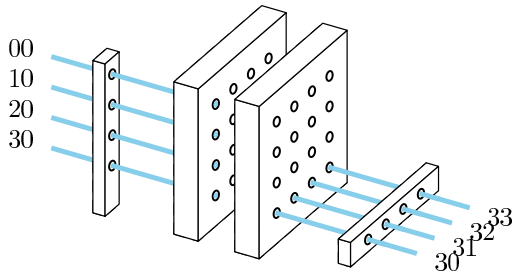


Fig. 7. Schematic of the novel 3D fiber optical crossbar approach

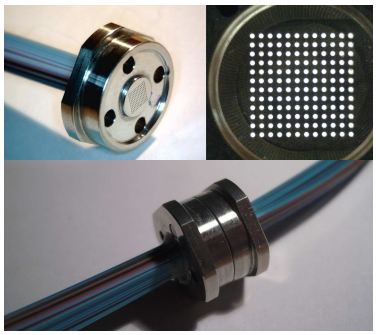


Fig. 8. a) Prototype of a 12×12 fiber connector. b) Picture of the fiber matrix with 144 channels. The position tolerance of the holes is ± 3 μm. c) Resulting crossbar interconnection.

B. System-on-Programmable Chip and IP Cores

Several factors limit a wider application of the System-on-Programmable-Chip solutions. One of the most critical is the utilization of FPGA vendor specific IP cores. To use the test system on an FPGA from a different vendor these blocks should be replaced with their functional equivalents available on the other platform, but supporting different system variants would increase the effort required. A more efficient approach is to replace the vendor specific IP cores with free and open-source equivalents available on different target platforms.

The most complex and important block in the prototype system specific to the Altera platform is the NIOS II CPU core, so the authors decided to replace it with one of the free and open-source CPU cores available. Several cores have been considered as a replacement candidates and the choice was made for LEON3 processor from Aeroflex Gaisler AB [20]. The LEON processor is "a 32-bit synthesisable processor core based on the SPARC V8 architecture. The core is highly configurable, and particularly suitable for system-on-a-chip (SOC) designs." The LEON3 core is distributed as a part of GRLIB IP library. The library also includes many communication peripheral controllers, configuration utility, and several pre-configured design examples. The configuration utility allows setup of various LEON3 core parameters (such as size of a register window, cache type and sizes, etc.) and configure system peripheral devices. At last but not least the system include Debug Support Unit (DSU), which allows external connection to the system via various interfaces (JTAG, serial, ethernet network) and on-chip software debugging access. Availability of this functionality is invaluable during the early firmware porting.

The remaining proprietary cores (test data pattern generator and checker, external bus controller) are easier to replace and do not require toolchain and embedded software porting efforts. The work on their replacement with equivalents available as open-source or developed by the authors is underway.

The modified architecture of the System-on-Programmable Chip is presented in Figure 9. Other than processor core replacement and associated changes in the peripheral devices and connection bus architecture there is one other notable change compared to the prototype system. To reduce FPGA resource usage and power consumption associated with high-speed circuitries and to simplify time-closure of this part of the design without compromising the ability to test all twelve serial communication links the decision has been made to provide just one data pattern generator and checker and implement a daisy-chain connection for the serial transceivers on their digital data interface to the FPGA. In this configuration test data received on the first transmitter channel (which are the data sent on this channel and returned via optical loopback) are retransmitted on the second channel and so forth. A parallel multiplexer allows to select a transceiver channel to be connected to the data pattern checker.

The synthesis results for the system with the new processor core and modified architecture show that the FPGA resource

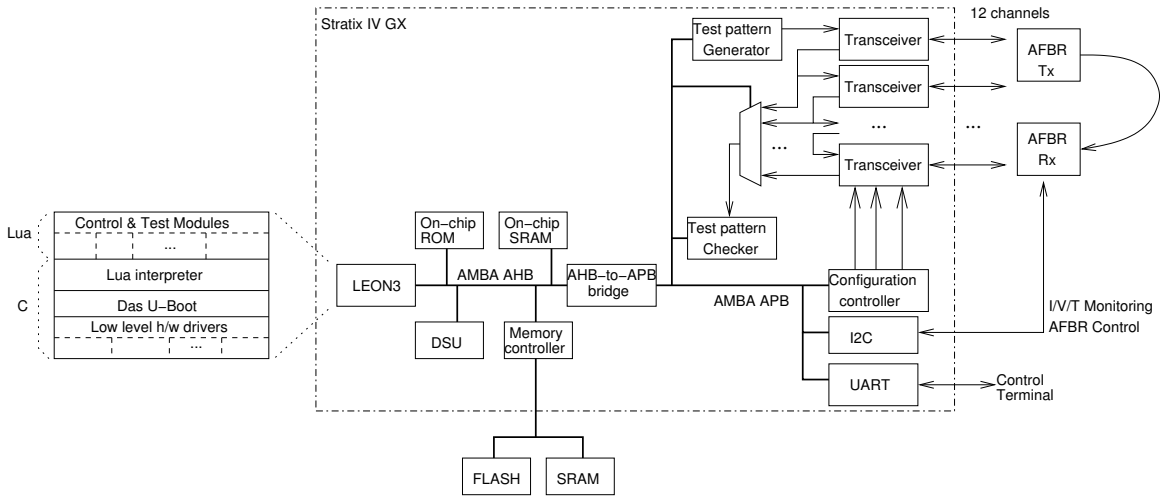


Fig. 9. Second generation of the SoPC architecture.

usage has not been significantly affected by the changes and remains below 5%. That makes it possible to keep the test system in the FPGA configurations for different applications and to use the embedded LEON3 processor for management, debugging support and performance monitoring tasks in the application system.

C. Embedded Software

The chosen approach to implement embedded software for the SoPC on the interpreted language facilitating fast and interactive development directly on the target system proved to be an efficient one. That is especially notable in the areas where a non-standard interfaces or functionality have to be implemented or there is a need to experiment with different algorithms or test various approaches with a quick prototype code.

On the other hand, this approach requires to re-implement a lot of common support functions (e.g., filesystem and networking), which come for granted with a "standard" embedded software development tools and frameworks. The switch to a advanced processor core also requires additional efforts to be invested in development of the general purpose utility code such as processor boot, cache and interrupt management, etc. Anticipating utilization of the LEON3 processor inside the system for other tasks and the need for software support for a number of standard interfaces and protocols when the boards are integrated into a system the authors decided to introduce an extra level into the system embedded software.

This new level of software is to replace custom written processor bootstrap code, initialization and drivers for generic hardware interfaces and basic C runtime library from the prototype system. An industry standard open source firmware for the embedded platforms with a wide range of supported processor architectures currently is "Das U-Boot" [21]. It implements all low-level processor and interface management tasks and allow load and debugging of application software from different medias.

While the primary goal of U-Boot in an embedded system is to load application software (often it is a Linux kernel along with the initial ramdisk image) from external storage or over a network interface and pass control to it, the firmware provides and Application Programming Interface (API) for so-called "U-Boot standalone applications". These applications are loaded dynamically and can have an access to the U-Boot console I/O functions, memory allocation and interrupt services. The Lua interpreter with serial links and system control, monitoring and test software plays the role of such standalone application.

The modified embedded software architecture allows to combine positive sides of the interactive interpreter with custom experimental software running on the target system with an industrial-strength support for multiple interfaces, protocols and debugging capabilities coming with a standard cross-compiled firmware.

VIII. Conclusion and Future Work

The implementation of the optical link test system clearly demonstrated the feasibility and effectiveness of the proposed approach to utilization of the on-chip diagnostic capabilities of FPGAs with high-speed serial transceivers. The use of the soft-IP controller instantiated in the FPGA allows a single-point access to the control and diagnostic interfaces of all components forming the link. Combined with computational capabilities and a high-level programming language interpreter running inside the FPGA, it enables extensive optical link performance evaluation without relying on any additional test and measurement equipment and significantly shortens the system debugging and testing times. As an additional benefit all the implemented functionality is still available in the deployed system and may be used for remote monitoring and diagnostics.

Detailed analysis of the dependencies between the test loopback configurations, data patterns, transceiver parameters and observed eye-diagram is required to develop effective

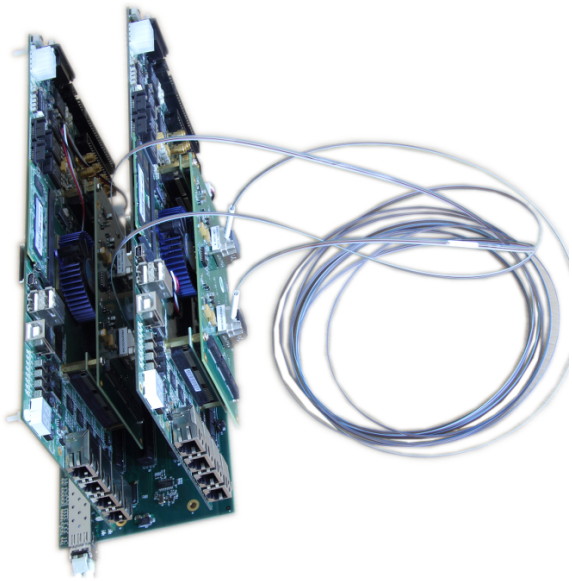


Fig. 10. Two DE4 boards cross-connected with parallel optical links.

algorithms for link parameters tuning. Current consumption monitoring hardware integrated into the system facilitates measurements of the system power efficiency. This work provides efficient tools for these researches and demonstrates their feasibility.

Another area for improvement is the automated integration of separate IP blocks from different sources into a system. Vendor specific tools have progressed notably in this area in recent years, however, they are still limited with regard to support of "foreign" IP cores. On the other hand, while efforts have being made to provide similar functionality for free and open-source cores, the tools that have emerged so far are not well integrated in the FPGA and embedded software design flows. The authors' experience gained in course of conversion the system to open IP cores shows that despite availability of automated integration and configuration tools a manual intervention and hand-written RTL code are still required to combine IP blocks from different sources into a working system.

The developed hardware platform, IP blocks and embedded software form a base for integration of parallel optical links into a multi-FPGA reconfigurable computing system. Current hardware platform consisting of two DE4 FPGA boards cross-connected by twelve parallel optical links is shown in Figure 10. This platform is used for development of streaming video processing and HPC applications. When a framework for such application reaches some maturity and its hardware resource requirements exceed limits of the current platform, the next extension step is to switch from dual-board system to the multiple cross-connected boards configuration.

References

[1] A. Kuzmin and D. Fey, "Optical link testing and parameters tuning with a test system fully integrated into FPGA," in *The Fourth International Conference on Advances in System Testing and Validation Lifecycle (VALID 2012)*. IARIA, 2012, pp. 121–126.

[2] A. F. Benner, M. Ignatowski, J. A. Kash, D. M. Kuchta, and M. B. Ritter, "Exploitation of optical interconnects in future server architectures," *IBM Journal of Research & Development*, vol. 49, no. 4/5, p. 755, July/September 2005.

[3] S. Nakagawa, Y. Taira, H. Numata, K. Kobayashi, K. Terada, and M. Fukui, "High-bandwidth, chip-based optical interconnects on waveguide-integrated SLC for optical off-chip I/O," in *Electronic Components and Technology Conference*, 2009, pp. 2086–2091.

[4] B. E. Lemoff, M. E. Ali, G. Panotopoulos, E. de Groot, G. M. Flower, G. H. Rankin, A. J. Schmit, K. D. Djordjev, M. R. T. Tan, W. Gong, R. P. Tella, B. Law, and D. W. Dolfi, "Parallel-WDM for multi-Tb/s optical interconnects," in *Lasers and Electro-Optics Society (LEOS) IEEE Meeting*. Agilent Technologies Laboratories, 2005, pp. 359–360.

[5] O. Liboiron-Ladouceur, H. Wang, A. S. Garg, and K. Bergman, "Low-power, transparent optical network interface for high bandwidth off-chip interconnects," *Optics Express*, vol. 17, pp. 6550–6561, 2009.

[6] A. C. Xiang, T. Cao, D. Gong, S. Hou, C. Liu, T. Liu, D.-S. Su, P.-K. Teng, and J. Ye, "High-speed serial optical link test bench using FPGA with embedded transceivers," in *Topical Workshop on Electronics for Particle Physics (TWEPP)*, 2009, pp. 471–475.

[7] M. P. Li, J. Martinez, and D. Vaughan. Transferring high-speed data over long distances with combined FPGA and multichannel optical modules. [Online]. Available: http://www.altera.com/literature/wp/wp-01177-AV02-3383EN-optical-module.pdf [retrieved: May, 2014].

[8] G. Breed, "Bit error rate: fundamental concepts and measurement issues," *High Frequency Electronics*, pp. 46–48, January 2003.

[9] M. Müller, R. Stephens, and R. McHugh, "Total jitter measurement at low probability levels, using optimized BERT scan method," in *DesignCon*. Agilent Technologies, 2005.

[10] W. Ding, M. Pan, T. Tran, W. Wong, S. Shumarayev, M. Peng Li, and D. Chow, "An on-die scope based on a 40-nm process FPGA transceiver," in *DesignCon*. Altera Corporation, 2010.

[11] Virtex-6 FPGA GTX transceivers user guide. UG366 (v2.6). Xilinx, Inc. [Online]. Available: http://www.xilinx.com/support/documentation/user_guides/ug366.pdf [retrieved: May, 2014].

[12] [Online]. Available: http://opencores.org/ [retrieved: May, 2014].

[13] Avalon interface specifications. [Online]. Available: http://www.altera.com/literature/manual/mnl_avalon_spec.pdf [retrieved: May, 2014].

[14] Wishbone B4. WISHBONE System-on-Chip (SoC) interconnection architecture for portable IP cores. [Online]. Available: http://cdn.opencores.org/downloads/wbspec_b4.pdf [retrieved: May, 2014].

[15] [Online]. Available: http://www.lua.org/about.html [retrieved: May, 2014].

[16] R. Ierusalimschy, *Programming in Lua*. Lua.org, 2006.

[17] Micropod and minipod 120g/150g/168g transmitters/receivers. [Online]. Available: http://www.avagotech.com/pages/en/fiber_optics/parallel_optics/minipod_micropod [retrieved: May, 2014].

[18] U. Lohmann, J. Jahns, A. Kuzmin, and D. Fey, "Optical multi-Gbps board-to-board interconnection with integrated FPGA-based diagnostics," in *Optical Interconnects Conference*. IEEE, 2013, pp. 120–121.

[19] U. Lohmann, J. Jahns, T. Wagner, and C. Werner, "Ultra-precision fabrication of high density microoptical backbone interconnection for data center and mobile application," *Proc. SPIE Optics and Photonics*, 2012.

[20] [Online]. Available: http://www.gaisler.com/index.php/products/processors/leon3 [retrieved: May, 2014].

[21] Das U-Boot – the universal boot loader. [Online]. Available: http://www.denx.de/wiki/U-Boot/WebHome [retrieved: May, 2014].

Design Criteria and Design Concepts for an Integrated Management Platform of IT Infrastructure Metrics

Christian Straube, Wolfgang Hommel, and Dieter Kranzlmüller
Munich Network Management Team, Ludwig-Maximilians-Universität München, Leibniz Supercomputing Centre
[*straube,hommel,kranzlmueller*]*@mnm-team.org*

***Abstract*—Most measurement and metrics as they are used in today's IT management are not suitable for profound upper level management decisions. We identify four major gaps between the information that can be measured on a technical level and the information that is needed for management decision making: 1) The currently provided information is not suitable for decision making on higher abstraction levels. 2) Interdependencies between the metrics are not sufficiently considered. 3) There is no support for the derivation of improvement recommendations based on the metrics values. 4) Existing approaches lack the flexibility to incorporate organization-specific requirements. Based on state-of-the-art energy efficiency, performance, and security metrics taken from related work, we present how these gaps affect a complex real-world scenario. Consequently, we argue that an integrated management approach for IT infrastructure metrics is necessary and present the core components of our solution, referred to as the *management cockpit*. We therefore discuss its four-layered architecture, which deals with measurements and metrics, dependency handling, aggregation logic, and graphical representation as well as its information model backend. Finally, we present an overview of related work and give an outlook to open issues and future work.**

***Keywords*-Decision making support, measurements and metrics, integrated IT management, management tools, energy efficiency management, IT service management, service reporting**

I. Introduction

Contemporary *Information Technology* (IT) infrastructures are highly complex and mostly distributed artifacts, forming several specialized groups, like supercomputers, clusters, Grids or enterprise IT infrastructures. The ongoing development in the related areas, the by now short improvement cycles of the employed hardware and software, manifold business needs and the ever-changing environment of IT infrastructures, like changing customer demands or legal aspects, require a continuous adaption of the entire IT infrastructure and its sub parts. For instance, new security threats must be addressed, faster hardware is required to tackle strong competitors, or existing hardware turns out to be error prone.

Adapting an IT infrastructure to the above outlined situation and aligning it to the current requirements can be achieved by changes and modifications to the hardware, the software, and the configuration of an IT infrastructure. For instance, adapting CPU frequency to achieve a power consumption decrease or introduce redundancy to improve reliability are typical modifications. Each of them, however, must be thoroughly planned for two reasons [1]. First, planning is required to address the mostly business-critical dependency on IT infrastructures in nearly all areas, since IT infrastructures provide the foundation of IT services and IT-based business initiatives [2], [3], [4] a whole enterprise might depend on [3]. Second, planning should circumvent unnecessary adaptations and avoid their (mostly) costly investigation. A decision to purchase new hard disk drives for an entire cluster to improve reliability, for instance, requires a study covering potential manufactures as well as investigating the interoperability with the existing hardware. If it turns out that other components were much more unreliable than the replaced hard disk drives, the study (*investigation*) and the replacement efforts (*change*) were wrongly placed.

The important role of IT infrastructures for science and industry, the (potentially) severe impact of IT infrastructure changes to an enterprise's success and the power of changes to align an IT infrastructure to its surrounding cause an involvement of (upper and top-level) management in nearly every change planning and decision making process, especially strategic, large-scale, and cost-intensive decisions.

Sustainable decisions about the continuous improvement and future development of an organization's IT infrastructure should be based upon solid knowledge and database about an IT infrastructure's current state. The big importance of information for the decision making process has been investigated by current and comprehensive studies, e.g., "Big data Harnessing a Game-Changing Asset" conduced in 2011 by the *Economist Intelligence Unit* [5]. One of its key results: 90% of the decisions made within the last three years would have been significantly better, if (more) relevant information would had been available. Even if this study focused on retail and financial data, this is, in principle, also true for information about IT infrastructures.

Achieving *objective*, *transparent*, and *quantitative* decisions poses several requirements on the above mentioned solid database and its creation and maintenance, i.e., incorporating *measurement values* of selected characteristics and the employment of carefully chosen *metrics*. Over the past few years, multiple metrics have been specified by re-

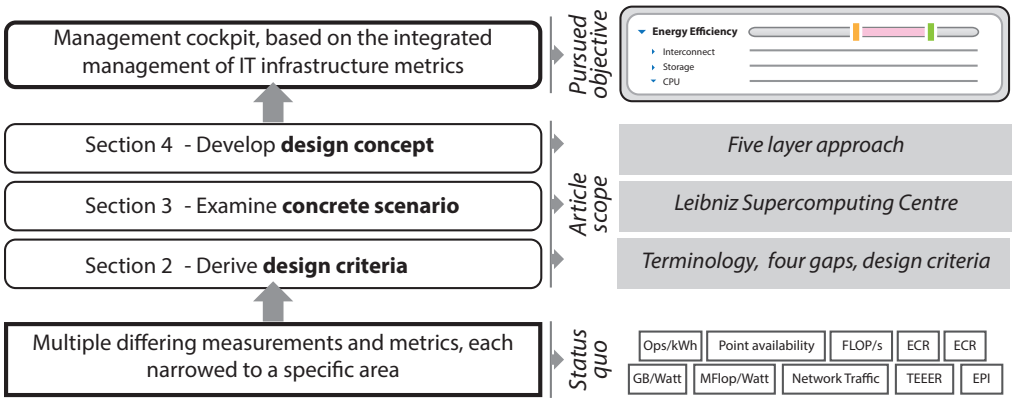


Figure 1. Scope of the article, i.e., the process of deriving requirements from a concrete (exemplary) scenario, resulting in a design concept for an integrated management platform of IT infrastructure metrics

searchers and practitioners for several IT management areas, including, among others, quality of service, performance, energy efficiency, and information security. Unfortunately, only few of them have been standardized, and most literature and real-world implementations focus on only a single one of these metric categories. For instance, metrics regarding energy efficiency on the one hand and information security on the other hand are rarely discussed by the same group of technical experts as of today.

In this context, a solid database for decision support can be achieved by an *integrated IT infrastructure metrics management*, emphasizing the holistic manner and exhaustive consideration of an IT infrastructure. The development of an integrated IT infrastructure metrics management faces several challenges and problems, especially when it comes to supporting high-level management decisions, in particular

1) the insufficiency of information provided by (low-level) metrics for decisions on higher levels,
2) the dependencies between measurements and metrics,
3) the incorporation of environmental influencing factors,
4) the lack of activity recommendations that can be deduced from metric values, and
5) the plethora of existing measurements and metrics at a lower abstraction level or rather at hardware-level.

This article presents a design concept for a *management cockpit* that addresses the above itemized challenges and aims at supporting top-level management decisions about planned changes based on a profound knowledge and database applying an integrated management of IT infrastructure metrics. In particular, the design concept describes a possible way to split the aforementioned extensive development problem in smaller parts and an architecture of implementation building blocks and their interactions. Additionally, the design concept describes guidelines and advice how to implement a particular part. In contrast, the design concept presented in this article does not provide concrete low-level implementations, e.g., a concrete aggregation rule. This is on purpose, since the concrete implementation of the outlined building blocks heavily depends on a variety of factors, like the objectives of the top-level management or the existing measurements and metrics.

In short-term, the management cockpit allows the visualization of raw measurements as well as derived metrics and shows their interdependencies to facilitate profound decision making. In long-term, it can also be used as a simulation tool for planning, optimizing, and choosing between mutually exclusive alternative options. This is of special importance for management decisions regarding investments in hardware, since unlike for the tweaking of software parameters, there usually is no viable rollback plan for hardware changes, such as upgrading CPUs or replacing HDDs, if it turns out that the performed change did not bring the desired effects.

Figure 1 depicts the article's scope and the discussed elements in the context of developing the outlined management cockpit. On the left hand side, Figure 1 shows necessary steps and section structure, on the right hand side it provides some detailing information. Since a systematic development process starts with analyzing *requirements* [6], [7], the process begins with a thorough consideration of the current situation, i.e., "multiple differing measurements and metrics, each narrowed to a specific area", depicted at the bottom of Figure 1. The findings are summarized in Section II in a set of design criteria, consisting of a basic terminology, four identified gaps, and some gap-spanning design criteria. To further substantiate the discussed design criteria and to provide a tangible example, Section III examines a concrete real-world scenario, the *Leibniz Supercomputing centre* (LRZ) and its High Performance Computing (HPC) system SuperMUC. Section IV presents a layered design concept to accomplish the above outlined management cockpit in general and to close the identified gaps in particular. It presents, among other topics, our information model for metrics, dealing with their inter-dependencies and visualization challenges. Section V discusses related work that has influenced our design; finally, Section VI concludes this article with a summary and an outlook to our next steps.

II. Gap analysis and design criteria

This section provides a *fundamental terminology*, identifies *four gaps*, and outlines gap-spanning *design-criteria*.

The fundamental terminology in Section II-A aims at ensuring a common understanding of important terms and concepts in the context of this article. This is required by the overload and differing meanings in different areas, like *metric* or *IT infrastructure*. Section II-B identifies a set of gaps that have to be filled or solved by the management cockpit development. The subsequent Section II-C outlines gap-spanning design criteria that have to be addressed during development, e.g., criteria for high quality measurements. Both, the four gaps and design criteria, act as *requirements* that start a systematic development process [6], [7]. For simplicity, gaps and design-criteria are derived and presented in a non-formalized way, e.g., without formulating *use cases* (cf. Jacobson [8]) and hence, the specific term requirement is omitted.

The four gaps and design criteria are substantiated for illustrative purposes in Section III, which examines the LRZ and SuperMUC.

A. Terminology

Almost all important terms used in the context of this article suffer from an overload. For instance, there are several definitions of *IT infrastructure* covering several focal points and granularity levels (cf. [9], [10]) but no definition is commonly accepted as standard or widely applied [11]. This is also valid for terms that are used in the context of assessing, characterizing, and valuating IT infrastructures, e.g., *metric*, *measurement*, or *key performance indicator (KPI)*. Despite a mass of literature about these terms (e.g., [12], [13]) there is no commonly accepted definition yet.

To address this situation and to avoid the risk of comparing and aggregating values with different meanings and intentions, the subsequent itemization provides a set of non-formal definitions. It is supported by Figure 2 and Figure 3, addressing *IT infrastructure* as well as *measurement* and *metric*, respectively. In contrast, we do not target a universal definition.

IT infrastructure – As motivated above, the article does not aim at developing or providing a long-term definition but a common terminology for the design concept presented here. Hence, we apply a very generic definition of *IT infrastructure* to cover as much situations as possible. In particular, the term "infrastructure" is composed of "infra" (lat. "beneath", "under") and "structure", and can be interpreted as "beneath the structure" [11, p. 36]. Despite the focus on information technology implied by the prefix "IT", *IT infrastructure* still might contain elements that are not considered, i.e., non-technical aspects [14], [10], like knowledge, skill-sets or *IT service management* (ITSM) processes, which can be summarized to the "human IT infrastructure" [4]. Furthermore, "IT structure" is interpreted as "IT system", i.e., "a set of things working together as parts of a mechanism or an interconnecting network" [15].

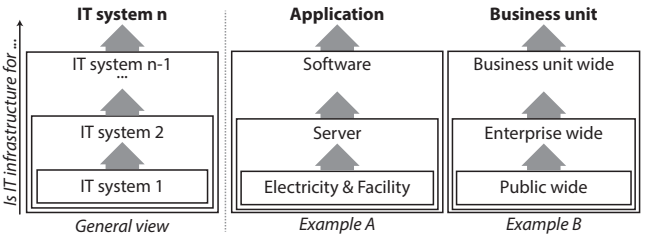


Figure 2. The term "IT infrastructure" is defined relatively to the applying or consumint IT system.

Summarized, *an IT infrastructure is the set of hardware resources that are necessary for running and using an IT system. Relations between set elements are built by functional dependencies and interactions*. Figure 2 illustrates the understanding relative to a given IT system: an IT system can be both, a consuming system and an IT infrastructure at the same time. On the left hand side, the generic layered pattern is depicted, on the right hand side two examples are given. Example A (taken from [11, p. 36]) emphasizes that an IT system can be both, consuming an IT infrastructure and providing an IT infrastructure at the same time. Example B (taken from [16, Figure 1]) illustrates the deployment at multiple levels. Especially example B endorses the relative understanding of IT infrastructure and that *the* one IT infrastructure can not exist and a relative view is mandatory [4].

Component type – Obviously, an IT infrastructure consists of manifold differing components. Enabling the assignment and selection of measurements and metrics requires a distinction of component types. For this, a component type is defined by a component's *capability*, i.e., a well-defined low-level functionality, like computation, data storage, and data transfer, that is exposed to a user or application [17]. Hence, exemplary component types are "data transfer" or "storage". These types can be extended and defined individually for a particular scenario.

Measurement – The considered (real world) objects own an arbitrary set of characteristics that describe the object, summarized as *facts* on the left hand side of Figure 3. In order to enable a reasonable processing of the extensive set of facts, a *measurement* abstracts these facts by reducing information complexity [18], [19] and mapping the resulting remaining facts onto a symbol set, which enables the execution of mathematical functions [20] (cf. Figure 3). Measurement results in a set of *measurement values*, depicted at the middle of Figure 3. There are three types of measurement values, i.e., *simple* values – directly compiled by a measurement, *additive* – compiled by adding two or more simple measurement values, and *derived* – compiled by applying a more complex operation on a set of simple or additive measurement values.

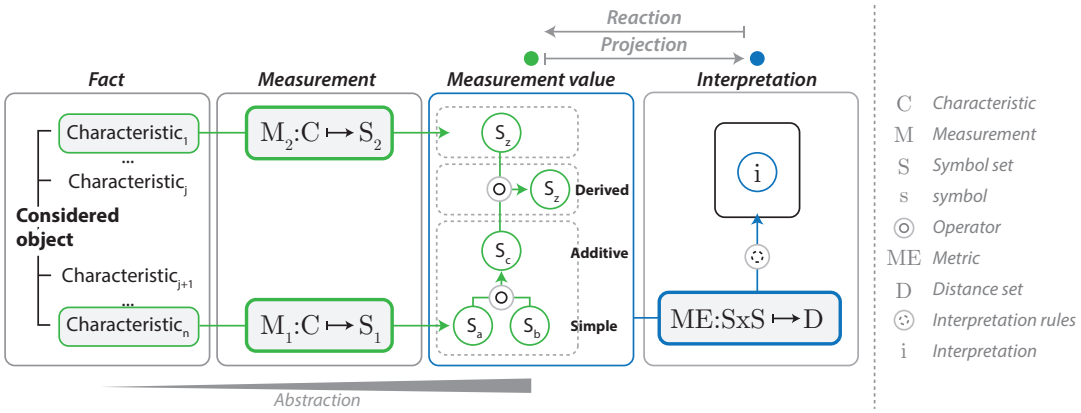


Figure 3. Process from facts that describe a real-world object to an interpretation that is used for top-level management decision making.

Metric – As depicted in Figure 3 on the right hand side, a metric is a distance function, i.e., a function that maps two measurement values $S \times S$ onto a (numerical) distance DV. A metric is required to satisfy four conditions, i.e., *non-negativity*, *identity of indiscernible*, *symmetry*, and *triangle inequality* [18]. In contrast to the (optimally) objective nature of a measurement, a metric creates correlations and evaluation.

Interpretation – An interpretation finally defines the semantics that are relevant to the top-level management decisions. In particular, an interpretation consists of a metric and a finite set of interpretation rules that can be applied on the compiled distance. Consequently, an interpretation evaluates correlations between measurement values. For instance, the distance "20 Bit/second" is enhanced by a good/bad assessment.

Key Performance Indicator (KPI) – A KPI is a specifically selected interpretation of subjective importance.

B. Metrics-based management decision-making gaps

This section identifies and discusses gaps that have to be filled or solved by the management cockpit development. Figure 4 overviews them as well as their general correlations and context. At the bottom of Figure 4 are the existing low-level measurements and metrics, exemplarily represented by some (arbitrarily) selected metrics that consider the energy efficiency of interconnect, storage, and CPU hardware types of HPC IT systems. The aggregation of these low-level metrics and their enhancement for top-level decision support is covered by *Gap 1 – The information gap*. The handling of potential correlations and dependencies between two or multiple metrics is covered by *Gap 2 – The dependency gap*. To further support top-level management in the decision-making process, not only a profound information base is required, but also the (automatic) derivation of recommendations how the top-level management should react on aggregated values and how modifications should be executed and achieved. The thereby arising challenges are covered by *Gap 3 – The activity gap*. Furthermore, the implications and influencing factors caused by the considered IT infrastructure's surrounding range from formal aspects like national law to immutable electricity prices or the housing building. All these elements are covered by *Gap 4 – The environment gap*.

The gaps are subsequently detailed further.

Gap 1 – Information Gap – This gap can be considered as the main gap, as it reflects the fact that information from existing low-level approaches are not available at higher abstraction levels, where it would be necessary to create a comprehensive holistic view. Without the information from low-level approaches, incomplete and incorrect information has to be used to make strategic decisions, e.g., decisions regarding the energy efficiency of the IT infrastructure for a procurement decision. This gap becomes even more severe in the context of the commonly accepted management principle that an activity cannot be managed if it is not measurable [13].

The information gap describes the transformation from the existing measurement and metric values to a (small) set of aggregated values for the top-level management. Available information often is unsuitable for the target audience. For instance, hardly any top manager will be fond of making a decision about which new server CPU hardware to invest in given a metric such as *MegaFlops per Watt*, even if this is an interesting energy efficiency metric for a technician. For a holistic view, the (purely) technical information has only limited expressiveness and must be enriched by context and comparison information.

The information gap covers two questions, i.e., *what* should *how* be aggregated. Additionally, all this information has to be aggregated to provide comprehensive information to support decision making at high level. Therefore, conversions, e. g., into currencies or hours of work, may be required.

The first question about the **what** addresses the selection of a set of metrics for aggregation. Usually there are several measurements and metrics, each focusing on a (slightly) different aspect. Hence, in a very first step it must be

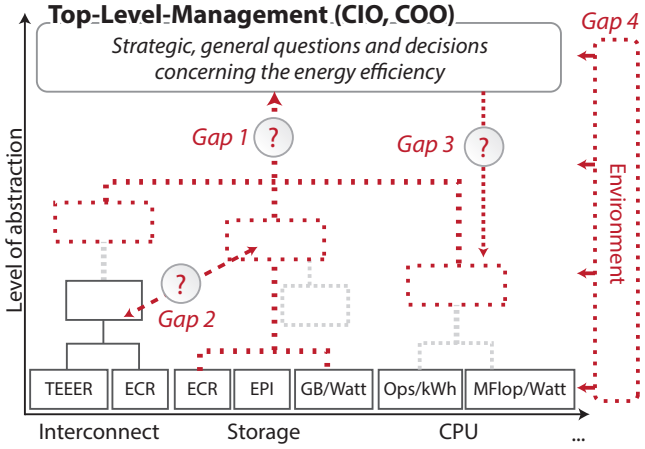


Figure 4. The four gaps in today's situation that hamper a holistic view.

decided, which existing values should be aggregated, which metrics are or could be relevant for the currently considered question, and if each value should be respected with the same priority.

The second question about the **how** covers the differing measurement and metric scales as well as the aggregation function to apply. Not all metrics can be applied to the same component types (cf. Section II-A). For instance, a CPU/GPU metric like *MFlops/Watt* cannot be applied to storage, interconnect, or software components.

Gap 2 – Dependency Gap – The second gap addresses misleading results and interpretation if measurements and/or metrics are considered separately without respecting dependencies and hence, without including all relevant information. Additionally, it mostly is not adequate to improve only one or a small set of metrics, i.e., partial optimization does not yield optimum results. Instead, all (involved) metrics should be improved [21], correlations have to be covered, and conclusions have to be drawn from these correlations [22]. Besides the big advantage of respecting the reciprocity of metrics, considering metric dependencies facilitates the uncovering of strategic goal conflicts [23] and the consideration of trade-offs, for instance, between energy efficiency and performance. Those trade-offs are already analyzed in some lower-level-approaches, e.g., by Rossi et al. [24], but not yet at upper-level management.

Dependencies can be split in *intra-* and *inter*-metric dependencies. **Intra**-metric dependencies address dependencies within a single metric group. In the realm of performance, for instance, an IT infrastructure is determined not only by the computing cores, but also by the communication, interconnect, and I/O performance [25], [26], [27]. **Inter**-metric dependencies cover dependencies between metrics of different groups. For instance, increasing hot-standby redundancy to address short-time breakdown and to improve reliability [28], simultaneously increases energy consumption and degrades performance due to redundancy overhead [29]. Another example is the trade-off between a new type of network component might have a better energy efficiency, but worse performance than a competitive product. Both products may also have different security properties and investment cost.

Gap 3 – The Activity Gap – The provision of a profound knowledge and database to the top-level management as motivated in Section I is covered by the already discussed Gap 1 and Gap 2. Both represent a big step towards a management cockpit based on the integrated management of IT infrastructure metrics. Nevertheless, the values resulting from metric selection and aggregation mostly require a focused activity from top-level management. Hence, Gap 3 questions how to react on certain aggregated values and implications. In other words, in order to perform the best adoptions on the analyzed IT infrastructure, activity recommendations have to be generated out of the holistic view in a (semi-) automated way. It is not sufficient to define or use a set of metrics in order to evaluate a situation. Instead, there are additional information required to assess the impact of a particular value [21]. Figure 3 depicts this as *interpretation* that triggers a feedback mechanism [19].

Recommendation compilation faces two challenges, i.e., *cost optimization* and *metric reciprocity*.

As outlined in Section I, each modification or rather its execution causes a variety of cost, ranging from preparatory investigation to implementation related cost. Additionally, there are mostly at least two potential reactions on an aggregated value and its implications. For instance, an alarming low reliability could be addressed by either introducing redundancy or by replacing the unreliable elements. The caused costs, respectively, have to be considered during recommendation compilation to achieve results as good as possible.

The complexity of contemporary IT infrastructures renders the separation of the specific contribution of a particular infrastructure component very difficult [30], [31]. Additionally, effects induced by local modifications on a single component can quickly and easily cascade and affect the HPC infrastructure partly or completely [32]. Activity recommendation compilation is required to consider effect cascading in order to avoid unpredictable issues.

Gap 4 – The Environment Gap – This gap is orthogonal to the different abstraction levels and metric dependencies discussed in the last three gaps as Figure 4 illustrates, and hence, the environment gap affects all other gaps. In particular, the environment gap affects *metric semantics* and covers *external factors* that might influence measurements and compiled values as well as factors that (obligatorily) have to be addressed or covered in an abstraction level spanning way.

The **metric semantics** states that the same measurements and metrics may not have the same expressiveness and purpose in different scenarios. Each organization might have

to make specific adoptions to existing metrics, create complementary metrics for its specific environment, and specify, for example, how results must be interpreted properly. This is of special importance to the activity gap (cf. above), since result interpretation and implications might have tremendous impact on the taken actions and made decisions. For instance, energy consumption figures might be alarming in a location facing high electricity prices, whereas the same numbers might have low or no importance in a location benefiting from low electricity prices.

The **external factors** cover a diversity of elements, like the building that houses the IT infrastructure or the national law, which guides a company's compliance. The integration of these external factors heavily depends on the specific objectives of top-level management. For instance, if a holistic energy efficiency investigation is pursued, the *Power Distribution Unit* (PDU) must be covered as well as the building and supporting infrastructure, like cooling.

C. Gap-spanning design criteria

There are some important aspects that affect all mentioned gaps and that can be considered as the *non-formal* requirements to fulfill.

Not delimited set of metric groups – Closing the above detailed gaps is already a non-trivial task for the exemplary used areas energy efficiency, performance, and information security. Nevertheless, the integrated metric management and management cockpit must be capable of integrating or at least be extendable to an arbitrary set of measurements and metrics. Many more IT infrastructure capabilities can be harvested for various types of metrics, such as cost, reliability, re-usability, and degree of standardization. It should, however, be kept in mind, that the complexity of the concepts discussed in this article increase with the applied breadth of the capability spectrum.

Allowing multiple-perspective scenarios – Mostly, top-level management consists of several experts from different areas. Hence, the management cockpit should support "multiple-perspective scenarios", i.e., "many different narratives about the same events, with the intention being to explore how the different perspectives might be coordinated or might reach some accommodation" [33]. For instance, there is a user type *client category A*, a business unit *highly critical storage services* or a topic *Urgent Computing*. Each view defines its specific obliged values and objectives, which are in turn considered while planning strategic actions or behavioral guidelines concerning the IT infrastructure. In summary, multiple views shall be consolidated into one holistic view [33] and strategic goal conflicts between perspectives are exposed.

Integrating measurement and metrics data – Supporting the severally motivated holistic view and root cause analysis requires the integration, i.e., the "selection, embedding, and handling of the underlying data sources" [34] and the use of as many data sources as possible. This in turn calls for the consolidation of several data structures and the identification of a valid data context [22]. Enabling the use of the management cockpit from the first day and needs to avoid the "cold-start–problem" [35], and existing and actual data, measurements, and metrics have to be embedded [22].

III. Scenario Leibniz Supercomputing Centre

The complexity of working with a large number of measurements and metrics is easier to grasp when a real-world example is used. Hence, this section illuminates a concrete scenario in order to make this variety more tangible and to provide some examples for the high level of abstraction of the above discussed challenges and issues.

Smaller IT infrastructures, such as small number of servers operated by a university computer science chair or a very small enterprise do not exhibit the same IT management decision problems as large IT infrastructures. Similarly, when research is focused on only a single metric category, the issues resulting from interdependencies we have to face in real-world scenarios are often neglected. Having said that, we use the LRZ as an example because we know the set of problems there in-depth based on several projects and the IT service operations we are involved in.

LRZ is located in southern Germany and has a twofold mission. On the one hand, it is the common IT service provider of all higher education institutions in the greater Munich area. It offers several dozen IT services for more than 130.000 students, faculty, and staff; for this purpose, it operates a four-digit number of server machines and a communication network infrastructure consisting of more than a dozen *Internet Protocol* (IP) routers and about 1.500 network switches, making it well-comparable with larger enterprises. On the other hand, LRZ is one of Germany's largest scientific HPC sites. Besides a large Linux cluster with about 10.000 CPU cores, it operates a supercomputer named SuperMUC, which entered the Top 500 HPC list at place 4 in June 2012 and was Europe's fastest supercomputer. LRZ had to construct a completely new building for SuperMUC, which uses hot liquid cooling, and has received a national award for the energy efficiency of its infrastructure in 2012.

SuperMUC's architecture and relevant characteristics are subsequently outlined in Section III-A. Section III-B then focuses on the selected metrics in general and concrete examples for the SuperMUC in particular. As a core issue, let us assume the following questions that LRZ's management wants a profound answer for: "How can future HPC systems at LRZ be made even more energy-efficient without impacting their performance and scrutinizing their security? Can some of these measure even be already applied to SuperMUC without exceeding a given yearly maintenance budget?"

A. LRZ's and SuperMUC's IT infrastructure

In a *High Performance Computing* (HPC) system there typically are dedicated worker/compute nodes, storage components, a head node and an interconnecting high-bandwidth network [36]. These components interact and exchange information to expose HPC capabilities. Additionally, there are several metrics and measurement approaches for multiple areas, such as availability, performance, or quality of service. Additionally, there are manifold approaches within a single area, e.g., ranging from low-level considerations like investigating the power consumption of an Intel PXA255 processor [37], to high-level considerations, like investigating the power consumption behavior in an actual data centre [38].

Figure 5 depicts a schematic view of SuperMUC's architecture: SuperMUC's compute elements are built of 18 identical *IBM System x iDataPlex* thin node islands. An island comprises 512 nodes, each employing two *Sandy Bridge-EP Intel Xeon E5-2680 8C* processors having 8 cores each, resulting in 147.456 cores. There is an additional fat node island with 40 cores per node and 6.4 GB RAM per core, providing additional 8.200 cores.

Storage elements are split in three areas accordant to their intention. The temporary disk storage for compute job execution is run in IBM's *General Parallel File System* (GPFS), a high-performance clustered file system. The permanent storage, e.g., for `home` directories, are located on a *Network Attached Storage* (NAS) based disk storage.

As depicted in Figure 5, also the network is split in different areas and employs different technologies. Islands and their nodes as well as the temporary disk storage are connected via an *Infiniband interconnect*. The Infiniband interconnect is operated at a *Fourteen Data Rate* (FDR)-10. SuperMUC's size requires the employment of several switches, in particular 20 big island switches and several smaller switches within an island. The archive and backup system is connected via a slower 10 Gb Ethernet.

Additionally, the campus on which LRZ building resides is one of the major backbone sites of the networking infrastructure referred to as the Munich Scientific Network, and provides a 23.5 GBit/s uplink to German's national research and education network, X-WiN. Because LRZ also operates several thousand Linux and Windows servers, NAS filers, and a tape backup and archival infrastructure, several hundred edge and access network switches are used in the LRZ building. In total, more than 450 kilometres of copper and glass fiber cables are used in the single data centre building to provide the required connectivity with a carefully crafted redundancy for high availability that covers technical failures as well as major incidents such as room-local fires.

B. Exemplary metric categories for use at LRZ and SuperMUC

Out of the variety of potential metric categories, we further detail *energy efficiency* (Section III-B1), *performance* (Section III-B2) and *information security* (Section III-B3). The former two are considered the be among the most important ones accordant to the PRACE scientific case [39]. The latter one is an essential area of responsibility for both system administrators and management. Unfortunately – and directly related to how management decisions are made due to the identified gaps – security is not yet in the core focus of most HPC installations. However, as security often requires a trade-off with other goals, such as performance, intertwining all three metrics categories can be expected to become more important in the future.

For each metric category, a general discussion is succeeded by a concrete consideration in the context of the above outlined scenario.

1) Energy efficiency: EE is a severe problem given the background of expected consumption levels of hundreds of megawatts in the future [40], [41] and steadily increasing electricity prices. For many data centres and other IT service providers, raising energy consumption costs are the primary motivation for an in-depth examination of EE technology. EE obviously is important to consider before hardware investments are made; for example, buying new servers with CPUs supporting frequency scaling helps to level energy costs with the current workload throughout the lifetime of the server machines. Buying cheaper servers and replacing the CPUs afterwards typically would lead to a much higher total cost of ownership. However, EE capabilities need to be constantly monitored and several EE parameters need to be dynamically re-configured. For example, air-conditioning for the servers typically needs to be adjusted to environmental characteristics such as the current outdoor temperature.

EE obviously is of utmost importance also for LRZ: SuperMUC consumes about 3 MegaWatts of power when it is under full load, leading to multi-million Euros power cost per year for this single system. SuperMUC's EE therefore clearly dominates LRZ's power bill, but several thousands of other server machines and network components must not be neglected either. For example, state-of-the-art network switches by well-known international vendors differ by factor 2 regarding their waste heat production when power-over-ethernet-enabled models are concerned. This does not only influence the power consumption of the IT equipment itself, but also has consequences for the climate / re-cooling infrastructure because cooling airflows need to be increased.

We now give some examples for the gaps identified in Section II related to EE. The same gaps exist for the metrics categories discussed below but are omitted there for brevity.

Example for Gap 1 – In order to answer the management question how SuperMUC's EE could be further improved, we first have to decide, which components have the poorest energy efficiency in SuperMUC at the moment, as their potential for further improvement during the next system extension is the highest. Besides a few generally applicable metrics, most metrics can be applied only in one area, for

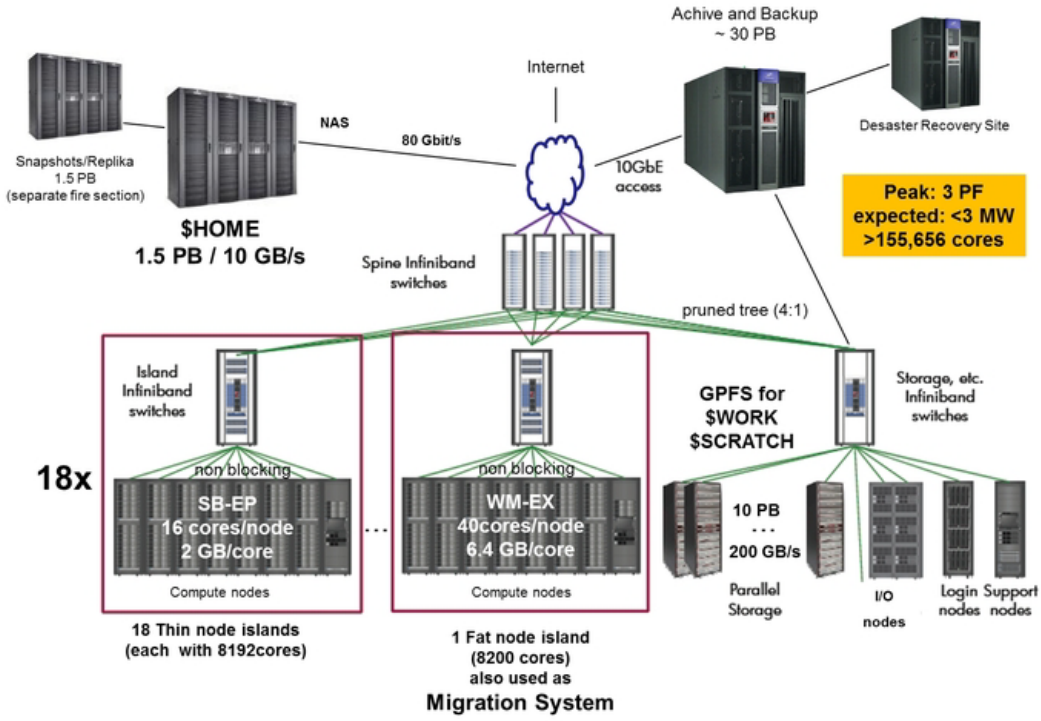


Figure 5. Schematic view of SuperMUC at LRZ (see `www.lrz.de/services/compute/supermuc/systemdescription/`, last accessed at 29th of May 2014)

instance GB/Watt. Hence, there are several different metrics that have to be considered for SuperMUC as a complete system.

Example for Gap 2 – In the SuperMUC scenario, changing the CPU type to achieve a higher energy efficiency would have (strong) side effects on other components of SuperMUC. For instance, using CPUs with a smaller L2 cache size might improve the CPU energy efficiency, but at the same time, SuperMUC's system interconnect between the CPUs and the non node-local memory will have higher workloads and therefore, its energy efficiency is decreased. This may lead to a decreased overall energy efficiency.

Example for Gap 3 – Bavaria in southern Germany, where SuperMUC is operated, is a relatively warm region compared to, for example, Iceland. The location therefore influences the demand for cooling and climate infrastructure. SuperMUC works energy-efficient because it supports hot-liquid cooling, i.e., free cooling from LRZ's rooftops can be used throughout the year without the demand for energy-demanding cooling machines. However, if SuperMUC was operated in Iceland, cooling it with fresh cold air from outside may be even more energy-efficient.

Example for Gap 4 – LRZ performs many different measurements regarding energy consumption and efficiency of both SuperMUC and the other IT infrastructure. However, decisions about further improvements have to be made manually by individuals from different departments as there is no common understanding of LRZ-wide EE yet and there is a complete lack of tool support when it comes to anything more than the simple visualization of raw measurement data.

It should also be mentioned that SuperMUC is one area in which EE is actively researched. EE for other parts of LRZ's IT infrastructure is hard to improve. For example, research papers have often suggested to turn network links between routers and switches off, e.g., outside office hours, to lower the energy consumption of the networking infrastructure. This does not work in practice for the simple fact that during the night often more traffic is generated than during the day, for example, due to automated backups and other bulk data transfers. Also, in an academic environment, it is impossible to completely shut down the networking infrastructure for whole building, e.g., over the weekend or holidays, because some researchers might still be working and depend on a working infrastructure.

2) Performance: Higher PE for the IT services that support business processes is the primary driver for investment in new and additional hardware and software. However, benchmarking and scaling PE often is tricky. For example, a computationally intensive application may benefit from faster CPUs and additional RAM, whereas a database server may best be sped up by replacing HDDs with SDDs; also, increasing the LAN bandwidth from 1 Gbit/s to 10 Gbit/s does not imply that employees have ten time faster access to local file servers or Internet content.

At LRZ and in the Munich Scientific Network, performance is critical for user experience: Students and faculty expect, for example, access to LRZ's central file servers from labs and offices to be as fast as locally operated storage solutions. However, the central file servers need to accommodate many more users who are active in parallel and the communication network needs to transport all the data across the backbone and access networks with the same quality-of-service parameters as a LAN, for example, regarding bandwidth and IP packet delay.

3) Information security: The primary goal of *Information Security* (IS) is to ensure the confidentiality, integrity, and availability of services and data. For example, the highly innovative research carried out on SuperMUC must not leak to unauthorized third parties and an attacker must be prevented from manipulating code, input as well as output data of HPC job submissions. Consequently, IS is an essential area of responsibility for both system administrators and management because of two reasons. First, it is a key component for compliance, i.e., the fulfillment of laws, like Germany's strict data protection and privacy laws, industry-sector-specific regulations, contracts with business partners, and intra-organizational policies. Second, many university departments and chairs store project data in cooperation with industry partners, resulting in high confidentiality, integrity, and availability demands. Measuring IS and providing adequate evidence even to third parties becomes more and more important. Despite this important role, IS often is perceived as a necessary evil, especially from the management perspective, because it costs money but, unlike other investment, cannot generate any direct return on invest (ROI) due to its nature. The aspects that are in the focus of IS are inherently hard to quantify because there are no standardized units of measurement yet. While many security experts have a reliable gut feeling about the security state of a system they analyze and there are many standardized IS controls, e.g., those specified in ISO/IEC 27001 [42], objectively assessing arbitrary security properties and making them comparable across organizations' boundaries is still impractical.

Concerning IS metrics, LRZ uses more than 50 measurement procedures and metrics to monitor the overall security level of its infrastructure. For example, regarding system management the delay between the vendor publication of software security patches/updates and their application to at least 80 percent of all relevant LRZ servers is measured. Each server's network traffic is monitored for suspicious IP packets and changes to its communication characteristics, which may indicate a compromised machine. *Virtual Private Network* (VPN) and *Wireless* (WiFi) users are monitored for Internet SMTP connections, and if certain thresholds are exceeded, these client machines are flagged as probably malware-infected, Spam-sending devices and are put into quarantine.

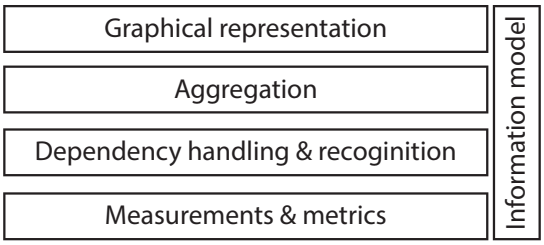


Figure 6. The presented design concept applies the layer pattern to achieve a separation of concern

4) Difficulties with metrics handling in practice: The results of each of the metrics categories discussed above are currently assembled and evaluated by different personnel: An EE project team works closely together with facility management, PE is handled by system administrators together with a central server operations group, and an inter-department security team handles IS and risk management.

For comprehensible pragmatic reasons, for example the security team does not always pay attention to EE issues, and software and hardware changes that are supposed to enhance PE do not always keep IS in mind. For design decisions on all layers of the organizational hierarchy, a holistic metrics management approach therefore would lead to more profound and even better results.

IV. A DESIGN CONCEPT FOR AN INTEGRATED MANAGEMENT PLATFORM OF IT INFRASTRUCTURE METRICS

This section presents a design concept for an integrated management platform of IT infrastructure metrics. The design concept aims at implementing the outlined design criteria (cf. Section II) in general and at closing the identified and details four gaps in particular.

Figure 6 depicts the design concept's four layers. As the layer titles imply, each layer has a dedicated topic. The undermost layer covers existing measurements and metrics and their integration. The next layer addresses the handling, use, and recognition of dependencies between the elements of the undermost layer. Based on that the third layer defines and implements aggregation logic whose results are (graphically) represented by the uppermost layer. Orthogonal to the four layers, the information model describes all relevant entities and their attributes.

The layer structure depicted in Figure 6 also guides the section's structure: We discuss the measurements and metrics layer in Section IV-A, followed by a description of a middle layer that handles dependencies detection and management in Section IV-B. The logic for aggregating and combining multiple measurements and metrics is then described in Section IV-C. Finally, Sections IV-D and IV-E deal with various aspects of the graphical representation of the results, which constitute the top layer of the architecture, and presents our information model in depth.

A. Layer 1 – Measurements and metrics

The undermost layer 1 comprises the measurements and metrics that provide the data, which is processed by all upper layers. Hence, errors and discrepancies in the initial "raw" data might raise to higher power, depending on the applied aggregation and processing algorithms. Consequently, a data quality as high as possible should be achieved. Layer 1 implements this role by considering *measurement quality* and *metric quality*.

Measurement quality – The most important aspect about measurements is the accuracy of the compiled values. This accuracy can be penalized by technical and social factors. Technical factors cover the measurement setup, e.g., the accuracy of the applied instruments or errors while storing the values. These problems can be addressed by a thoroughly planned measurement setup. Additionally, for each instrument or measurement procedure, the accuracy should be provided. For instance, most power consumption instruments like multimeters provide an accuracy about 2%. More challenging is the avoidance of social factors, especially avoiding the *feedback mechanism*: since measurement and metric results might affect the measuring entity in a negative way, the entity might (subconsciously) influence the measurement [19]. In other words, a measurement should be stable, i.e., compile the same results even if different entities or persons conduct the measurement [20].

Metric quality – The definition and enforcement of metric quality is a broad field and there are manifold approaches. For layer 1, we select the subsequently itemized definitions:

SMART A metric is considered to be good if it is *specific*, *measurable*, *attainable*, *repeatable*, and *time-Dependent* (SMART) [13], [43], [44, 6–10]. This set of quality criteria is in close correlation to the criteria defined by Bianzino et al. [24], i.e., *simple* (enough to understand), *accurate* (enough to withstand scrutiny), *usable* and *relevant* (enough to be an effective agent of change).

Stability A metric's semantic must remain the same during the entire life cycle and/or use time [45]. Additionally, the semantics is independent from the concrete description language, like QML, Windows Management Instrumentation or vendor specific SLA management solutions.

Empirical validation A metric must be defined in a way that it can be validated efficiently and empirically [45]. General speaking, a metric is of no use if it is not possible to validate its implementation or application [45]. For instance, defining "response time" as metric also requires a possibility to check the response time empirically.

B. Layer 2 – Dependency handling and recognition

The layer 2 covers the topic of handling and recognizing dependencies between two or multiple metrics. Dependencies are split in *reciprocity*–dependencies and *aggregation*–dependencies: former describes correlations between metrics, for instance, improving CPU energy efficiency potentially decreases interconnect energy efficiency. The latter addresses the aggregation of metrics to form new statements. Both dependency types comprise a *definition* phase and a *detection* phase.

The **definition** phase of a dependency covers the semantics of a dependency and influences the detection and modeling. Basically, there are different types of dependencies according to the considered attribute categories and hardware types. Additionally, there are direct and indirect dependencies. The direct dependencies affect the metric itself, for instance, the current load of a hardware component influences the measurements and metrics about time to completion or current power consumption. Indirect dependencies are between the hardware components and hence, affect the applied measurements and metrics only indirectly.

According to the definition, there are different ways of dependency **detection**, i.e., analytically or empirically. An analytic detection mechanism processes (structural) information about the considered IT infrastructure, like a *Configuration Management Database* (CMDB) [46], and derives dependency insights. An empirical detection mechanism collects data at different points in time at different sensor points in the IT infrastructure, e.g., before and after a reconfiguration. Another example is the mechanism of failure injection as applied by Bagchi et al. for uncovering resource dependencies in a dynamic distributed e-commerce environment [47].

C. Layer 3 – Aggregation logic

Layer 3 uses the information provided by the two layers below, i.e., the (revised) raw measurement and metrics data provided by layer 1 and the (incorporated) dependency information generated by layer 2. The definition and implementation of aggregation rules and concepts are encapsulated in layer 3 and split in three aspects, i.e., the aggregation *direction*, the applied aggregation *rules* and the aggregation rule *declaration*.

There are three possible aggregation **directions**, i.e., bottom-up, hypothesis generation on middle, and top-down. *Bottom-up* aggregation uses existing data from low abstraction levels and aggregate them iteratively until the pursued granularity level is reached. The most difficult task while doing a bottom-up generation is the "correct" selection of attributes/values at the lowest level. *Hypothesis generation* formulates hypotheses on an intermediate level and tries to prove or disprove those hypotheses by applying data from low abstraction levels. Those (dis)proved hypotheses are afterwards used to generate statements for a higher-level consideration. *Top-down* starts at certain points in the upper levels and tries to create the data tree beginning at the root by recursively finding suitable metrics on the next lower

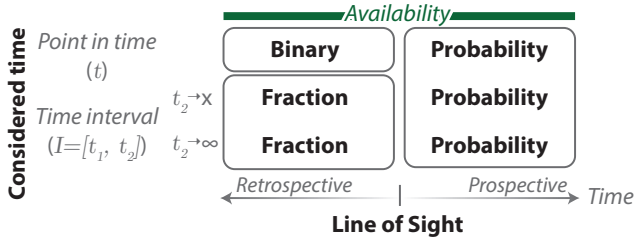


Figure 7. Metrics about component availability result in *binary*, *fraction*, and *probability* data types (taken from [49])

level. The aggregation logic layer applies the *Bottom-up* direction, since we aim at integrating existing measurements and metrics and hence, have to start at this (immutable) point.

Having set the aggregation direction, the next step is the definition of aggregation **rules** that describe metric aggregation (in a mathematical way). As explained in Section II-A, a metric is a mapping of two measurement values on a distance, which is the *image* of that mapping. To cover the variety of existing measurements and metrics as well as the different metric categories (cf. Section I), aggregation focuses solely on that *image*. This narrows the problem domain of aggregation to a set of very basic data types and semantics. e.g., *binary*, *fraction*, and *probability*.

Figure 7 explains these data types exemplary for the metric category *availability* that describes a component being in the ***up state***, i.e., it delivers a *correct service* or a *system function* as it is described by the functional specification [48]. The line of sight describes whether the considered availability values are in the past or in the future. The considered time describes whether a component's availability is considered for a discrete point in time or a time period. Bringing these dimensions together, only *binary*, *fraction*, and *probability* values are possible. For instance, a component was (*retrospective*) available at t (*point in time*) yes or no (*binary*). According to the data type and the implicitly contained semantics the aggregation rules can be formulated. An extended explanation and further details are provided in [49].

Finally, the selected aggregation rules must be **declared**. Basically, an arbitrary language for rule declaration can be applied, as long as it meets the following requirements that were developed in previous work in our research group by Sailer (see [50]):

1) Expressiveness: A declarative programming language shall be used; this enhances the legibility of the metrics specification, e.g., compared to XML-based specifications, and is sufficiently decoupled from specific implementations.
2) Access to data: Any derived measure or metric is a synthesis of data retrieved from various sources. The used language must make this data available, e.g., as read-only variables.
3) Aggregation operations: Many metrics can be expressed using basic arithmetic operations. However, more complex metrics require statistical function libraries and first-order logic. Ideally, language users can define their own functions.
4) Triggers: To ensure that accessed data is up-to-date and eventually trigger other preparations of the environment before aggregation operations are performed, the interaction capabilities of the used language must include ways to start and control measurements and other processes.

We propose to use the *Service Information Specification Language* (SISL) that has been introduced by Danciu et al. [51]. It has explicitly been designed independent of specific IT systems, metrics categories, or implementation technologies. It is strictly typed and provides support for integers, floating point numbers, strings, and temporal as well as Boolean expressions.

D. Layer 4 – Graphical representation

According to one of the fundamental design principles of software development, i.e., separating (graphical) representation and logic, layer 4 encapsulates the graphical representation of the results compiled by the other three layers. The implementation of the layer 4 highly depends on the individual objectives of top-level management, the required insights for decision making and the characteristics of the information provided by layer 3.

Figure 8 depicts an exemplary graphical representation of information about the energy efficiency of LRZ's SuperMUC (cf. Section III). The depicted management cockpit comprises three areas, i.e., a *tree-view* for aggregated values (labeled "1"), a *delta-view* of current and oblige values (labeled "2"), and a *activity recommendation* (labeled "3").

The **tree-view** provides information about the sources of a particular value. This information is required by the urgent need of provenance and to facilitate root cause analysis: starting at the top level, any aggregated metrics value can be broken down into smaller pieces and it can be explained how this high-level current value materializes. Figure 8 illustrates that the overall energy efficiency value of SuperMUC is aggregated from *Interconnect*, *Storage*, and *CPU* values. The CPU value, in turn, is composed of *Operations/kWh* and *MFlops/Watt* values.

The **delta-view** compares the current (aggregated) value and its assigned obliged value. The delta's color is determined by the predefined allowed threshold for a particular metric or rather its interpretation: if the threshold is exceeded, the delta is colored red. For each perspective (cf. Section II-C) a different set oblige values can be defined.

The **activity recommendations** depend on the delta of oblige and current values, a optionally predefined escalation mechanism or a criticality level. A possible recommendation could be to decrease the CPU clock time. Obviously, the

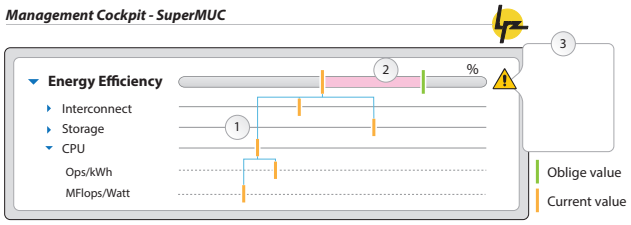


Figure 8. Exemplary graphical representation of high-level, management-relevant information about SuperMUC's energy efficiency.

challenges described in the activity gap (cf. Section II-B) have to be thoroughly incorporated.

E. Information model

As further detailed in the following Section V, the strictly separated development of metrics in different research areas and the absence of inter-domain metrics aggregation concepts results in a lack of existing common information models, standards, and best practices for the internal representation of measures, aggregation rules, reports, and other ways to refine and present metrics to decision makers and other users. Hence, a uniform information (and data) model for the various metrics categories is imperative for scalability.

This uniform information model is provided by the orthogonal layer *Information model* that covers all important elements and attributes of the above detailed four layers. Depending on the implementation technology, a platform specific data model can easily be derived from the provided information model. Figure 9 depicts the information model as *Unified Modeling Language* (UML) class diagram. Guided by the overall article's aim of providing a design concept but concrete implementations (cf. Section I), the class diagram representation is chosen to ease extending the provided information model by individual sub classes. For the same reason, some classes come with less attributes, like a `Role`, since the set of reasonable attributes depends on the individual situation.

The classes are organized in three packages, i.e., `general`, `datasources`, and `representation`, which are subsequently further detailed.

`general` package – The package contains all classes that are relevant for storing meta information about the core classes. The class `UniqueID` provides a globally unique identifier to ensure an ambiguous identification of each particular element. Further description of entities is achieved by assigning an arbitrary set of `Keyword` and `Category` objects. The classes `Timestamp`, `Formula` and `Frequency` provide (complex) data types and describe their representation. A `Formula`, for instance, has a natural language label and description and a specific way of declaration it (cf. Section IV-C). A `Role` is used to describe a responsibility, e.g., a system administrator or laboratory employee.

`datasource` package – This is the main package, since it contains all elements for gaining, gathering and compiling information for the management cockpit. The contained classes are further structured in `measurement`, `metric` and `interpretation`, guided by the concepts presented in Figure 3 (page 4).

The `Datasource` class collects all attributes that are in common for a measurement, a metric, and an interpretation. Plain management aspects are described by a `label` and the `objective` in natural language and an arbitrary number of `Keyword` and `Category` objects to facilitate searches for suitable measurements and metrics, e.g., if new reports have to be designed. Additionally, a `version` is stored to allow the application of different versions at the same time and to support provenance. The version information is enhanced by a `DatasourceStatus` enumeration, comprising items such as `Active` or `Retired`.

Responsibilities are described by an arbitrary set of `Role` objects and the accordant association class. In this `Responsibility` class, the responsibility can be detailed, e.g., performing a measurement, reviewing a metric or being the authoritative source for a measurement or metric, like a SLA or policy.

To enable a reuse of scales, there is a dedicated class `Scale`. Besides a natural language `label` and `description`, the `Scale` class most importantly describes a `unit`. Exemplary values are *Watt* (for a measurement), *Ops/kWh* (for a metric) or *school grade* (for an interpretation). To further detail the scale, a `ScaleType` enumeration entry can be assigned.

Besides the above detailed general elements, the `datasource` package contains additional packages for each element depicted in Figure 3, i.e., a `measurement` package, a `metric` package, and an `interpretation` package.

`measurement` package – All entities relevant for measurement and storing measurement values are collected in this package. The `Measurement` class describes the activity of measuring or in other words, the mapping of facts to a symbol set (cf. Figure 3). Consequently, the class contains information about the measurement activity, i.e., what (`measuredComponent`) was when (`timestamp`) how (`isAutomated`) measured. The applied procedure is further detailed by the `MeasurementProcedure` class. The frequency of reviewing the measurement activity, e.g., analyzing the applied procedure or re-checking for necessity and suitability, is described in the `reviewFrequency`.

The compiled measurement values or the image of the mapping (cf. Figure 3) are stored in `MeasurementValue` objects. The class' `value` attribute is dependent on the assigned scale. The differentiation between simple and derived measurement values introduced in Section II-A is represented by the dedicated class `DerivedMeasurementValue` that is assigned to an

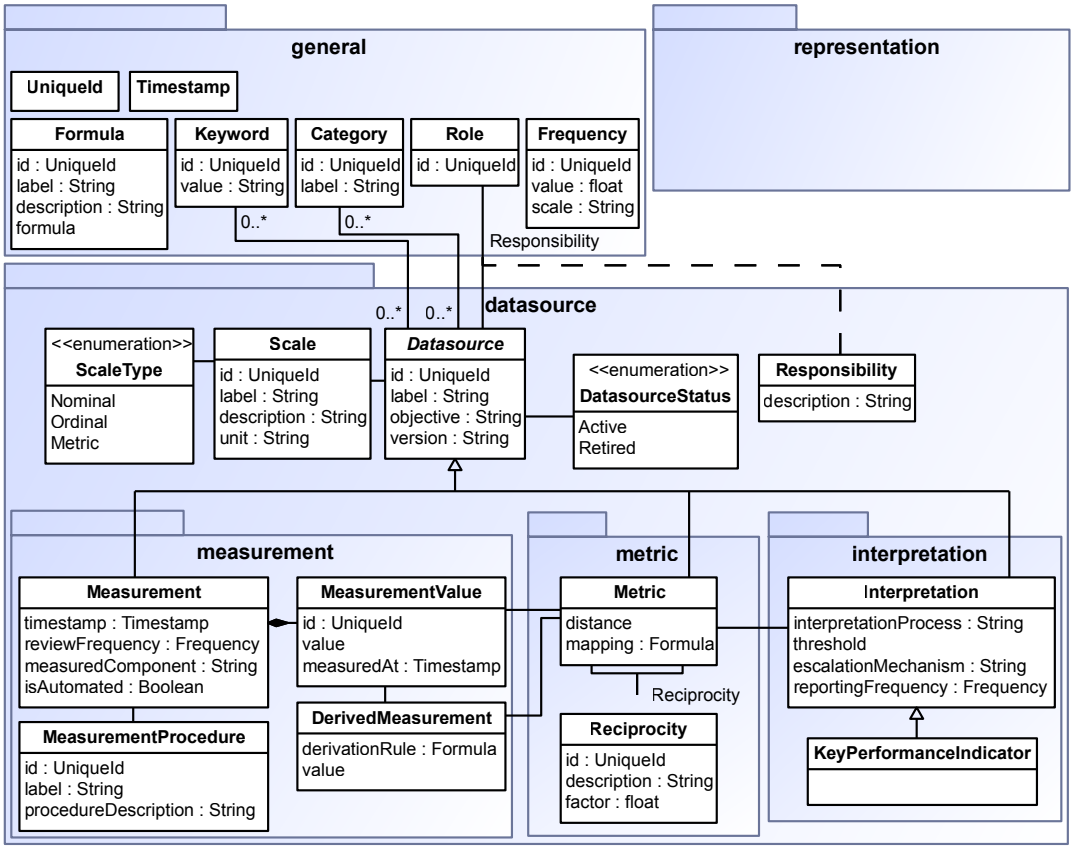


Figure 9. The design concept's information model.

arbitrary set of `MeasurementValues` and describes a `derivationRule`.

`metric` package – The package contains elements to describe the mapping of two measurement values on a distance and related aspects (cf. Figure 3). The `Metric` class stores a `mapping` of the assigned measurement values to the compiled `distance`. Dependencies between metrics are modeled by the `Reciprocity`, which contains a natural language `description` and a `factor` defining how two metrics are related. A `factor` value of 2, for instance, would describe a positive correlation by the factor 2.

`interpretation` package – The elements to describe interpretations for a metric (cf. Section II-A) are summarized in the `interpretation` package. The `Interpretation` describes a `interpretationProcess`, i.e., how the interpretation is conducted. To support automated interpretation, it additionally contains a `threshold` and an `escalationMechanism`, which is triggered on threshold violations. The `escalationMechanism` is not contained in the `Datasource` class but only assigned to the `Interpretation`, since we are interested in top-level management decision support and not in low-level monitoring, which mostly already applies mature escalation and trigger mechanisms. A `reportingFrequency` can be specified if the interpretation is not only used interactively via the management cockpit, but also included in periodic reports.

The `Interpretation` is refined by a `KeyPerformanceIndicator` class, which is typically required for organization-internal audits or are included in reports. This class could be used as interface to ITSM processes, which can be based, for example, on the *IT Infrastructure Library* (ITIL v3 [52]) or the ISO/IEC 20000-1 standard [53]: IT service providers have contracts with their customers, which are referred to as SLAs, and each SLA typically specifies thresholds and nominal values for KPIs, e.g., the service's monthly availability must at least be 99.9 percent. SLA violations by the service provider can then, for example, lead to penalties.

`representation` package – The above mentioned software design principle of separating logic and representation guided not only the layered architecture of the presented design concept, but also the package structure depicted in Figure 9. As described in layer 4 (cf. Section IV-D) the graphical representation heavily depends on several individual parameters. Consequently, the `representation` package is only a placeholder to emphasize the urgent need to separate information representation from the other

elements. Hence, the package contains all components that are necessary to achieve the graphical representation. For the example provided in Figure 8, there could be color assignments or labels. Further examples are suggestions for *graphic representation* could be stored, e.g., whether the measure's history should be visualized as line chart, *interpretation rules*, i.e., guidelines for understanding what the measure expresses, *decision rules*, i.e., guidelines for acting appropriately based on changes to the measure or a lack thereof, and *instructions*, i.e., suggestions for actions that should be taken depending on which decisions have been made.

V. Metrics management in related work

In the following subsections, we discuss related work and the current state-of-the-art. Discussion covers a quantitative additive metric, i.e., *energy efficiency*, and a qualitative metric, i.e., *information security*. Additionally, approaches for structuring metrics and aggregating their values are considered.

A. Energy efficiency metrics

There are a lot of metrics dealing with different energy efficiency aspects, e.g., measuring the power consumption of computing servers and clusters [54], [55], or the power consumption in optical IP networks [56]. Further examples are TEEER [57], EPI [58], ECR, and ECRW [59], [60], [61]. All of them are defined by providing calculation and interpretation rules, partially in a very comprehensive way, but nevertheless they all focus on technical aspects of a single entity on a very low level. Hence, they do not facilitate a holistic view on the energy efficiency situation of SuperMUC, which, being a large HPC system, aggregates many different hardware components in a complex architecture. There is some work postulating the extension of existing metrics, e.g., Banerjee et al. [58] propose to define energy consumption not in an absolute way, but proportional to the workload.

B. Security metrics

The work on security metrics, which earned its spot on the INFOSEC research council's *hard problems list* in 2005 [62], is motivated by the difficulty of answering seemingly simple questions, such as: Which of the n possible system configuration variations is the most secure? Is it advisable to invest in security measure x? Is organization i's security level higher than organization j's? As there is no physical unit of measurement for security and we still lack established standards and best practices, the currently only commonly accepted conclusion is that each single security measurement or security metrics has limited expressiveness [63].

The most wide-spread approach to use IS metrics is hypothesis-based [64], [65]: Hypotheses are derived from known risks or attacker models, and metrics are defined to corroborate or vitiate them. Many dozens of security metrics have been suggested, e.g., by [64] and [66]. A common denominator of many IS metrics is that the involved units are currencies or durations; this facilitates a direct mapping to operational costs or amount of work, which often is preferred by top management according to [64]. NIST has published its *directions in security metrics research* in 2009 [67] and defined milestones for the improvement of security measurability.

IS metrics are closely related to investment models, such as Gordon's and Loeb's [68], which allows for ex-ante security measure cost-benefit calculations. It is motivated by the problem that classic economic models, which typically involve some sort of return on invest (ROI), are unsuitable for IS investments because security measures usually cannot directly increase the volume of sales or profit; instead, they only impede or reduce the effects of security events causing damage.

Security is, especially due to the heterogeneity of existing metrics, probably not the youngest research area for measurements and metrics, but the most complex one of the three we investigate. This assumption is supported by the almost complete lack of IS metrics management software so far. Several researchers and commercial vendors have attempted to adapt existing security management software, such as security information & event management (SIEM) systems, for security metrics and security report creation purposes.

However, as already discussed by Jaquith in [64], such systems have a strong focus on real-time monitoring, whereas security metrics are intended to facilitate long-term processes and decision making. They also focus on single measurements or the extraction of information from log entries, whereas several security measurements typically need to be aggregated and combined to form a security metric. IS metrics are not necessarily purely technical either, for example when the percentage of employees who already received the quarterly security instructions is calculated, whereas SIEM systems and similar solutions focus on technical measurements only. A comprehensive tool set for integrated metrics management therefore would highly benefit IS.

C. Aggregation of measurements and metrics

According to the basic ideas presented for layer 3 in the previous section, the most important aspect for aggregating metrics is the metric image. Consequently, some structuring and taxonomy approaches are considered, since their results could be used to gain insights in a particular metric's image.

There is literature and ongoing research in several topics about metrics taxonomy [69], [70], classification [24], and comparison [13]. These approaches structure and compare

the aforementioned metrics in a single specific metric category or granularity level, like equipment-level metrics, and confine themselves to comparison. Therefore, they do not allow a holistic view either.

There are some comprehensive papers that compare and classify existing metrics, like [24], which proposes four hierarchical levels "equipment", "facility", "corporate", and "country". Wang et al. [70] recommends server level benchmarks and data centre benchmarks. Also these approaches focus on a specific class of metrics, like Bianzino et al. [24] do on equipment-level, and confine themselves to comparing single metrics. Therefore, they do not allow a holistic view neither.

VI. Summary and outlook

In this article, we first outlined the importance of accurate information about complex IT infrastructures to support profound decision making. Quantitatively expressing the key properties of IT systems, and aggregating raw measurements to meaningful higher-level information is a non-trivial task when data from various domains, such as energy efficiency, performance, and security have to be integrated. After introducing the basic terminology, we conducted a gap analysis with the following four key results: First, there is an information gap, i.e., information provided by most current metrics is not suitable for decision making on higher abstraction levels. Second, the interdependencies between the metrics are not sufficiently considered. Third, there currently is no support for the derivation of concrete improvement recommendations based on the metrics values. And fourth, existing approaches do not allow for customization in order to incorporate organization-specific requirements. To exemplify these four gaps, we outlined the Leibniz Supercomputing Centre scenario and described how measurements and metrics are applied in practice currently, along with the drawbacks that result from a non-integrated approach.

Motivated by these deficiencies, we then presented our design concept for an integrated management platform for IT infrastructure metrics. The overall design is based on a four-layered architecture, which we described in detail: On the lowest layer, 1, measurements and metrics are handled. Layer 2 then deals with the recognition and handling of dependencies between two or more metrics. Layer 3 uses the information provided by the lower layers to conduct the required aggregation logic, and Layer 4 covers important aspects of the graphical representation of the *management cockpit*. Our information model describes all relevant entities and their attributes as they are used across those four layers. Finally, we investigated the state of the art of metrics management for energy efficiency and security metrics as well as for metrics aggregation.

Our ongoing work will focus on the following open issues next:

Target values and comparison In order to provide "Warnings and activity recommendations", target values and interpretation rules for a delta between those target values and current values are mandatory. We have to investigate how to define or rather find those target values. This step is very critical, because having wrong target values would lead to optimizing the infrastructure towards wrong values. Additionally, we have to analyze how to interpret a delta between the current value and the target value for any given metric. This interpretation has three dimensions: overall meaning, timing aspects (e.g., "delta implies the necessity to act immediately", "delta is just for the annual, paper-based report"), and impact (e.g., "the severity of the delta is very high", "solving the delta is very costly").

Validation We need to perform a practical evaluation of our approach, i.e., the metrics in use today in our scenario need to be analyzed for their interdependencies and implemented based on our information model. This will serve as a basis for a prototype implementation of the management cockpit, which will be used to demonstrate the benefits of our solution in a real-world scenario. We will then also include metrics from additional categories and analyze the scalability of our integrated management approach.

Prospective view The management cockpit presented in this article uses measurement data, i.e., data about the (recent) past. Enabling comprehensive *what-if* analysis about planned modifications would require the application of models and their compiled predictions, i.e., data about the future. Consequently, we currently investigate the integration of existing models for manifold IT infrastructure types and architectures.

Acknowledgment

The authors thank the members of the Munich Network Management (MNM) Team (www.mnm-team.org) for valuable comments on previous versions of this article. The MNM-Team, directed by Prof. Dr. Dieter Kranzlmüller and Prof. Dr. Heinz-Gerd Hegering, is a group of researchers from the Ludwig-Maximilians-University Munich, the Technische Universität München, the Munich University of the German Federal Armed Forces, and the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities.

References

[1] C. Straube, W. Hommel, and D. Kranzlmüller, "A Platform for the Integrated Management of IT Infrastructure Metrics (Best Paper Award)," in *Proceedings of the 2nd International Conference on Advanced Communications and Computation (INFOCOMP'12)*, 2012, pp. 125–129.

[2] L. Xue, G. Ray, and B. Gu, "Environmental Uncertainty and IT Infrastructure Governance: A Curvilinear Relationship," *Information Systems Research (INFORMS)*, vol. 22, no. 2, pp. 389–399, 2011.

[3] S. H. Chung, T. A. Byrd, B. R. Lewis, and F. N. Ford, "An Empirical Study of the Relationships between IT Infrastructure Flexibility, Mass Customization, and Business Performance," *ACM Special Interest Group on Management Information Systems (SIGMIS)*, vol. 36, no. 3, pp. 26–44, 2005.

[4] P. Weill, "The Role and Value of Information Technology Infrastructure: Some Empirical Observations," Massachusetts Institute of Technology (MIT), Sloan School of Management, Tech. Rep. Sloan WP No. 3433-92, 1992.

[5] D. Briody, "Big Data Harnessing a Game-Changing Asset – A Report from the Economist Intelligence Unit Sponsored by SAS," Economist Intelligence Unit, Tech. Rep., 2011.

[6] C. Rupp, *Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis*. München: Hanser, 2009, vol. 5.

[7] S. Kleuker, *Grundkurs Software-Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten*. Springer, 2013, vol. 3.

[8] I. Jacobson, *Object Oriented Software Engineering: A Use Case Driven Approach*. Wokingham, UK: Addison-Wesley, 1992.

[9] N. B. Duncan, "Capturing Flexibility of Information Technology Infrastructure: A Study of Resource Characteristics and their Measure," *Journal of Management Information Systems*, vol. 12, no. 2, pp. 37–57, 1995.

[10] T. A. Byrd and D. E. Turner, "Measuring the Flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct," *Journal of Management Information Systems*, vol. 17, no. 1, pp. 167–208, 2000.

[11] S. Laan, *IT Infrastructure Architecture – Infrastructure Building Blocks and Concepts*. Lulu Press, Inc., 2011.

[12] H. Schackmann and H. Lichter, "Process Assessment by Evaluating Configuration and Change Request Management Systems," in *Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010 (WUP'09)*, 2009.

[13] S. C. Payne, "A Guide to Security Metrics," SANS Institute, Tech. Rep., 2006.

[14] M. Broadbent and P. Weill, "Management by Maxim: How Business and IT Managers Can Create IT Infrastructures," *Sloan Management Review*, vol. 38, no. 3, pp. 77–92, 1997.

[15] C. Soanes and A. Stevenson, *Oxford Dictionary of English*. Oxford University Press, 2010, vol. 3.

[16] P. Weill, M. Subramani, and M. Broadbent, "IT Infrastructure for Strategic Agility," Massachusetts Institute of Technology (MIT), Sloan School of Management, Tech. Rep., 2002.

[17] C. Straube and D. Kranzlmüller, "An Approach for System Workload Calculation," in *Proceedings of the 12th International IASTED Conference on Parallel and Distributed Computing and Networks (PDCN'14)*, 2014.

[18] R. Böhme and F. C. Freiling, "On Metrics and Measurements," in *Dependability Metrics*, I. Eusgeld, F. C. Freiling, and R. Reussner, Eds. Springer, November 2008, pp. 7–13.

[19] R. Böhme and R. Reussner, "Validation of Predictions with Measurements," in *Dependability Metrics*, I. Eusgeld, F. C. Freiling, and R. Reussner, Eds. Springer, November 2008, pp. 14–18.

[20] H. Koziolek, "Goal, Question, Metric," in *Dependability Metrics*, I. Eusgeld, F. C. Freiling, and R. Reussner, Eds. Springer, November 2008, pp. 39–42.

[21] C. Villarrubia, E. Fernández-Medina, and M. Piattini, "Metrics of Password Management Policy," in *Conference on Computational Science and Its Applications (ICCSA'06)*. Springer, 2006, vol. 3982.

[22] R. Ramler and K. Wolfmaier, "Issues and Effort in Integrating Data from Heterogeneous Software Repositories and Corporate Databases," in *Proceedings of the 2nd International ACM/IEEE Symposium on Empirical Software Engineering and Measurement (ESEM'08)*, 2008.

[23] C. Schaller, A. C. Neuroni, D. Mares, R. Riedl, and S. Urs, "Cockpits for Swiss Municipalities: a Web Based Instrument for Leadership," in *Proceedings of the 11th International ACM Digital Government Research Conference on Public Administration Online: Challenges and Opportunities (dg.o'10)*, 2010.

[24] Aruna Prem Bianzino and Anand Kishore Raju and Dario Rossi, "Apples-to-Apples: a Framework Analysis for Energy-Efficiency in Networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 3, pp. 81–85, 2010.

[25] D. Chen, N. Eisley, P. Heidelberger, S. Kumar, A. Mamidala, F. Petrini, R. Senger, Y. Sugawara, R. Walkup, B. Steinmacher-Burow, A. Choudhury, Y. Sabharwal, S. Singhal, and J. Parker, "Looking Under the Hood of the IBM Blue Gene/Q Network," in *Proceedings of the International ACM/IEEE Conference on High Performance Computing, Networking, Storage and Analysis (SC'12)*, 2012.

[26] T. Hoefler, T. Mehlan, A. Lumsdaine, and W. Rehm, "Netgauge: A Network Performance Measurement Framework," in *Proceedings of the High Performance Computing and Communications (HPCC'07)*, L. Yang, R. Mello, J. Subhlok, B. Chapman, and R. Perrott, Eds. Springer, 2007, vol. 4782, pp. 659–671.

[27] M. Meswani, M. Laurenzano, L. Carrington, and A. Snavely, "Modeling and Predicting Disk I/O Time of HPC Applications," in *Proceedings of the High Performance Computing Modernization Program Users Group Conference (HPCMP-UGC)*, 2010.

[28] J. Elliott, K. Kharbas, D. Fiala, F. Mueller, K. Ferreira, and C. Engelmann, "Combining Partial Redundancy and Checkpointing for HPC," in *Proceedings of the 32th International IEEE Conference on Distributed Computing Systems (ICDCS)*, 2012.

[29] I. Eusgeld, B. Fechner, F. Salfner, M. Walter, P. Limbourg, and L. Zhang, "Hardware Reliability," in *Dependability Metrics*, R. Reussner, F. C. Freiling, and I. Eusgeld, Eds. Springer, 2008, vol. 4909, pp. 59–103.

[30] B. Farbey, D. Targett, and F. Land, "The Great IT Benefit Hunt," *European Management Journal*, vol. 12, no. 3, pp. 270–279, 1994.

[31] M. Al-Mashari and M. Zairi, "Creating a Fit Between BPR and IT Infrastructure: A Proposed Framework for Effective Implementation," *Journal of Flexible Manufacturing Systems*, vol. 12, no. 4, pp. 253–274, 2000.

[32] C. Straube and D. Kranzlmüller, "An IT-Infrastructure Capability Model," in *Proceedings of the 10th ACM Conference on Computing Frontiers (CF'13)*. ACM, 2013.

[33] W. Smith, D. Acay, R. Fano, and G. Ratner, "Tools for Designing and Delivering Multiple-Perspective Scenarios," in *Proceedings of the 18th ACM Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments (OZCHI'06)*, 2006.

[34] Klaus R. Dittrich and Patrick Ziegler, "Three Decades of Data Integration - All Problems Solved?" in *Proceedings of the 18th World Computer Congress on Building the Information Society (IFIP)*, 2004.

[35] David M Pennock and Alexandrin Popescul and Andrew I. Schein and Lyle H. Ungar, "Methods and Metrics for Cold-Start Recommendations," in *Proceedings of the 25th International ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, 2002.

[36] H. Bauke and S. Mertens, *Cluster Computing: Praktische Einführung in das Hochleistungsrechnen Auf Linux-Clustern*. Springer, 2006, vol. 1.

[37] G. Contreras and M. Martonosi, "Power Prediction for Intel XScale Processors Using Performance Monitoring Unit Events," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'05)*, 2005.

[38] X. Fan, W.-D. Weber, and L. A. Barroso, "Power Provisioning for a Warehouse-Sized Computer," in *Proceedings of the 34th International Symposium on Computer Architecture (ISCA'07)*, 2007.

[39] M. Guest, *The Scientific Case for High Performance Computing in Europe 2012-2020*. Insight Publishers Ltd, 2013.

[40] D. Jensen and A. Rodrigues, "Embedded Systems and Exascale Computing," *Computing in Science Engineering*, vol. 12, no. 6, pp. 20–29, 2010.

[41] D. Hitchcock and L. Nowell, "Advanced Architectures and Critical Technologies for Exascale Computing," U.S. Department of Energy (DoE), Tech. Rep. DE-FOA-0000255, 2010.

[42] ISO/IEC 27001:2005, *Information technology – Security techniques – Information security management systems – Requirements*. International Organization for Standardization and International Electrotechnical Commission, 2005.

[43] "System Security Engineering - Capability Maturity Model – Model Description Document," Carnegie Mellon University, Tech. Rep. 3.0, 2003.

[44] Reijo Savola, "Towards a Security Metrics Taxonomy for the Information and Communication Technology Industry," in *Proceedings of the International IEEE Conference on Software Engineering Advances (ICSEA'07)*, 2007.

[45] Kenneth Chan and Iman Poernomo, "Consistent Metric Usage: From Design to Deployment," in *Proceedings of the Dependability Metrics: Advanced Lectures [result from a Dagstuhl seminar]*, 2008.

[46] S. Knittl, "Werkzeugunterstützung für interorganisationales IT-Service-Management - ein Referenzmodell für die Erstellung einer ioCMDB," Ph.D. dissertation, Technische Universität München (TUM), 2012.

[47] S. Bagchi, G. Kar, and J. Hellerstein, "Dependency Analysis in Distributed Systems using Fault Injection: Application to Problem Determination in an e-commerce Environment," in *Proceedings of the 12th International Workshop on Distributed Systems (DSOM'01)*, A. Pras and O. Festor, Eds., 2001.

[48] A. Avizienis, J.-C. Laprie, and B. Randell, "Fundamental Concepts of Dependability," University of California, Los Angeles (UCLA), Tech. Rep., 2001.

[49] C. Straube and D. Kranzlmüller, "Model-Driven Resilience Assessment of Modifications to HPC Infrastructures," in *Proceedings of the 6th Workshop on Resiliency in High Performance Computing (Resilience) in Clusters, Clouds, and Grids in Conjunction with Euro-Par 2013*, 2013.

[50] M. Sailer, "Konzeption einer Service-MIB - Analyse und Spezifikation dienstorientierter Managementinformation," Ph.D. dissertation, Ludwig-Maximilians-Universität München, 2007.

[51] V. A. Danciu, N. gentschen Felde, and M. Sailer, "Declarative Specification of Service Management Attributes," in *Proceedings of the 10th International IFIP/IEEE Symposium on Integrated Network Management (IM'07)*, 2007.

[52] Office of Government Commerce (OGC), *IT Infrastructure Library v3: Service Design, 2nd impression*. ISBN 978-0113310470, The Stationery Office (TSO), 2007.

[53] ISO/IEC 20000-1:2005, *Information technology – Service management – Part 1: Specification*. International Organization for Standardization and International Electrotechnical Commission, 2005.

[54] Christos Kozyrakis and Parthasarathy Ranganathan and Suzanne Rivoire and Mehul A. Shah, "JouleSort: a Balanced Energy-Efficiency Benchmark," in *Proceedings of the International ACM Conference on Management of Data (SIGMOD '07)*, 2007.

[55] Christos Kozyrakis and Justin Meza and Parthasarathy Ranganathan and Suzanne Rivoire and Mehul A. Shah, "Models and Metrics to Enable Energy-Efficiency Optimizations," *Computer*, vol. 40, no. 12, pp. 39–48, 2007.

[56] Robert Ayre and Jayant Baliga and Kerry Hinton and Wayne V. Sorin and Rodney S. Tucker, "Energy Consumption in Optical IP Networks," *Lightwave Technology*, vol. 27, no. 13, pp. 2391–2403, 2009.

[57] T. Talbot and L. C. Graff, "Verizon NEBS TM Compliance: TEEER Metric Quantification," Verizon Communications Inc., Tech. Rep. VZ.TPR.9207, 2009.

[58] Sujata Banerjee and Priya Mahadevan and Parthasarathy Ranganathan and Puneet Sharma, "A Power Benchmarking Framework for Network Devices," in *Proceedings of the 8th International IFIP-TC 6 Networking Conference (NETWORKING '09)*, 2009.

[59] Luc Ceuppens and Daniel Kharitonov and Alan Sardella, "Power Saving Strategies and Technologies in Network Equipment Opportunities and Challenges, Risk and Rewards," in *Proceedings of the International IEEE Symposium on Applications and the Internet (SAINT'08)*, 2008.

[60] A. Alimian, B. Nordman, and D. Kharitonov, "Network and Telecom Equipment - Energy and Performance Assessment – Test Procedure and Measurement Methodology," IXIA Corp / Lawrence Berkeley National Lab / Juniper Networks, Inc., Tech. Rep. Draft 1.0.4, 2008.

[61] "Energy Efficiency for Network Equipment: Two Steps beyond Greenwashing," Juniper Networks, Inc., Tech. Rep., 2010.

[62] INFOSEC Research Council, "INFOSEC Hard Problem List," http://www.infosec-research.org/docs_public/20051130-IRC-HPL-FINAL.pdf, 2005, [Online; accessed 30-May-2014].

[63] R. Böhme, "Security Metrics and Security Investment Models," in *Proceedings of IWSEC 201, LNCS 6434*. Springer, 2010, pp. 10–24.

[64] Andrew Jaquith, *Security Metrics — Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley Longman, Amsterdam, ISBN 978-0321349989, 2007.

[65] H. Langweg, "Framework for Malware Resistance Metrics," in *Proceedings of the 2nd Workshop on Quality of Protection (QoP'06)*. ACM, 2006.

[66] V. Ertürk, "A Framework Based on Continuous Security Monitoring," Master Thesis, The Middle East Technical University, 2008.

[67] W. Jansen, "Directions in Security Metrics Research, NISTIR 7564," National Institute of Standards and Technology Report, 2009.

[68] L. A. Gordon and M. P. Loeb, "The Economics of Information Security Investment," *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 438–457, 2002.

[69] Ronda Henning and Ambareen Siraj and Rayford B. Vaughn, Jr., "Information Assurance Measures and Metrics - State of Practice and Proposed Taxonomy," in *Proceedings of the 36th International IEEE Hawaii Conference on System Sciences (HICSS'03)*, 2003.

[70] L. Wang and S. U. Khan, "Review of Performance Metrics for Green Data Centers: a Taxonomy Study," *The Journal of Supercomputing*, vol. 63, no. 3, pp. 639–656, 2011.

Generic and Adaptable Online Configuration Verification for Complex Networked Systems

Ludi Akue, Emmanuel Lavinal, Thierry Desprats, and Michelle Sibilla
IRIT, Université de Toulouse
118 route de Narbonne
F31062 Toulouse, France
Email: {akue, lavinal, desprats, sibilla}@irit.fr

***Abstract*—Dynamic reconfiguration is viewed as a promising solution for today's complex networked systems. However, considering the critical missions actual systems support, systematic dynamic reconfiguration cannot be achieved unless the accuracy and the safety of reconfiguration activities are guaranteed. In this paper, we describe a model-based approach for runtime configuration verification. Our approach uses model-driven engineering techniques to implement a platform-independent online configuration verification framework that can operate as a lightweight extension for networked systems management solutions. The framework includes a flexible and adaptable runtime verification service built upon a high-level language dedicated to the rigorous specification of configuration models and constraints guarding structural correctness and service behavior conformance. Experimental results with a real-life messaging platform show viable overhead demonstrating the feasibility of our approach.**

***Keywords*—Network and Service Management; dynamic reconfiguration; configuration verification; online verification; model-based approach.**

I. INTRODUCTION

Complex networked systems and services are a fundamental basis of today's life. They increasingly support critical services and usages, essential both to businesses and the society at large. The evident example is the Internet with all its services and usages in a variety of forms, architectures and media ranging from small mobile devices such as smartphones to large-scale critical systems such as clusters of servers and cloud infrastructures. Consequently, it is indispensable to ensure their proper and continuous operation.

Network and Service Management (NSM) is a research and technical discipline that deals with models, methods and techniques to ensure that managed networked systems and services operate optimally according to a given quality of service. To cope with the increasing complexity of managed systems, NSM has evolved into self-management, a vision that consists in endowing management solutions with a high degree of autonomy to allow them to dynamically and continuously reconfigure managed systems in order to maintain a desired state of operation in the face of unstable and unpredictable operational conditions.

A main obstacle to the diffusion of dynamic reconfiguration solutions is the lack of standard methods and means to ensure the effectiveness of subsequent configuration changes and prevent erroneous behaviors from compromising the system's operation. This issue is particularly significant in today's mission critical systems management like cloud infrastructures, avionics, healthcare systems or mobile multimedia networks. This will also help increase users' confidence in the automation of reconfiguration, thus ease the adoption of ongoing autonomic solutions.

This article extends our recent work [1] on a model-based approach for online configuration verification with a running prototype architecture. It also provides additional concepts, methods and tools forming an online configuration verification framework. In particular, we describe how we use the framework to enrich a management system for a message oriented middleware platform with online configuration checking capabilities. Following the same process, the verification framework could be integrated with other existing management solutions.

Our approach to build this framework was first to define MeCSV (Metamodel for Configuration Specification and Validation), a high-level language, dedicated to the specification and verification of configurations. MeCSV allows operators to specify at design time, a platform-neutral configuration schema of their managed system with constraints guarding the desired service architecture and operation. One novelty of the metamodel is to include the capability to express both *offline* and *online* constraints. Offline constraints are typically structural integrity rules, that is, rules that govern a system's configuration structure. Online constraints concern service operation, they consist of rules to be enforced with regards to runtime conditions to avoid committing inconsistent configurations. An earlier version of the metamodel has been presented in [1].

Second, we have designed a runtime verification service, able to manipulate the concepts defined in this language. This service offers two interfaces, a verification interface for invoking configuration verification and an edition interface for managing specifications at runtime. The verification interface is flexible as it provides different operations to tailor configuration verifications to the usage scenarios, e.g., verifying only a subset of constraints regarding their severity or importance. The edition interface enables constraints updates at runtime to cope with changing management requirements.

These two phases allow our framework to support a verification process that starts at design time with a rigorous specification of verification models and continues at runtime through an automatic checking of configurations based on these models.

The rest of the paper is organized as follows: Section

II identifies runtime configuration verification requirements and positions existing works. We give an overview of the framework in Section III, and through a case study included in Section IV, elaborate on its building blocks in Section V and Section VI. We describe the integration of the framework with a real-life messaging platform in Section VII along with experimental results, proving the feasibility of the approach. Section VIII concludes the paper and identifies future work.

II. Background and Related Work

We begin this section by introducing the terminology used throughout the paper, then we expose the runtime configuration verification requirements in the current context of autonomous management approaches like in [2] and [3]. Finally, we present how those requirements have been addressed in related works.

A. Terminology

This section recalls definitions of key terms used throughout the paper.

1) Configuration: A configuration of a system is a collection of specific functional and non-functional parameters (also known as configuration parameters or configuration data) whose values determine the expected functionalities that the system should deliver.

2) Execution context: The execution context of a system comprises every element that can influence the system's operation. It includes both the system's technical environments, i.e., its interactions with other systems, its supported services and usages and the users' expectations, i.e., management and service objectives.

3) Operational state: The operational state of a system qualifies its observed operation in terms of the current values of its state parameters (or state data). The operational state is issued by a monitoring or a supervisory system. It reflects the behavior of the system relevant to the context at hand.

4) Dynamic Reconfiguration: Reconfiguration is the modification of an existing and already deployed configuration. Reconfigurations can be static, that is configurations are modified offline, when the system is not running. They can also be dynamic, that is configurations are modified online, while the system is running. We are especially interested in managed systems that are reconfigured dynamically.

5) Configuration verification: Configuration verification is the process of examining configuration instances against a set of defined requirements according to system architecture and service objectives. Configuration verification checks the correctness of proposed configuration instances, thus detect misconfigurations prior to changing the productive system.

B. Configuration Verification Requirements

Verification has always been critical to check that a given configuration meets its functional as well as non-functional requirements. When considering the lifecycle of a configurable system, in contrary to software verification that occurs mainly during the development phase of a system, configuration verification rather occurs in the use phase of a system to enforce its operation and maintenance (Fig. 1).

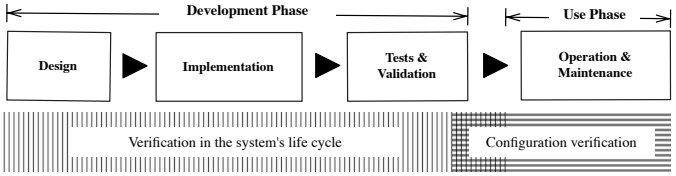


Fig. 1. Configuration verification in a simplified system's lifecycle

Configuration verification is traditionally done offline, either at design time, for example in test environments, or in production environments to enforce static reconfigurations. This verification is limited to structural sanity checks, typically testing the correct structure and composition of configuration parameters in terms of authorized values, consistent cross-components dependencies and syntactical correctness. This type of verification involves configuration data only. We have called it *structural integrity verification* [4].

By definition, dynamic reconfiguration implies configuration verifications should be carried out automatically at runtime for checking live configuration changes. Therefore, the verification process should take into account running operational conditions (1). It should also be flexible and adaptable to cope with the changing execution context in terms of architectural dynamics and changes of service objectives (2). Besides, it should accommodate the heterogeneity of management domains, representations and tools (3).

1) Operational Applicability Verification: Configuration verification should go beyond structural checks to assess the operational applicability of proposed configurations regarding runtime conditions at hand.

As systems adapt dynamically, ongoing operational states can invalidate the suitability of a produced configuration despite its structural correctness. For instance, one of the most common causes of a failed live virtual machine migration is not checking that the current physical resources of the destination host are sufficient before performing the live migration.

A runtime configuration verification should include *operational applicability verification*, that is, checking if a proposed configuration fulfills some running conditions [4]. This type of verification requires the knowledge of monitored operational state data. In other words, a runtime configuration verification should cover both *structural integrity verification* and *operational applicability verification*.

2) Flexible and Adaptable Verification: Configuration verification should be flexible and adaptable to cope with the changing execution context in terms of architectural dynamics and changes of service objectives.

The execution context of actual complex systems is highly dynamic in terms of operational conditions variations and dynamic usages where they are added, removed, migrated according to changing management and service objectives. This dynamicity implies configuration changes of different spatial and temporal scopes, e.g., local changes to system-wide changes, planned or spontaneous changes, punctual or life-long changes [5], [6].

A runtime configuration verification solution should thus accommodate these new characteristics by being flexible and

adaptable in order to be tailored to reconfiguration needs and usage scenarios, e.g., adapting the verification scope and perimeter.

3) Platform-independent Verification: Configuration verification should be platform-neutral to accommodate the heterogeneity of management domains, representations and tools.

Management applications domains are highly heterogenous in terms of the nature and importance of configuration data, (e.g., configuration data can be functional, non-functional, static or dynamic), their different representations due to different management standards and protocols, as well as the diverse nature of properties to be checked (e.g., hard constraints, soft constraint) according to diverse usage scenarios (mobility, performance, functional, non-functional) [7], [8].

In consequence, a configuration verification solution should be high-level and capable to integrate this heterogeneity.

C. Related Work

This section discusses existing works regarding the identified configuration verification requirements.

The need for configuration representation standards and configuration automation are growing concerns regarding the complexity of the configuration management of today's large-scale systems [9], [7]. Our work is at the junction of these two topics as the verification framework we provide enables a platform independent online configuration verification that is a prerequisite for configuration automation as well as dynamic reconfiguration.

Existing management standards like the Distributed Management Task Force Common Information Model (CIM) [10] and the YANG data modeling language [11] include constructs for configuration verification, yet their enforcement is left to implementors and solutions developers. Furthermore, those standards do not propose any mechanism for flexible and adaptable configuration verification as well as the management of the resulting verification lifecycle.

Beyond management standards, related works can fall into two groups: constraint checking approaches [12], [13], [14], [15] and valid configuration generation approaches [16], [17], [18].

In the first group, configuration experts are given a specification language to express some constraints that a verification engine checks on proposed configuration instances. In the second group, configuration decision is modeled as a Constraint Satisfaction Problem, adequate SAT solvers generate valid configurations or prove insatisfiability.

Both groups present common limitations, they do not address online configuration checking with regards to operational conditions: they provide only structural integrity verification that relies purely on configuration data or confines configuration verification to design time. They do not support a flexible and adaptable verification (e.g., only check a subset of constraints, modify and manage constraints during the reconfiguration life cycle). In addition, they mainly propose domain-specific tools with use-case specific verification (networks, distributed applications, JAVA applications, virtual machines).

Our work in contrast proposes a generic configuration verification approach that targets specifically online configuration checking, considering the influence of ongoing execution conditions on the verification process. It is thus profitable for complementing existing verification approaches.

In particularly, our work shares common foundations with SANChk [14], a SAN (Software Area Networks) configuration verification tool. They both use formal constraint checking techniques and enable a flexible and adaptable configuration verification. However, SANChk is specific to SAN configurations and does not target online configuration verification.

III. Online Configuration verification Framework

This section presents our verification approach and subsequent assumptions and concepts.

A. Assumptions

The following assumptions characterize the class of managed systems that we currently consider:

- The system is supposed known, observable, it is under a supervisory control that collects measures about its operating states and environment.
- The system is dynamically configurable, that is to say its current configuration can be altered at runtime if needed.
- The system's execution context is highly dynamic, hence is subject to sudden and often unpredictable variations.
- Either the supported management goals are clearly specified in order to derive properties to validate, or these properties are already defined.

B. Vision and Design Principles

The verification framework aims at offering an online configuration checking service that can be used by management solutions without changing existing tools. It specifically targets current autonomous and self-management approaches. As such, it purposefully addresses the runtime management of the systems' dynamics and the rapidly changing service and architecture requirements. The framework supports these new requirements through three main design principles according to the requirements exposed in Section II:

- Enabling an operational verification of configurations that takes into account their dependency on running execution states: in the context of self-adaptation, configurations are highly dependent on the operational conditions that can invalidate the suitability of a candidate configuration.
- Allowing modification of validity properties at runtime: management systems are likely to have their requirements evolve at runtime, and these evolutions are to be translated at runtime into the creation or modification of properties on configurations.

- Supporting existing management systems in order to enhance their reliability with a verification functionality. This can notably be achieved by integrating existing management standards such as CIM and YANG.

C. Integration within the Management Control Loop

Self-management is generally performed through a control loop called the MAPE (Monitor-Analyze-Plan-Execute) loop: the managed system is monitored (Monitor) to produce metrics that are analyzed (Analyze) to detect or prevent any undesirable behavior. Corrective changes are then planned (Plan) – either in the form of a new configuration or as a sequence of actions – then effected on the system (Execute) [2].

Runtime configurations decision is normally the responsibility of the *Plan* function. Consequently, a runtime configuration verification function that handles the two types of configuration verification is worthy to extend the MAPE loop to assure the validity of proposed configurations.

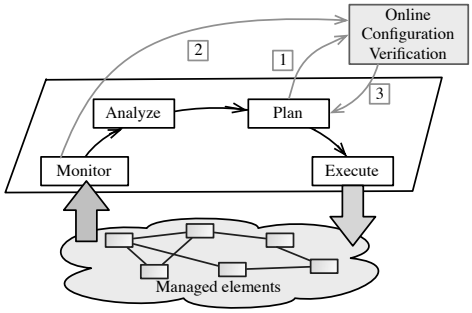


Fig. 2. The vision of online configuration checking

Fig. 2 illustrates our vision of a standard online configuration checking. An external and protocol-independent verification solution interacts with a management system (represented through the MAPE loop) through the Plan block (1), center of runtime configuration decisions, while relying on the Monitor block (2) for the retrieval of the required ongoing execution states, and thus processes an online verification of configurations (3).

As a result, the verification solution we propose, can extend any self-configurable system that does not have built-in online configuration verification. The only requirement for the self-configuring system is to allow access to its configurations and monitored data at runtime. Furthermore, this verification solution can be independently tuned and managed giving ongoing usage needs.

D. Overview of the Framework's Building Blocks

This section describes the building blocks of our verification framework. The framework supports a model-based approach for runtime configuration verification relying on a high-level specification language MeCSV and a verification service able to manipulate the concepts defined in this language.

1) MeCSV language: MeCSV is a metamodel dedicated to the formal modeling of configuration information for runtime verification. It offers platform-neutral configuration specification constructs including innovative features that enable

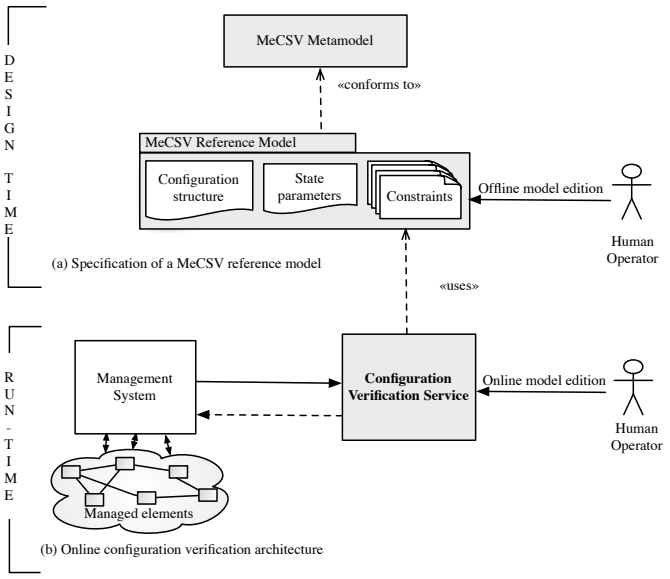


Fig. 3. Framework's approach and architecture

verification against runtime execution conditions. Even though MeCSV allows to model a system's configuration, it is intended for verification purposes only and is not suitable neither to model exhaustive management information nor to handle the management of the configuration's lifecycle (configuration data stores, configuration deployment etc.).

2) Target Domain Reference Model: The central objective of MeCSV is to allow the definition of a *Reference Model* that every possible configuration of the target system should conform to (Fig. 3 (a)). Operators or vendors can thus use the MeCSV metamodel at design time to define the reference model of a given managed application domain (e.g., an application server, a messaging middleware). Defined reference models are instances of the MeCSV metamodel, they are dedicated to a target application domain but do not rely on platform-specific representations. This reference model is to be defined only once, it will be processed at each reconfiguration decision, to dynamically evaluate configuration instances.

3) Verification Service: The framework includes a runtime architecture, Fig. 3 (b), with a verification service. This service needs to be initialized with the defined reference model. A related management system assuring monitoring and reconfiguration capabilities can then invoke specific operations at runtime to perform online verifications of decided configuration instances. This verification service also supports online modification of reference models to cope with the evolution of management and system requirements.

E. General Life Cycle of the Framework

The framework's building blocks support the following verification process:

1) At design time: A human operator, (e.g., an administrator or a configuration expert) uses the MeCSV metamodel to formally specify the *MeCSV Reference Model* of a given application domain (cf. Fig. 3 (a)). This reference model is made of a configuration schema of the domain, a relevant set

of operational state parameters to monitor and the offline and online constraints, necessary to enforce the structural integrity and operational applicability of configuration instances.

2) At runtime: this reference model will be used by the Verification Service for the automatic evaluation of proposed configuration instances (cf. Fig. 3 (b)). This online verification can evolve along with management objectives as the deployed reference model can be updated by human operators.

These two phases will be detailed subsequently in Sections V and VI, respectively.

IV. USE CASE

This section illustrates a Message-oriented Middleware (MOM) case study on which the examples given in the following sections will be based.

A. Introduction to MOM

A MOM system is a specific class of middleware that supports loosely-coupled communications among distributed applications via asynchronous message passing, as opposed to a request/response metaphor. They are at the core of a vast majority of financial services. Client applications interact through a series of servers where messages are forwarded, filtered and exchanged.

The middleware's operation involves the proper configuration of numerous elements such as message servers, message destinations and directory services. Involved configuration and reconfiguration tasks can be classified into two categories: first, setup operations that include defining the number of servers, where they will run and the messaging services each will provide. Second, maintenance operations that use the platform's monitoring metrics to adjust initial setups such as memory resources, message thresholds and users access.

By adding a management interface, an operator can monitor and tune the system's performance, reliability and scalability according to the monitored metrics (e.g., memory resources and users access) and management objectives.

B. JORAM's platform

JORAM (Java Open Reliable Asynchronous Messaging) [19] is an open source MOM implementation in Java. JORAM provides access to a MOM platform that can be dynamically managed and adapted, i.e., monitored and configured for the purpose of performance, reliability and scalability thanks to JMX (Java Management eXtensions) management interfaces.

Principal managed elements are message servers that offer the messaging functionalities such as connection services and message routing and message destinations that are physical storages supporting either queue-based (i.e., point-to-point) or topic-based (i.e., publish/subscribe) communications.

A JORAM platform can be configured in a centralized fashion where the platform is made of a single message server and a distributed fashion, the platform is made of two or more servers running on given hosts. A JORAM platform can be dynamically reconfigured, message servers can be added and removed at runtime. A platform configuration is described by an XML configuration file according to a provided DTD (Document Type Definition).

Fig. 4 shows a centralized configuration example, that will be further experimented in Section VII (Test Case 1). This configuration is made of one server, several middleware services (connection manager, naming service, etc.), two message queues and a user's permissions (note that due to space limitations, some configuration elements have been discarded).

```
<?xml version="1.0"?>
<config name="Simple_Config">
 <server id="0" name="S0" hostname="localhost">
  <service class="org.[…].ConnectionManager" args="root root"/>
  <service class="org.[…].TcpProxyService" args="16010"/>
  <service class="fr.[…].JndiServer" args="16400"/>
 </server>
</config>
<JoramAdmin>
  <InitialContext> […] </InitialContext>
  <ConnectionFactory> […] </ConnectionFactory>
  <Queue name="myQueue" serverId= "0"
         nbMaxMsg="200" dmq="dmqueue">
    <freeReader/> <freeWriter/>
    <jndi name="myQueue"/>
  </Queue>
  <User name="anonymous" password="passwd" serverId="0"/>
  <DMQueue name="dmqueue" serverId = "0">
     <freeReader /><freeWriter />
  </DMQueue>
</JoramAdmin>
```

Fig. 4. A Joram's configuration example

C. Verification Requirements

The following requirements are considered for the purpose of the case study, they encompass the manufacturer's set of configuration constraints and custom configuration constraints that guarantee memory performance. A valid configuration of the platform should provide the necessary messaging features in order for client applications to communicate efficiently. More precisely,

- Correct configuration structure: it should respect the platform's architecture and the relationships between the configuration parameters. (Req1)
- Object discovery and lookup: connection factories and destinations should be accessible via a naming service, i.e., the platform should provide an accessible JNDI service where the reference of the administered objects should be stored. (Req2)
- Memory optimization: message queues should not run low in memory, i.e., queues should not be loaded at more than 80% of their maximum capacity. (Req3)

V. MECSV METAMODEL

This section presents the salient features of the metamodel depicted in Fig. 5. MeCSV is organized in three categories of constructors: the first category is dedicated to configuration description, the second to operational state data description and the last for constraint expression.

A. Configuration Data Description

This part of the metamodel, depicted in Fig. 5 - Configuration, represents concepts to describe configuration data.

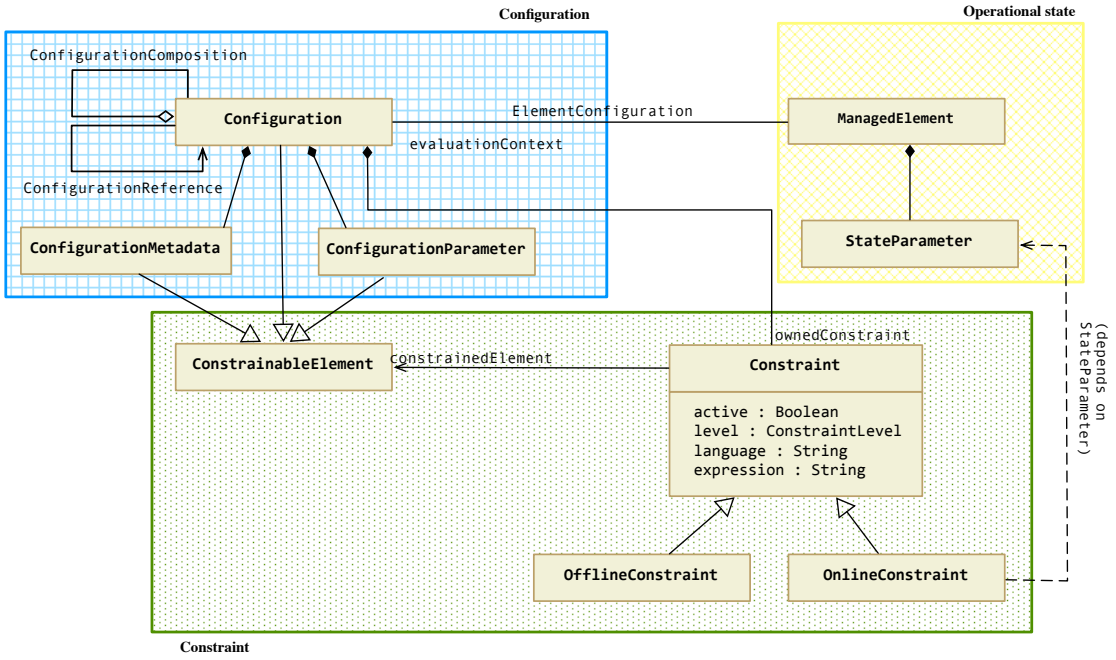


Fig. 5. Overview of the MeCSV metamodel

Configuration data are generally described in a set of configuration files where their structure is specified through the setting of some configuration properties with appropriate values and options.

There are a lot of bindings between the system's elements that should be reflected in their configuration: for example, the dependency of a server on its host machine should be specified by the coordination of the server's hostname value with the machine's hostname value.

These needs are addressed by the following constructs that are protocol and tool independent.

1) Configuration Parameter: represents quantifiable configuration parameters of managed elements; their expression defines the configuration data structure. A message server's identifier or hostname information are examples of its *configuration parameters*.

2) Configuration: allows to coordinate configuration parameters and group them in categories. Configuration elements act as containers for configuration parameters. For example, a configuration file can be modeled as a single *Configuration*, or for more flexibility, divided into multiple *Configurations*.

3) Configuration Dependency: to model bindings between two configuration elements, a *configuration dependency* should be defined between them. It means that a configuration parameter of some configuration references a whole or a part of another configuration. Typically, a server's hostname references its host machine's name information. It is an example of a *configuration dependency* between the server and its host.

4) Configuration Composition: this relationship allows to divide a main configuration into partial configurations. For example, a message server's configuration is logically split into message services, connection factories and destinations sub-configurations. These sub-configurations are linked to their parent configuration through a *configuration composition*.

5) Configuration Metadata: provides a means to specify metadata for configuration lifecycle management. For instance, one could want to tag specific configurations as default or initial. Another example is the `visited` metadata used in the JORAM platform to mark deployed configurations.

B. Operational State Data Description

As our work targets a global management environment where the managed system is both observable and reconfigurable, we provide constructs to represent information about managed elements as well as their monitored state. A knowledge of the monitored state is required to guide reconfigurations and to assert the operational compliance of proposed configurations. The following concepts allow to describe operational state data (Fig. 5 - Operational state).

1) Managed element: represents the notion of managed element like it is similarly defined in several management information models. A common pattern is to separate managed elements representation from configuration modeling; managed element representations containing monitoring-oriented information. In the case study, the message server is an example of *managed element*.

2) State Parameter: models the traditional operational state attributes like operational status, statistical data, in sum, any collected metrics about the system's operation. The current queue's load or the number of active TCP connections, are examples of state parameters.

In our approach, *Managed Element* and *State Parameter* are the necessary management building blocks for configurations and runtime constraints definition. Their values are supposed

to be provided by the existing monitoring framework. They are read-only elements contrary to configuration data.

C. Constraint Specification and Management

The following elements allow to define the constraints that configuration instances should respect as shown in Fig. 5 - Constraint.

1) Constraint: represents the restrictions that must be satisfied by a correct specification of configurations according to the system's architecture and management strategies. Constraints are boolean expressions in a given executable language. The *Constraint* element is subtyped into *offline* and *online constraints* to support the specificities of the two types of configuration validation.

2) Offline Constraint: represents structural integrity rules that a correct configuration data structure and composition of the system should respect. They can be checked either beforehand at design time or during runtime; they do not involve any check against operational conditions. For example, each message destination should have a JNDI name (in order to be looked up by client applications).

3) Online Constraint: defines rules for the operational applicability enforcement. *Online constraints* use *state parameters* values, their evaluation tests the configuration data against convenient state parameters. For instance, a queue's maximum capacity should be kept greater than the current number of pending messages.

4) Constraint Lifecycle Management: Constraints also have additional attributes for their life cycle management: they have a "constraint level" attribute to modulate their strictness. In particular, this allows to assign a severity level to the different constraints (e.g., high, medium, low) and an "active" attribute to activate or deactivate them depending on the operational context and management strategies (e.g., critical vs non-critical).

MeCSV has been formally specified as a UML profile [20]. UML constructs have been tailored to the MeCSV concepts to enable its usage in available UML modelers and to ease the adoption of the MeCSV language. Indeed, UML is well supported by many modeling tools and widely accepted as a standard modeling language.

D. Reference Model Specification Process

The specification process is a two-step process that occurs at design time: first the representation of the reference model structural classes, that is the representation of the configuration data and state data structure, and second, the expression of offline and online constraints.

The completion of these two steps provides the MeCSV reference model of a given management domain that is to be registered into the verification service. It will be used at runtime to check decided configurations.

1) Direct Modeling: Direct modeling is the general process for a MeCSV Reference Model specification. Operators install the MeCSV metamodel into a compliant model editor such as Eclipse Model Development tool (ECLIPSE MDT) and use MeCSV constructs to represent each part of the subsequent reference model.

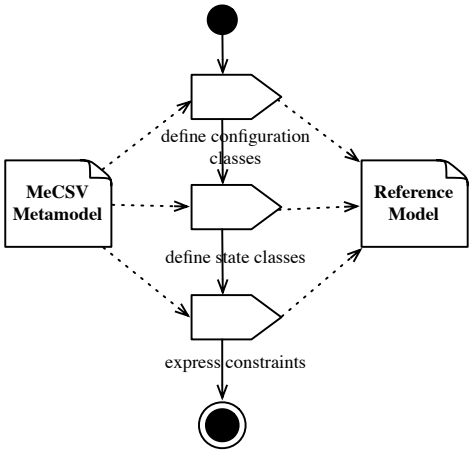


Fig. 6. Reference model design process: direct modeling

As it is shown in Fig. 6, they first describe the configuration parameters and state parameters organized into classes with convenient composition and dependency associations, then they specify the offline constraints that constrain the pure structure of configuration information and the online constraints that help evaluate the compliance of a given configuration information with the execution context at hand.

2) Model Transformation: This general specification process slightly differs when a management information model already exists (Fig. 7).

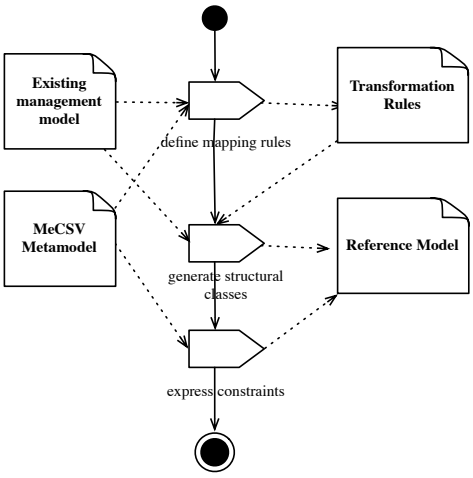


Fig. 7. Reference model design process: model transformation

Indeed, the first step can be automated, mapping rules can be directly defined between the specific management model and MeCSV, thus translating the legacy constructs into the related MeCSV ones. For example, one could use model-driven techniques such as model to model transformation or reflection for the implementation of such mapping rules. The second step of constraints expression remains identical.

VI. Verification Service Architecture

This section details the architecture of the runtime verification service and the resulting verification process.

A. Overview

Fig. 8 gives an overview of the main components of the verification service, a verification engine and a model repository, offering two interfaces, the verification and edition interfaces respectively for the usage of the service and the online edition of MeCSV reference models. The verification interface is supported by the verification engine and the edition interface enables modification of reference models stored in the model repository.

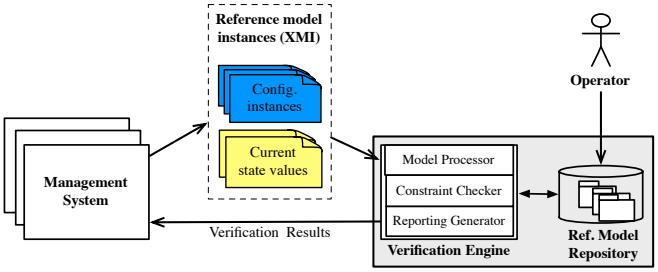


Fig. 8. Online Verification Service

1) Verification Interface: This interface is to be used by a management system to request the verification of configuration instances at runtime. It offers several functions, listed in Table I, that allow to trigger two types of verification: *i)* a complete verification where every existing constraint, in *active* state, is verified or *ii)* a selective verification where a specific subset of constraints are verified according to their type and severity level (e.g., online constraints with a highest severity level).

The verification engine is designed to process model instances conforming to an existing MeCSV reference model classes. Consequently, every API call must contain configuration instances and running operational state values described in a MeCSV-compliant format. Each call returns a verification result object including potential verification errors.

2) Edition Interface: This interface allows the registration of MeCSV reference models to the model repository. It also supports the online modification of registered reference models. Constraints can thus be added, removed, their status and severity can also be updated any time (Table I).

Note that the reference model is reloaded each time it is updated, allowing both the configuration structure and the set of constraints to be modified at runtime. This feature is particularly useful to add or remove constraints according to the management requirements that may change over time.

3) Reference Model Repository: The reference model repository stores MeCSV reference model classes and constraints to be processed during the verification. It also supports the usual creation, update, deletion and querying functions of a database, to adapt the model to evolving management requirements.

4) Verification Engine: The verification engine is the system component that checks provided configuration instances and reports inconsistencies. It provides three capabilities:

- A model processor, capable of analyzing and parsing model elements. It handles verification requests and ensures the existence of a related MeCSV reference model for received configuration instances.
- A constraint-checker, capable of checking dynamically received configuration instances against related reference model classes and available set of constraints. If a constraint is not satisfied, it notifies found errors to a reporting submodule.
- A reporting generator, capable of issueing an indication that contains flawed elements and violated constraints.

The verification engine is built-upon the open source Dresden OCL library [21]. Dresden OCL includes an OCL parser and interpreter that we have enriched with MeCSV specific features such as offline and online constraints differentiation and selective constraint checking, and with new capabilities like runtime modification of reference model constraints.

The *Verification Service* thus allows an existing management system to request verification of live configuration changes, It supports a single tenant as well as a multi tenant usage.

B. Online Verification Process

The following sequence diagram (Fig. 9) shows the interactions involved when a management system requests verification of some configuration instances. Two types of interactions can be identified: internal interactions between the decision and the monitoring modules of the management system and external interactions between the management system and the verification service.

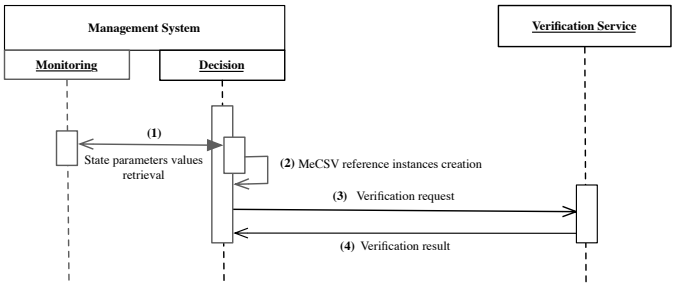


Fig. 9. Online Verification Process

When the decision module elects a configuration to verify, it first interacts with the monitoring module to retrieve current values of defined state parameters, Fig. 9 - (1), then it transforms those data into MeCSV-compliant instance models, Fig. 9 - (2), and sends them to the verification service, Fig. 9 - (3). The verification service checks received configuration instances both structurally and according to retrieved operational state instances of the step (2) and returns verification results, Fig. 9 - (4). It thus enriches management systems with a runtime configuration verification capability.

TABLE I. API CALLS SUPPORTED BY THE VERIFICATION AND THE EDITION INTERFACES

Interface	API call name	Functionality
Verification Interface (access to the Verification Engine)	`validateAll()`	Check given configurations against all existing constraints.
	`validateByConstraintType()`	Check given configurations against constraints of a certain type, e.g., online only.
	`validateByConstraintLevel()`	Check given configurations against constraints of a certain severity, e.g., fatal.
	`validateByConstraintFeatures()`	Check given configurations against constraints of a certain type and severity, e.g., online and fatal.
Edition Interface (access to the Reference Model Repository)	`registerReferenceModel()`	Register a MeCSV reference model.
	`updateConstraintStatus()`	Edit the status of a given constraint, e.g., deactivate a constraint.
	`updateConstraintLevel()`	Edit the severity level of a given constraint, e.g., decrease the severity.
	`updateConstraintFeatures()`	Edit both the status and the severity level of a given constraint.

VII. EXPERIMENT

This section describes the application of the verification framework to the JORAM case study presented in Section IV. Sections VII-A and VII-B describe how we have used the framework at design time to specify a reference model for the case study and how this reference model has been exploited at runtime to execute verification. Section VII-C evaluates this prototype experiment and Section VII-D discusses observations and results.

A. Design time: Reference Model Specification

Following the direct modeling specification methodology, exposed in Section V-D, we installed the MeCSV Eclipse Plugin in the ECLIPSE MDT model editor.

The different configuration concepts of Joram's DTD grammar have been modeled as adequate MeCSV-stereotyped UML classes, attributes and associations forming the configuration information model of the platform.

Constraints have been manually derived from requirements 1, 2 and 3 (cf. Section IV) and expressed in OCL. The following are examples of constraints that have been implemented:

- Each server should have an unique server id.
- Each server should provide a message destination and a connection factory (administered objects).
- Each administered object should have a JNDI name.
- A directory service (JNDI) should be available.
- The JNDI service should be activated and running.
- A queue should not be loaded at more than 80% of its maximum capacity.

The last two constraints are online constraints, they can only be evaluated at runtime, against operational conditions, thus requiring access to monitored data. This operational data has been identified (e.g., servers' operational status, queues' pending messages size, current number of client connections) and modeled as classes and attributes thanks to MeCSV *ManagedElement* and *StateParameter* stereotypes.

Fig. 10 shows an excerpt of the defined MeCSV reference model. This reference model subset contains the high-level configuration structure of a message server including a message queue, the offline and online constraints that should be respected and depending state parameters necessary to enable the operational applicability verification.

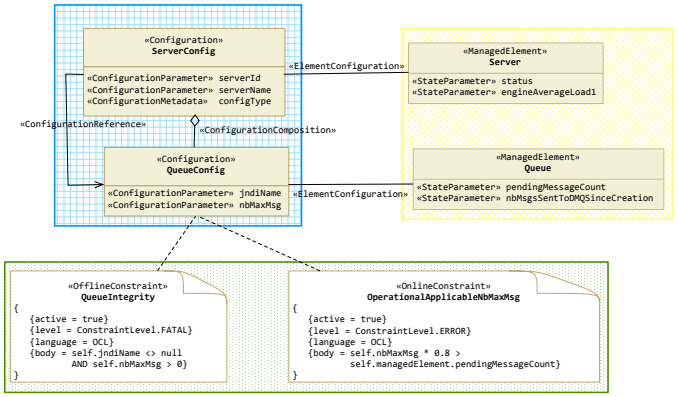


Fig. 10. Excerpt of the MeCSV reference model for a MOM application domain

B. Runtime: Online Verification Process

1) Dedicated Management System: For the verification process, we have set up a dedicated management system built upon the JMX management interfaces provided by the JORAM platform. The JMX interfaces comprise a monitoring interface allowing to collect metrics of interest about the running platform and a configuration interface capable of tuning the platform's configuration at runtime. Messaging servers, as well as messaging destinations, can be dynamically added or removed.

This management system is composed of a monitoring module and a decision module. The decision module is capable of choosing a configuration at runtime, requesting running operational data from the monitoring module and transforming these data into MeCSV-compliant instance models to be sent to verification.

In the same time, we implemented several client applications exchanging a high load of fictive messages to act on the monitored metrics (e.g., servers' average message flows, destinations' number of pending messages).

2) Verification Process: The verification process starts with the initialization of the verification service with the defined reference model. As for the management system, the decision module embeds different pre-defined configurations. At runtime, the decision module arbitrarily switches from one configuration to the other and requests the verification of its choice before deploying them.

Once the target configuration is selected, the management system follows the process previously illustrated in Fig. 9. It first retrieves running state values from the monitoring module,

then translates them in instances conforming to the defined reference model and finally requests their verification.

C. Experimental Setup

The goal of the experiments was both to test the ability of the MeCSV metamodel to serve as a formal specification notation and to evaluate the effectiveness of the verification service for processing online configuration checking against the defined reference model. We also measured the execution time of the verification process.

We performed our experiments on three different platform configurations varying in size and complexity, namely the number of system's elements (messaging servers, available services, message queues) and dependencies between them.

- The first configuration (Test Case 1) is a centralized messaging server offering basic message features for a total of nine configurable elements.
- The second (Test Case 2) consists of two messaging servers (about eighteen configurable elements).
- The third (Test Case 3) has three messaging servers and holds thirty configurable elements.
- To test the scalability of the validator, we defined a fourth configuration (Test Case 4), made of 300 managed elements, that has been programmatically tested with random state values variations.

Witness verification tests on correct configuration instances have also been conducted for each case.

The summary of test cases data is shown in Table II.

TABLE II. SUMMARY OF TEST CASES DATA

	Nb. servers	Nb. managed elements	Nb. configuration parameters	Nb. state parameters
Test case 1	1	9	57	25
Test case 2	2	18	110	64
Test case 3	3	30	197	103
Test case 4	30	300	1970	103

For each proposed configuration instance, we gradually ran complete verifications with ten, fifty and one hundred OCL constraints with a ratio of 80% offline constraints for 20% online constraints. For each verification request, we took 100 measurements of the execution time in milliseconds and computed the arithmetic mean.

Furthermore, we test selective verifications requesting the evaluation of specific subsets of constraints filtered according to their type and severity. We also test the online edition of constraints, verifying that the verification engine considers their modification.

The tests were run on a Intel® Core™ 2 Duo with 2.66 GHz and 4 Gigabytes of main memory.

D. Results and Discussions

1) Feasability: The verification service has been successfully tested: the received instances were checked against the stored reference model with both offline and online constraints violations detected and notified, both in the case of complete as well as selective verification requests. This permitted the decision module not to apply non-valid configurations.

The detection of online constraints violations, especially in the case of witness verification tests, confirms our thesis about *operational applicability verification*.

A first conclusion that can be drawn from these tests is the effectiveness of the verification service, thus the ability for MeCSV to be used to specify a real-life system's configuration schema and subsequent constraints for online configuration verification.

2) Verification Time: Concerning the verification time, the verification service has a noticeable but reasonable initialization overhead where the MeCSV reference classes and constraints are registered, but after this time, it processes constraint evaluations quickly.

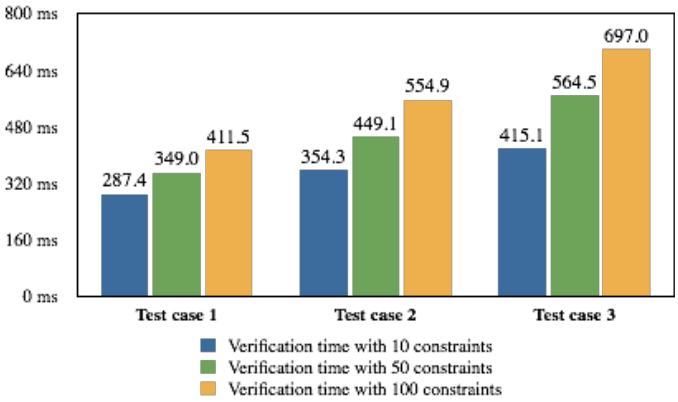


Fig. 11. Verification overhead for the first three test cases

The overall checking time for the three deployed scenarios is under 700 ms, which is very encouraging. It comprises the time taking to check the received instance conformance to the reference model, the constraint evaluation time and the reporting time (negligible).

A very important result lies in the effect of the number of system elements and the number of OCL constraints on the verification time. As shown in Fig. 11, the execution time is not proportional neither to the number of system's elements nor to the number of constraints. For example, while the size of elements quintuples from *test case 1* (6 managed elements) to *test case 3* (30 managed elements), their average verification time ratio hardly doubles (ratio is 1.73). Similarly, although the number of constraints increased by ten, the average verification cost is barely multiplied by 1.5. Further analysis of collected measures showed that constraints were checked in linear time.

We can conclude that in small configurations, the number of system's elements or the number of constraints scarcely affects the verification performance.

Furthermore, we observed that the error rate is not a factor impacting the verification time. An error-free configuration takes the same time as a highly erroneous configuration.

3) Scalability of the approach: The fourth case offers particular insights on the performance of the approach on a

very large configuration (Fig. 12). The worst case verification time of 30 message servers (2000 managements parameters and 100 constraints) is far below 2 seconds, which is still an acceptable time for a runtime verification program. This progression confirmed our first conclusion that the verification performance is not proportional to the size of configuration elements. While system's elements increased by 50 (from 6 managed elements to 300 managed elements), verification time increased by less than 5.

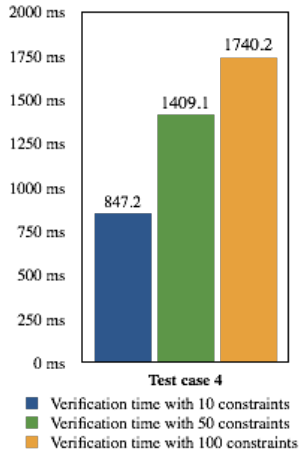


Fig. 12. Verification overhead for the scalability test case (Test Case 4)

The complete verification process overhead is very encouraging regarding the added capability to detect configuration errors at runtime. Indeed, even though configuration instances can be verified beforehand at design time, the difficulty to predict the varying operating conditions can compromise the success of runtime configuration changes.

Altogether, these experimental results confirm the importance of online configuration verification and show the feasibility of our verification framework to enrich a dynamically reconfigurable platform with runtime configuration verification.

VIII. Conclusion and Future Work

Designing lightweight online verification approaches is a critical requirement if we are to build reliable self-adaptive management systems and ease their adoption. This is fundamental as misconfigurations can be prejudicial to the proper operation of the system.

This paper presented a verification framework including an online configuration verification service relying on a high-level specification language named MeCSV. The framework aims to enrich existing management systems with platform-neutral and flexible configuration verification capabilities based not only on structural checks but also on running operational conditions.

We then described a methodology for using the framework from design time to runtime. We applied this methodology on a real-life message-oriented middleware case study where we successfully modeled the configuration schema, validity constraints and operational state data in a platform-independent fashion. This reference model was used by the verification service to process verification requests of configuration instances in viable time.

A series of verification experiments during reconfigurations allowed us to discuss results and observations demonstrating the feasibility of the approach. In future work, we intend to further experience the methodology and integrate more legacy systems so that we can ease the integration process and lower subsequent costs.

References

[1] L. Akue, E. Lavinal, and M. Sibilla, "A model-based approach to validate configurations at runtime," in *4th International Conference on Advances in System Testing and Validation Lifecycle (VALID)*, 2012, pp. 133–138.

[2] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[3] J. Strassner, N. Agoulmine, and E. Lehtihet, "Focale–a novel autonomic computing architecture," in *Latin–American Autonomic Computing Symposium*, 2006.

[4] L. Akue, E. Lavinal, and M. Sibilla, "Towards a Validation Framework for Dynamic Reconfiguration," in *IEEE/IFIP International Conference on Network and Service Management (CNSM)*, 2010, pp. 314–317.

[5] M. MacFaden, D. Partain *et al.*, "Configuring networks and devices with Simple Network Management Protocol (SNMP), RFC 3512," *IETF Request for Comment,[Online]*, pp. 1–69, 2003.

[6] B. H. Cheng, R. De Lemos *et al.*, "Software engineering for self-adaptive systems: A research roadmap," in *Software engineering for self-adaptive systems*, 2009, pp. 1–26.

[7] P. Anderson and E. Smith, "Configuration tools: working together," in *19th conference on Large Installation System Administration (LISA) Conference*, 2005.

[8] N. Samaan and A. Karmouch, "Towards autonomic network management: an analysis of current and future research directions," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 3, pp. 22–36, 2009.

[9] D. Oppenheimer, A. Ganapathi, and D. A. Patterson, "Why do internet services fail, and what can be done about it?" in *Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems*, ser. USITS'03, 2003, pp. 1–1.

[10] "CIM Schema version 2.29.1 - CIM Core," Distributed Management Task Force (DMTF), June 2011.

[11] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," Internet Engineering Task Force (IETF), RFC 6020, October 2010.

[12] A. V. Konstantinou, D. Florissi, and Y. Yemini, "Towards Self-Configuring Networks," in *DARPA Active Networks Conference and Exposition (DANCE)*, 2002.

[13] D. Agrawal, J. Giles *et al.*, "Policy-based validation of san configuration," in *5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY) 2004*, 2004, pp. 77–86.

[14] E. Gençay, C. Sinz *et al.*, "SANchk: SQL-based SAN configuration checking," *IEEE Transactions on Network and Service Management*, vol. 5, no. 2, pp. 91–104, 2008.

[15] P. Goldsack, J. Guijarro *et al.*, "The SmartFrog Configuration Management Framework," *ACM SIGOPS Operating Systems Review*, vol. 43, pp. 16–25, 2009.

[16] T. Hinrichs, N. Love *et al.*, "Using object-oriented constraint satisfaction for automated configuration generation," in *DSOM*, 2004, pp. 159–170.

[17] L. Ramshaw, A. Sahai *et al.*, "Cauldron: a policy-based design tool," in *7th IEEE International Workshop on Policies for Distributed Systems and Networks*, 2006, pp. 113–122.

[18] T. Delaet and W. Joosen, "PoDIM: A Language for High-Level Configuration Management," in *LISA*, 2007, pp. 261–273.

[19] "Java™ Open Reliable Asynchronous Messaging (JORAM)," OW2 Consortium, June 2013. [Online]. Available: http://joram.ow2.org/

[20] "OMG Unified Modeling Language (OMG UML), Superstructure V2.1.2," november 2007.

[21] "Dresden OCL," TU Dresden, Software Technology Group, June 2013. [Online]. Available: http://www.dresden-ocl.org/

Towards a Generic Framework of Engineering Design Automation for Creating Complex CAD Models

Gerald Frank, Doris Entner, Thorsten Prante, Vaheh Khachatouri
V-Research GmbH – Industrial Research and Development
Design Automation
Stadtstr. 33, 6850 Dornbirn, Austria
{Gerald.Frank, Doris.Entner, Thorsten.Prante, Vaheh.Khachatouri}@v-research.at

Martin Schwarz
Liebherr-Werk Nenzing GmbH
Technologiemanagement & Organisation Technik
Dr. Hans Liebherr Str. 1, 6710 Nenzing, Austria
Martin.Schwarz@liebherr.com

***Abstract*—For enterprises which engineer and produce highly customized products the reduction of design and manufacturing costs is of utmost importance. In this article, work towards a generic software framework for automating design processes resulting in complex customer-specific goods is described. Thereto, advanced product configurator user interfaces are tightly linked to CAD systems via an inference engine, which goes beyond product configuration in that it also facilitates generation of new parts. Knowledge-based engineering applications based on this framework support design engineers by automating those portions of a design process which are characterized by repetitive tasks. This is illustrated by two example use cases, namely design automation of ascent assemblies and box-type booms of cranes. On the application level, the implemented strategy is to reduce design-task complexity towards achieving significant speedups of up to 90 percent, which enables engineers to focus on creative, value-creating tasks. On the framework level, genericity and reusability is ensured by keeping the framework as CAD system independent as possible, by supporting different types of design procedures and complex assemblies, and by delivering added-value not only in the design and development phase but all along the process chain of integrated virtual product creation.**

Keywords—Engineering Design Automation; Knowledge-Based Engineering; Product Configuration and Generation; Software Framework; CAD.

I. INTRODUCTION

Production of industrial goods tailored to the requirements of ever smaller market segments and of individual customers has become a commonplace in many industries and is widely accepted as imperative to stay ahead of the competition when operating in highly developed markets. A direct consequence of that are higher complexities and smaller production batches, which can cause, in their turn, cost disadvantages. On the one hand, modular product design is widely advocated for combining the advantages of customization and flexibility with those of standardization and larger production batches. On the other hand, when it comes to products, where individualization/ customization entails complex engineering tasks to be performed per product and customer (termed engineer-to-order – ETO), other mechanisms of lowering costs while still maintaining the possibility of satisfying customers' specific needs have to be considered [1]. A powerful tool which is based on modularization and standardization, is the automation of (parts of) the engineering design process. This is generally termed *engineering design automation* and stands for numerous methodologies and applications in various industries, with the goal to automate the design process.

One approach to engineering design automation is knowledge-based engineering (KBE), which uses methodologies and technologies to capture and re-use knowledge of the product and its design process to reduce design and production time and costs. KBE can be seen "as a way of working intended to deliver engineering design

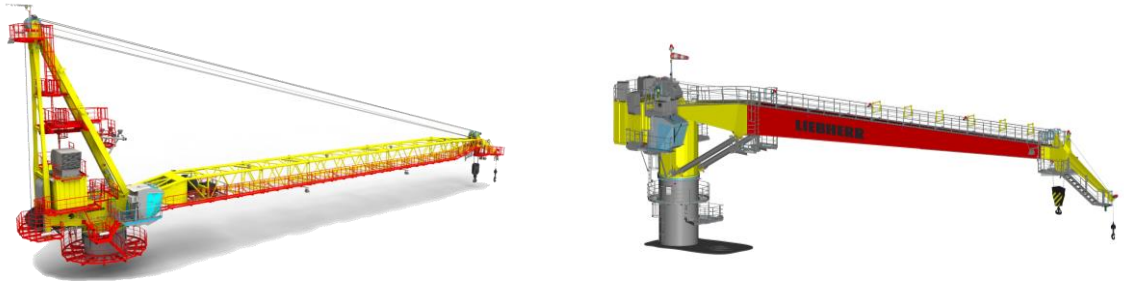


Figure 1. On the left, an example of a crane with attached ascent assemblies (highlighted in red); on the right, an example of a crane with a box-type boom (with the middle piece of the boom highlighted in red).

automation in scenarios where the retention of knowledge is critical" [2]. This includes the standardization of the parts/components of an artifact as well as the standardization and automation of the procedure which is used to assemble these parts.

Frank [1] presented a successful application of such an approach to ascent assemblies and box-type booms of cranes (Fig. 1). The application was realized in close collaboration with Liebherr-Werk Nenzing GmbH (LWN) [3], which is a manufacturer of a wide range of products including various types of cranes. The two mentioned use cases (ascent assemblies and box-type booms) are highly customized products, for which reason most orders involve the engineering department. Since this is a time intensive and thus very costly process, key aspects of the application are minimized design and production costs.

The KBE application is built upon a general framework, which uses a standardized part set as well as a rule base to represent the design knowledge. Furthermore, a structural analysis system is integrated in order to obtain the necessary statics requirements (for the use case of the box-type boom). The output of the application consists of a 3D CAD model as well as of the costs, bill of materials (BOM) and production drawings of the product.

This article further elaborates on the work presented by Frank [1]. First of all, the related work is more thoroughly reviewed and the approach of the paper is positioned clearly within the existing work. Furthermore, the underlying KBE system of the two mentioned use cases is discussed in more detail, giving an in-depth description of the steps from input to output as well as numerous illustrative examples. Finally, the general framework, on which these applications are based on, is also described in more detail.

The paper starts with a thorough overview of the related work concerning engineering design automation, including knowledge-based engineering and product configurators (Section II). After presenting the two mentioned use cases underlying our approach in Section III, a detailed explanation of the concept of the developed KBE application follows: In Section IV, an overview of the KBE system and its components (from input over the inference engine and the CAD model generation to the output) is given. The following Sections V to VIII discuss these building blocks of the KBE system in more detail. In Section IX, the underlying software framework is described. The paper closes with a demonstration of how the system is used in practice and its benefits for LWN (Section X), and a short conclusion and future work (Section XI).

II. Engineering Design Automation

The term of automated processes is most often associated with technological advancement in automation of production and manufacturing systems [4]. However, assuming the perspective of the overall product lifecycle, it becomes apparent that the highest potential of influencing costs lies in the earlier life phases of a product (Fig. 2; [5][6]). Namely, these are product and project planning (not shown in Fig. 2), design and development, and production planning, i.e.,

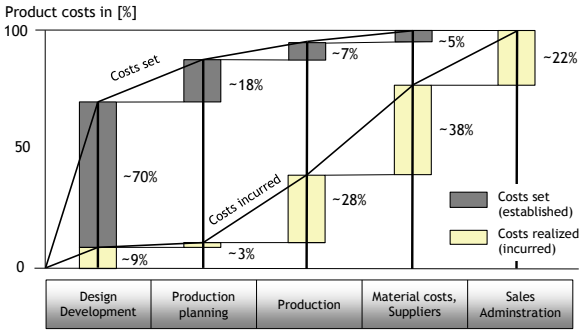


Figure 2. Determination of Production Costs.

generating/preparing production-ready documentation such as the production drawings and means, and manufacturing details. In abstracting the design process as a spanning tree to show the transformation of the problem specification into a detailed technical description through a series of design decisions [7], the causality within the design process is clearly indicated, emphasizing the importance of the initial stages as they effect significant alterations in the steps that follow. Accordingly, the main focus of *engineering design automation* (as specified in more detail in the following paragraphs) is on generating added-value inventions and innovations towards automated design and product development processes.

Engineering design automation comprises a wide range of methodologies and applications within various industries, and dates back many decades. In the early 1970s, *electronic* design automation was the first commercially successful application, allowing the automatic design of circuits and electronic chips which already then became too complex for human engineers [8][9]. The first CAD and CAE (computer aided engineering) systems appeared as well in the early 1970s [6][10]. Nowadays, CAx systems (with x being a placeholder for, e.g., design, engineering, planning or manufacturing) are available for all stages within the product lifecycle. Ever since these first successful applications, engineering design automation has been implemented in many fields, such as the automotive industry [11][12][13][14], in aerospace and aircraft design [15][16][17][18], as well as in mechanical and plant engineering, as demonstrated in the use cases of the here presented work and of some earlier papers [1][19][20].

Generally, engineering design automation requires a deep insight in the design process to be able to capture and formalize the principles in the design domain. This again typically requires a set of building blocks (i.e., components/parts or modules), which can be combined in certain ways to result in the product (or part of a product/ sub-assembly) fulfilling the customer's requirements. Depending on the purpose of the automation task, the assembling procedure can be fixed (e.g., given by a set of rules) yielding exactly one solution, or capable of exploring various assembling strategies (e.g., have a stochastic component) resulting in a solution space [21]. Creating several solution alternatives is essential in the early, conceptual phase of the design process, whereas routine

tasks are more prominent in the later, detailed design stage [22]. The approach of fixed assembling procedures is thus preferred for automating repetitive or routine design processes (consisting of nearly identical tasks), whereas methods capable of generating a set of possible solutions, by using different assembling strategies, are applied in innovative processes (including creative decisions; also termed computational design synthesis [21]).

In this paper, we concentrate on the problem of automating repetitive tasks, typical approaches for which are elaborated in the remainder of this section. We first review some literature on KBE and how it relates to expert systems and knowledge-based systems (KBS). Second, product configurators, which are one way of realizing a KBE application, are discussed. Finally, the approach of our system is outlined.

A. Knowledge-Based Engineering

The historical roots of KBE systems can be traced back to expert systems, which came up in the 1960s. The general term of an expert system is defined, according to Steinbichler [23], as a system that stores and accumulates specific knowledge of different areas and generates solutions in a user interface to given problems. Leondes [24] equates the terms "knowledge-based system" and "expert system". He also clarifies that a KBE system, the development of which is the topic of this paper, is a subset of a KBS.

In the field of KBE, methodologies and technologies are studied to capture and re-use knowledge of the product and the design process to reduce production time and costs, most often achieved through automating repetitive design processes [2]. The earliest ideas for KBE systems emerged in the late 1960s and early 1970s; more structured KBE systems have been around since the early 1980s [10][25].

According to Stokes [26], KBE can be defined as "the use of advanced software techniques to capture and re-use product and process knowledge in an integrated way." Thus, if applying the KBE approach, users' expertise has to be acquired and stored. A major advantage of this approach is that the captured knowledge is permanently available, and hence, the product development can be regarded as a holistic process. Thereto, all relevant design know-how is integrated into an overall product model, often stored in product data management (PDM) systems [27].

B. Product Configurators

In the context of automating repetitive processes, a common approach to the development of a KBE application is the use of a product configurator. *Product configurators* are defined in terms of finding a solution by combining components while fulfilling a set of constraints [28], or, putting it slightly differently, a configuration problem can also be understood as "the generation of a structure with predetermined properties by means of the combination of a certain number of objects" [29]. Bourke [30] expands this definition and describes a product configurator as "[...] software with logic capabilities to create, maintain, and use electronic product models that allow complete definition of all possible product options and variation combinations, with a minimum of data entries and maintenance". Another definition of product configurators is kept even more general, describing such a configurator as a tool assisting in the product design such that certain constraints are fulfilled [31]. In such definitions, parameterization of the components, i.e., dimensioning tasks such as adaptation of lengths and angles of a component, are considered to be part of the configurator as well. In [31], Brinkop presents a list of leading providers of product configurators in German-speaking markets.

Sabin et al. [32] classify product configurators according to their concept of configuration knowledge as rule-based, model-based and case-based product configurators. According to them, each approach represents the configuration knowledge and the instances of the product to be configured in a different way.

C. Our Approach

The KBE system approach presented in this paper not only uses complex product configurators, which allow for parameterization of the parts (e.g., adaptation of length or miter), but also incorporates the design of new parts (e.g., a handrail with arbitrarily angled corners formed from a basic straight part). This latter ability of the system to generate new parts goes beyond product configuration. The KBE system is based on an IT application of tested usability that supplies and processes knowledge and interacts with a CAD system.

The overall architecture represents a case of a customized system [33]. Keeping KBE functionality separate from CAD systems, while tightly interconnecting both via a bidirectional interface has a number of advantages. First, the captured and formalized design knowledge (in the form of parts and modular designs, and both simple and programmed rules as well as constraints) can be reused across different CAD systems. Second, also implemented and optimized variants of mechanisms processing and applying this knowledge remain available independent of the CAD system currently in use. The components guiding knowledge processing and application are, throughout this paper, collectively referred to as inference engine. For an overview, see the brown boxes at the center of Fig. 7 and the detailed discussions in Sections VI and IX.

Referring to Fig. 3, all kinds of input to the inference engine are also hosted within our KBE system/application and are thus kept CAD-system independent. This, third, gives us full flexibility to implement innovative user interfaces, without having to adhere to constraints set by design support for graphical user interfaces of any given CAD system. Beyond, fourth, the interface to structural analysis software is also managed by the KBE system.

Again referring to Fig. 3, the inference engine, at runtime, builds up a fully-fledged intermediate representation (i.e., an annotated tree structure, containing all necessary information about assemblies and assembly combinations), which can, fifth, be used to control different

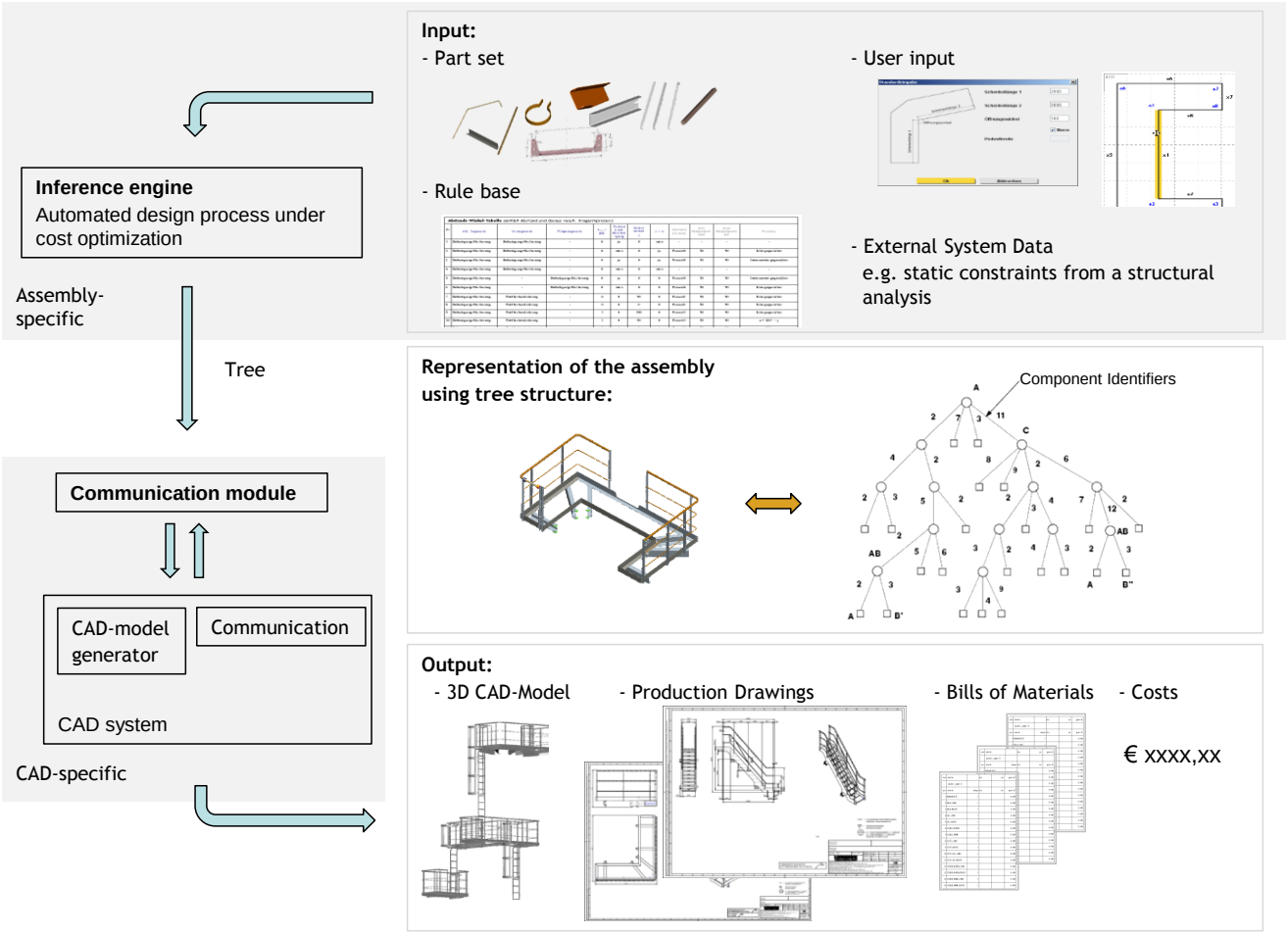


Figure 3. Overview of the KBE System, see Section IV for a detailed explanation.

CAD systems towards the generation of the customer-specific output shown in the lower part of Fig. 3. Here, the formation of CAD models and production drawings specifically necessitates the back channel to our KBE application, which is also indicated in Fig. 3. Finally, cost calculation is completely achieved within our KBE application.

Additionally, most of the just discussed functionalities and components are supported by generically implemented modules of our design-automation software framework discussed in section IX; exceptions to this are, e.g., our user-interface development environment and CAD-system application-programmer-interface (API). This, on top of the above presented advantages, allows for another level of reusability: Adaptations to other assembly types, differently tailored mechanisms of design-knowledge processing as well as combinations of such mechanisms.

In order to illustrate our approach and its advantages, applications to two use cases concerning the design of ascent assemblies and of box-type booms (see Fig. 1 and Section III for details) are based on the developed KBE system. The output of the KBE application comprises an automatically generated solution of the specified assembly, in accordance with a designer's input and statics constraints, and its visualization as a 3D CAD model. As indicated in the discussion as to system architecture, our concrete KBE systems thus consist of a user interface, the inference engine, and a module to communicate with the CAD system (Section IV). The rule base contains all information about a product, i.e., its structure, function and behavior as well as its manufacturability and quality. In case an engineer designs the CAD model manually, this knowledge is all that s/he needs to fulfill the task.

Furthermore, the whole design process of a specific product is supported: Besides a 3D CAD model, production drawings, BOM and production costs are provided. That is, the complete engineering process is automated.

III. USE CASES

The development of a KBE application is demonstrated by means of two use cases in the field of crane design and manufacturing, as collaboration between LWN and the industrial research center V-Research.

LWN is a manufacturer of a wide range of products including ship-, offshore- and harbor mobile cranes as well

as hydraulic duty cycle crawler cranes and lift cranes. Its mission is to fulfill customers' needs, and hence, to design and manufacture cranes according to the customers' requirements. While standardization is possible in many of their products, there are also segments which require customized adaptation of the crane to specific market demands (ETO). This can result in a partially or completely new engineered crane. In particular, the design of ascent assemblies for offshore, ship and harbor cranes as well as box-type booms for offshore and ship cranes requires substantial efforts. The costs related to the design of these assemblies are a major part of the overall engineering costs. Thus, we use these two types of assemblies as the use cases for our research for developing a KBE application to standardize and automate the engineering design task.

In Fig. 1, an example of each of the two use cases is given. In the left image, an offshore crane with attached ascent assemblies, highlighted in red, is shown. For maintenance, inspection and operating the crane, several strategic points on the crane, e.g., the machinery house, the pulley blocks, the winches and the operating cab, have to be easily accessible. The position of these access points depends on the crane design, which typically is customer specific. Therefore, an ascent concept, consisting of multiple ascent assemblies (platforms, roundplatforms, guardrails, stairs, or ladders), has to be developed anew for most orders. Each of these assemblies has a large range of variations in its specifications. For instance, platforms can have different shapes (rectangle, L-shape or U-shape, of almost any dimensions and angles), entries for ladders and stairs, and passages to connect to other platforms. The characteristics of the assemblies and their connections have to be specified by the designer according to the required access points.

The right image of Fig. 1 shows a ship crane with a box-type boom, the middle part of which is highlighted in red. The boom of such cranes has to be engineered to fulfill specific customer requirements, consisting of lifting capacity as well as of working and interference areas. These requirements are derived from the design of the ship or platform, on which the crane will be placed, and allow for little variation. Therefore, the boom section has to be designed individually for each application. This type of boom consists of a pivot, a middle and a head section. While the pivot and head sections are standard parts, the middle section, highlighted in red in the image and representing the second use case, requires custom engineering. Using a structural analysis (Subsection V.D), the thicknesses of the plates, which are assembled to form the boom of a pre-specified length, as well as the number of stiffeners and bulkheads to stabilize the boom have to be determined based on the customers' requirements (lifting capacity, working range, interference area).

The main goal of LWN was to reduce design and production costs by improving the design process in co-operation with the industrial research center V-Research. To achieve this goal, the possibilities of automating and optimizing the development phase were analyzed and, based on the results, a KBE application was developed. An overview of this application is given in the following section; the details follow in the subsequent sections.

IV. Overview of the KBE Application

Based on the explained background (Section II) and the requirements of LWN with regard to the two use cases (Section III), we developed a concept for automating the design process using a KBE system, including the integration of structural analysis. An overview of this approach is shown in Fig. 3, illustrated by the example of an ascent assembly. The details of the steps outlined in the figure are explained in the following sections. Here we just give a brief overview.

The *input to the KBE application* (Section V) consists of two types of information.

1. The first kind of input, consisting of the part set and the rule base, only depend on the assembly type (in Fig. 3 a platform), but not on the characteristics of the assembly (e.g., shape of a platform). The part set and the rule base result from formalizing and standardizing the engineering design knowledge (Subsections V.A and V.B). This type of input can be stored externally, e.g., in part templates or xml files.
2. The second type of input determines the characteristics of one specific artifact (e.g., the shape of the platform), and is obtained via interfaces. This information consists of the user input via the graphical user interface (GUI), e.g., the shape of the platform (Subsection V.C), and potentially some input from other external systems. For the use case of the platform, no such information is needed, however, for the box-type booms information about static constraints has to be retrieved from a structural analysis (Subsection V.D).

The input is then processed by an *inference engine* (Section VI), combining the information from the various input sources, as well as incorporating the logic for minimizing the costs. Both the input and the inference engine have to be adapted, based on the considered assembly (e.g., different rule base and user interface for a platform and a ladder). The inference engine includes procedures for generating new assemblies (Subsection VI.A), combining several assemblies (Subsection VI.B), and adapting already generated assemblies or combinations of assemblies (Subsection VI.C).

The output of the inference engine is an internal representation in from of a tree, representing the assembly and containing all necessary information: Every node of the tree represents a part of the assembly (or a subassembly, being a tree of parts itself) containing its geometry (e.g., length), its costs and its positioning information in reference to its parent part.

This internal tree representation is handed over to a CAD interface module to transform the information stored in the tree into data to generate the CAD model (Section VII). The communication module (and of course the CAD model generation) is specific to the chosen CAD system.

The output of the system is a 3D CAD model of the assembly, production drawings, BOM and the costs of the assembly (Section VIII).

The KBE application is based on a generic framework, which is explained in more detail in Section IX. As discussed in Section II.C, its core component, the inference engine, is designed such that only assembly specific tasks have to be changed. It is also possible to support different types of design procedures (e.g., bottom-up vs. top-down). Furthermore, the output in form of the internal tree representation, the generation of the CAD model, the calculation of the costs, creation of the BOM and the production drawings are as far as possible generically implemented. The communication module is also generic with regard to the assembly, and CAD system specificity is kept to a minimum, as the engine operates on a fully fledged intermediate tree representation for lossless bidirectional data exchange between framework and CAD systems. The thereby gained flexibility allows us to go beyond product configuration as its pure application is often too simplistic to cope with complexities encountered in today's engineering projects. As an example, the generation of new parts was discussed. Finally, on the other hand, graphical user interfaces are not yet implemented based on a framework or toolkit.

V. INPUT TO THE KBE APPLICATION

To model an engineering design process, it is necessary to investigate all factors that influence the design. The combination of these factors will lead to restrictions that have to be taken into account when formalizing the knowledge for the input of the KBE application. The main restrictions for engineering assemblies are the following:

- Industry and company standards,
- Statics requirements,
- Production costs,
- Implicit design restrictions (e.g., assembly erection or maintenance aspects), and
- Production restrictions (e.g., disposal factors).

For example, when designing a platform, the restrictions of the above bullet points result in platform entries conforming to standards (width, closing), or in special assembling logics, e.g., how to position the stays for the guardrail of a platform, how many pipes the guardrail requires and in which height they are positioned at the stays. Furthermore, the minimal and maximal realizable bending radius of a pipe is restricted by the manufacturing department.

A main challenge in building a KBE application is to extract the often implicit knowledge of the design engineers and formalize it appropriately. We next discuss this problem along the approach taken for both the ascent assemblies and the box-type booms.

A. Knowledge Acquisition

To build a KBE application, the relevant engineering expertise has to be acquired. As mentioned in Section II, engineering processes can be differentiated into repetitive and creative processes. In contrast to creative processes, repetitive ones consist of nearly identical tasks and are therefore independent of creative decisions. This condition is necessary for modeling the knowledge as a system of rules. In contrast to repetitive processes, creative ones occur typically only once. Because of that, modeling them as rules within reasonable time is economically not viable.

The main technique used to capture all relevant steps for designing the focused-on assemblies were interviews with the engineering experts at LWN [34]. In an iterative manner, interviews were conducted, the information was formalized, as well as the results were verified and additional questions were clarified in further interviews. The retrieved information served as a base for analyzing the repetitive design processes. Most of the time spent was used for identifying the restrictions of the bullet point list above.

Through the investigation of the design process of ascent assemblies and box-type booms, we found that the design process is indeed mainly based on repetitive tasks. One of the goals defined by LWN was that a specific repetitive design task should always result in the same, ideal solution. Because of the limited ability of a human to re-execute cognitive tasks identically, it is important to support users with a tool (i.e., a software application). Towards this end, the acquired knowledge has to be formalized, which is discussed next.

B. Knowledge Modelling – Part Set and Rule Base Design

The data obtained in the interviews first revealed that for the used building blocks of the assemblies a high amount of part variants existed (e.g., cantilever arms of arbitrary lengths). This, in turn, led to high costs not only in production but also in administration of the parts. To reduce the number of part variants, we developed a fixed set of standardized parts, which is sufficient for designing all required assemblies. This set of standardized parts contains not only fixed components (e.g., screws, stays, cantilever arms of discrete length), but also components which are capable of parameterization (e.g., adjusting to arbitrary length or miter of a component) and which may be generated completely new (e.g., a handrail with arbitrarily angled corners).

Based on the standardized part set, the knowledge about the design process was analyzed. Since the considered assembling processes are repetitive ones, designing these assemblies is based on a set of invariant rules that can be modeled and stored in an IT system. These rules represent directed dependencies in a form common for KBE systems, namely "IF (condition/-s) THEN (action/-s)"-statements, i.e., all conditions must be known and fulfilled before a rule can be applied [29].

Both the part set and the rule base are assembly specific. However, the approach of formalizing the knowledge in a rule set can be used for arbitrary types of assemblies. Rules can be changed without editing the source code. In addition, if a wide range of rules is acquired, nearly every form of assembly is supported. Therefore, repetitive tasks in

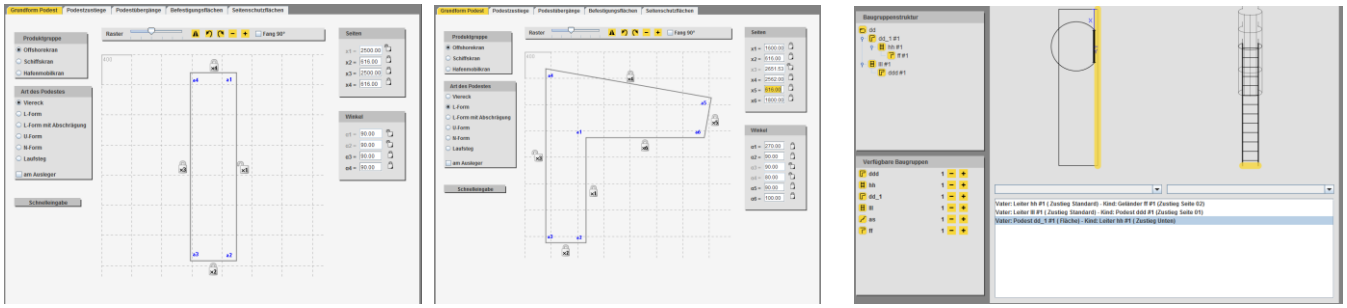


Figure 4. Graphical User Interface: Example of an interactive sketch for assembly dimensioning (left and middle image) and a wizard for the definition of an assembly-combination (right image).

designing new or adapting existing assemblies can be automated. This enables engineers to focus on creative, value-creating activities [35].

In the use case of ascent assemblies, an important subset of the rules belongs to the static requirements of the assembly. Each ascent assembly contains specific components (e.g., the cantilever arms for platforms, or stringers for stairs) which carry the main static load of the assembly and thus ensure the adequacy of the design with regard to safety and overall requirements of the structure. Based on these components, there is a limited set of parts with a fixed geometry (e.g., cantilever arms of discrete lengths). As a consequence, all statically relevant components can be pre-calculated using suitable software. The resulting parameters, e.g., the maximum load per square millimeter or the maximum gap to the next structurally relevant component, can be pre-assigned and therefore stored in rules. Based on the pre-calculated static parameters and the dimension of a given assembly variant, the number and/or dimensions of these parts are defined.

While the statics constraints of ascent assemblies can be represented by rules, the statics calculations of box-type booms are more complex. To verify static stability of these assemblies, dedicated structural analysis simulation algorithms have to be integrated into the design process (Subsection D).

C. User Interface

The GUI is an important component of the developed application. The focus was on minimizing user input, with the goal to allow users to define an assembly as efficiently as possible. The number and kind of input parameters depend on the specified assembly and the underlying rule set, such that design engineers only have to provide data which cannot be retrieved automatically. Furthermore, the user interface is supported by interactive sketches, and inputs are immediately visualized.

One tab of the user interface of the platform assembly is shown in Fig. 4. The interactive sketch in its initial form shows the starting-point platform layout. When changing, e.g., the shape and dimensions of the platform, this is immediately visualized, as shown in the middle image of the figure. Besides the geometry of the platform, various functionalities of the platform can be defined on further tabs of the user interface. These functionalities include for example platform entries and passages with arbitrary positioning (aligned at the left, right, or somewhere in the middle on the specified side of the platform), and several ways of closing (droplatch, gate, no closing). Which options are actually provided via controls in the GUI (e.g., if and how the geometry and functionalities of an assembly can be changed) depends on the specific type of assembly.

Every irregularity as to a defined process is highlighted by interaction dialogs. For example, if a design engineer defines inconsistent data, the application alerts the user.

In addition, a user is supported by some assisting tools. One of them is concerned with the combination of assemblies: a wizard visually supports the user to form a valid combination of assemblies (e.g., to define a complete access solution for an entire crane). The wizard shows all and only those combinations which are valid (e.g., it shows the possibility to attach a ladder or stairs to an entry of a platform, but not to connect stairs to a ladder). The right image of Fig. 4 shows the combination wizard for attaching the bottom of a ladder to the inside of a platform.

D. Structural-Analysis Software Integration

Structural analysis of an assembly is a major task in engineering design, in order to ensure the stability and safety of an artifact. In [36], the author defines structural analysis as follows: “Structural analysis is a process to analyze a structural system in order to predict the responses of the real structure under the excitation of expected loading and external environment during the service life of the structure.”

As explained in Subsection B, in the use case of ascent assemblies, the problem reduced to a few components of the ascent assembly, and all statics requirements could be pre-calculated and captured in a rule base. For the use case of box-type booms, the situation is much more complex, since all components of a boom are structurally relevant, and thus each individual box-type boom requires a separate structural analysis.

Because of the market-segment’s specific requirements and due to LWN’s commitment to fulfill customers’ demands (i.e., to provide arbitrary lengths and loads of the box-type booms), the dimensions of the boom vary considerably and cannot be limited to a standard set of parts. Thus, a structural pre-calculation of every possible

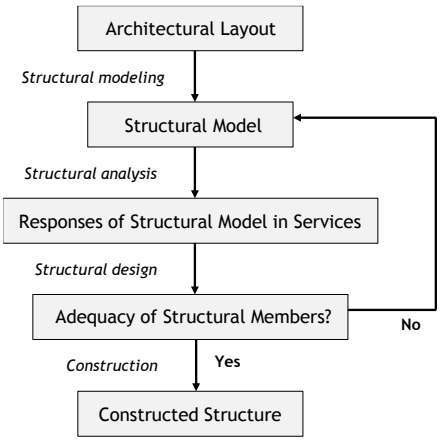


Figure 5. Structural analysis process.

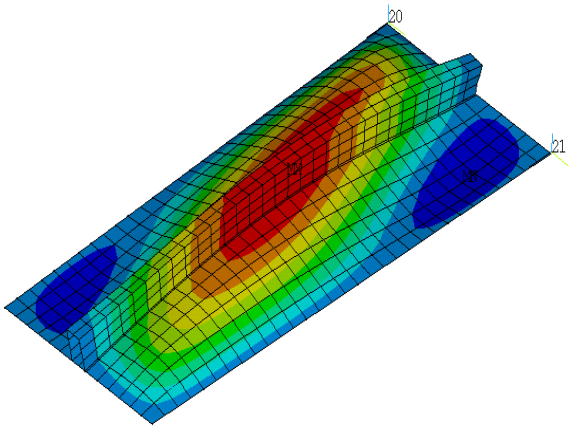


Figure 6. Result of a buckling analysis of a base plate section of a box-type boom.

dimension of the individual components and possible lifting capacities is not possible since the boom has to be considered as a complete system. Therefore, the logic regarding statics for box-type booms cannot be mapped to simple rules regarding its components. Hence, the developed KBE system integrates a structural analysis system (ANSYS [37]) to automatically obtain the necessary input regarding the statics requirements (e.g., thickness of the plates).

A standard structural-analysis process is shown in Fig. 5 [36]. This process applies to box-type booms at LWN in a similar way. In a first step, the design manager of the project converts the customer requirements into load cases. A load case mainly consists of a boom position (inclination angle) and a load capacity. After taking into account additional factors, a set of load cases is generated.

Based on this input, a simplified model is generated and processed by the structural analysis software ANSYS. In ANSYS, the model is analyzed with all the load cases. Based on multiple iterations, the defining parameters of the components (e.g., plate thickness) are optimized. As an example, Fig. 6 shows a visualization of the buckling analysis of a box-type boom base plate section. With the analysis it is determined how many stiffeners will be needed in a section to fulfill the requirements.

The output of ANSYS is an iteratively calculated optimal structure of a box-type boom: for each section, the material dimensions, part quantities and positions are defined. In particular, the calculated assembly structure consists of a weight-optimized geometry, i.e., the amount of used material is minimized. However, due to the nature of the boom production processes, a weight-optimized geometry is not necessarily cost-optimized, since production steps (e.g., welding) are not considered. Based on the results of the structural analysis, we developed an algorithm which uses a defined set of rules to translate the calculated geometry into a low-cost structure, while still adhering to the boundaries of the statics calculation. The parameters of the resulting structure (length and thickness of the plates) are used as input to the inference engine.

As the existing structural analysis procedure is a very time-consuming, complex and effort-intense process, its efficiency was increased by automating nearly all manual activities and integrating them into the developed KBE system. To supply the structural analysis software with all relevant information, a standardized data exchange format was developed. Now, the only manual activity consists of defining the load cases based on the customer's requirements. The developed method then transforms these data into an ANSYS-suitable configuration and hands them over to the structural-analysis simulation application. Once the simulation has started, no further user interaction is necessary. At the end of the simulation process, the structural engineer receives all the data for double-checking. For returning the results to the KBE system, an additional interface format was developed.

VI. Inference Engine

The inference engine is the heart of the KBE application. It processes the input by combining the information of the standardized part set, the modeled rule base, the user input and potentially the input from the structural analysis. The methods and algorithms of the inference engine are developed such that the assembling of the single parts to the specified assembly happens in a cost-efficient way, i.e., an optimization logic (based on rules) is implicitly included. The content of the inference engine is assembly-specific; however, the structure follows a general framework (see Section IX).

The goal of the inference engine is to store all the information of an assembly in a framework-hosted general intermediate representation, which can be used to communicate this information to the CAD system. The chosen structure is a tree in which each node represents a part of the assembly (or a subassembly, being a tree of parts itself) containing its geometry (e.g., length), its costs and its positioning information in reference to its parent part. In the following subsections, we discuss the way the tree is built

and the advantages of using such a structure when it comes to combining and adapting assemblies.

A. Generation of a Single Assembly

For the use case of box-type booms, only a single assembly (the box-type boom) has to be generated. In case of ascent assemblies, generating one single assembly (or several, not connected assemblies) is the simplest case. Typically, however, several assemblies (e.g., platform, ladder, stairs) are generated and combined, (see Subsection B for the combination).

For each assembly type (e.g., box-type boom, platform, ladder), one part is defined as the head-part, i.e., the root of the tree structure. According to the rule base and the design logic of the inference engine, all other parts are iteratively added to the tree as a child of an existing part. For example, the handrail is added as a child of the stay, from which the handrail starts. When determining the characteristics of a part, the following information is stored in the tree:

- The *geometry* of a part can be specified in two ways: (1) a part may be parameterized (e.g., by adjusting its length), or (2) a part may be generated from scratch by defining the required data (e.g., start, ending and corner points of a handrail, as well as an angle for each corner point for bending the handrail appropriately). For some parts, the geometry cannot be changed (e.g., bolts).
- The *positioning* information of each part contains the parent part in the tree (i.e., the part at which the current part is positioned in the CAD model), as well as the constraints how the current part is positioned in reference to its parents part (e.g., aligning two surfaces). The head-part of the tree is placed on a default position.
- Finally, the *costs* of each part, consisting of the material and production costs, are also stored in the tree.

B. Combination of Several Assemblies

As mentioned in the previous subsection, in the use case of ascent assemblies, typically several assemblies are generated and appropriately combined to form a complete ascent solution for a crane. For example, an ascent assembly may consist of a ladder, which connects to a platform, from which in turn stairs are mounted to reach another platform. As discussed in Subsection V.C and illustrated in Fig. 4 (right image), the definition of the combination is done via the GUI.

Each assembly (platform, ladder, etc.) is represented by a tree, as described in Subsection A above. To combine these assemblies to a single ascent solution, their trees have to be merged appropriately: For each such combination it is defined, how the two assemblies have to be connected, i.e., which part(s) of the first assembly is/are used as reference part(s) for which part(s) of the second assembly. Then, the trees of the assemblies can be combined accordingly into one bigger tree by simply adding the additionally required parent-child relationships between the corresponding parts of two assemblies.

C. Adaptation of Assemblies

For the use case of ascent assemblies, a further functionality of the inference engine is the adaptation of already generated CAD models of the KBE application (see Section VII for the generation of CAD models). If the engineer detects some inaccuracies or errors in the generated CAD model, the engineer could correct these manually in the CAD system, but also has the possibility to adapt the input in the GUI and re-generate the CAD model. In this case, not the complete model is re-generated, but only the necessary parts to reach the corrected solution are updated. This is done by comparing the trees of the initial and the updated model by traversing them systematically. In case of a detected difference in the two trees, the corresponding change is realized in the CAD model.

Such an adaptation could for example be necessary when a complex combination of ascent assemblies is mounted to a crane (in the CAD system), and intersections of the ascent assembly and the crane are detected. By, e.g., adapting the length of a platform, or replacing a ladder with stairs the mistake could be fixed (a detailed illustration is shown in Section X).

For the use case of ascent assemblies, the adaptation of assemblies or a combination of assemblies is of great importance since the generation of the CAD model is rather time intensive (many single parts have to be loaded and positioned in the CAD system). Using the adaptation generation only updated or new parts are loaded and positioned, saving a substantial amount of time. In case of the box-type boom, such an adaptation generation is not necessarily needed, since box-type booms consist of relatively few parts, and a major part of the calculation time is required for the statics calculations, not for the CAD model generation.

VII. CAD System Interface

Once an assembly is defined by the user, and, if necessary, the structural data are calculated, the respective data are handed over to the inference engine. After calculating all necessary information for generating a 3D CAD model and storing it in a tree structure, the computed data is sent to the CAD software in an iterative way (i.e., traversing the tree systematically). The communication between the inference engine and the CAD system is realized by using the API of a CAD system. The communication module encapsulates the handing over of the tree representation to CAD systems. This, being the only interface element, implements a clear and narrow interface as the basis for as much CAD system independence as possible.

When the data of a part in the tree is being sent to the CAD system, the part is first loaded and, if necessary, the geometry is adapted. Then, the part is positioned in reference to an existing part to ensure that all parts refer to each other. This is important since then every manual change directly

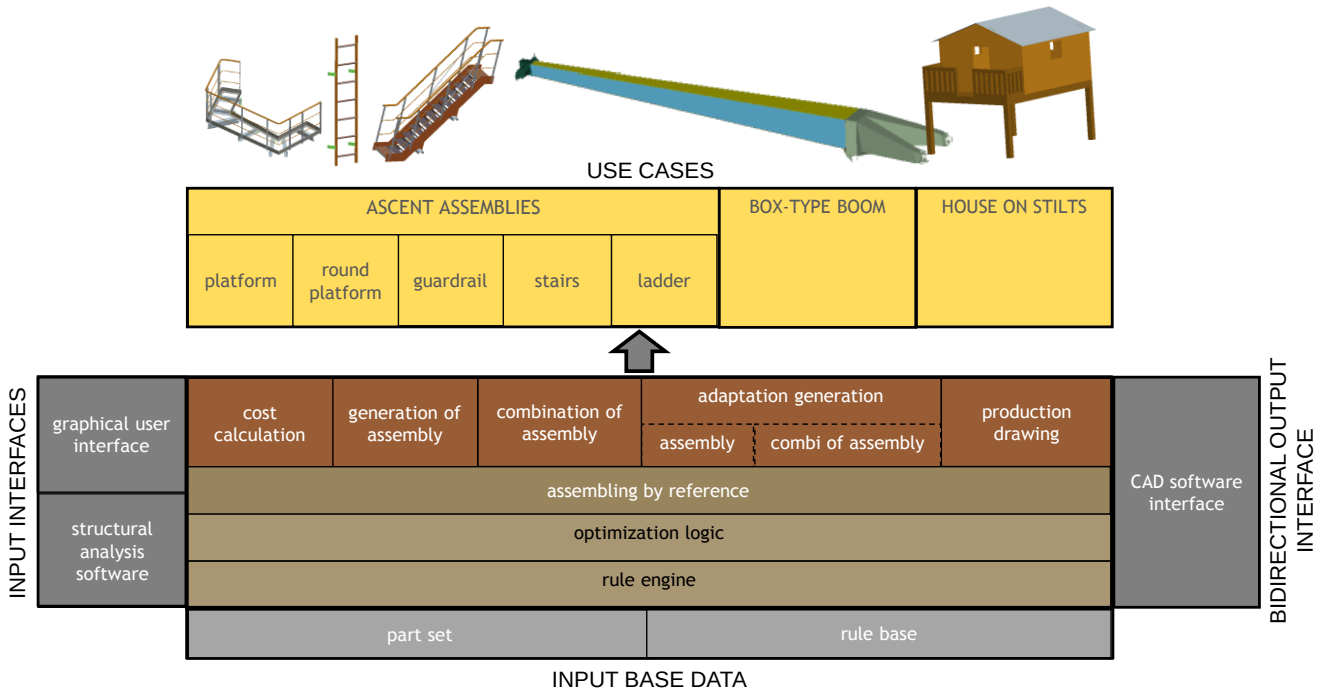


Figure 7. Overview of the framework, for details see Section IX.

affects all parts. For example, if a user manually changes the length of a part, the positions of all dependent parts are adjusted automatically.

The same principle has also been applied to assembly combinations as well as for the adaptation of assemblies. In the latter case, only adapted or new parts are loaded, changed (geometry) and positioned.

VIII. Outpuf of the KBE Application

The KBE application outputs for the chosen assembly or combination of assemblies (in case of the use case of ascent assemblies) the following information:

- CAD model (see Section VII),
- Costs of the assembly/combination of assemblies,
- BOM, and
- Production drawings.

The costs and BOM are directly inferred from the tree generated by the inference engine. To complete the design process, production drawings have to be generated.

The positions of all required views of the product are calculated by an algorithm based on cut optimization, i.e., on efficiently using the space available on the drawing sheets. To ensure a good and fast solution, the concept of cut optimization was simplified. Each view is reduced to a rectangle or a combination of rectangles, which is/are put at the most appropriate and still available position.

After all views have been positioned, all production-relevant dimensions and the according measurements are, after determining the relevant dimensions in the inference engine, automatically added by our generic framework (Section IX). The framework is implemented with the API provided by the CAD system. It is based on a classification of dimension types.

Dimensions can relate to

- an edge,
- an edge to an edge,
- an edge to a point,
- a point to a point, or
- an angle.

For every mentioned type, a dedicated positioning function has been implemented. Finally, the BOM is added to the drawing.

IX. Overview of Developed Framework

The developed framework of the here described software consists of several modules. Fig. 7 visualizes all these components.

The brown boxes in the middle of the image represent the core of the software: the inference engine. The top dark ones consist of the main functionalities of the inference engine (generation, combination and adaption of assemblies) as well as generate the in Section VIII defined output documents. They use all or some of the functionalities provided by the underlying three modules “assembling by reference”, “optimization logic” and “rule engine” (which are also part of the inference engine).

For example, the modules for the generation of assemblies and their combinations use the optimization logic

for calculating a tree representation corresponding to the assembly or assembly combination defined in the GUI. The optimization logic is based on a rule engine that retrieves all relevant rules and other data of the design knowledge (rule base), as well as adequate parts of the standardized part set (Section V.B). Thus, all necessary components, their geometry and their dependencies are calculated. The corresponding tree is established and completed with all necessary information. Then the module "assembling by reference" determines all information for positioning a component in the resulting 3D CAD model in reference to its parent part. Each component of the tree is enhanced by this information. In case of position changes, the advantages of assembling by reference become clear: The modification is handed over to all referenced components, and their positions are updated automatically.

The module "adaptation generation" is applied for the adaptation of already automatically generated CAD models (Subsection VI.C). This module uses functionalities for both calculating a new tree including the changes in the assembly specified by the user, and identifying differences between the new and existing tree.

The component "production drawings" also uses the described functionalities to automatically generate drawings as explained in Section VIII.

The task of the module "cost calculation" is the determination of the manufacturing costs. It uses both the existing tree and the rule engine to get all relevant data for calculation.

For data exchange, the core components of the inference engine use the interfaces the framework provides. There are two types of interfaces, which are prepared for assembly-independent use: Input and output ones (dark grey boxes). The input interfaces (currently GUI and structural analysis software) collect and determine all use case specific external data which are required by the inference engine to proceed with the calculation. The output interface enables the exchange of the tree, containing all necessary data about the assembly or assembly combination, between the inference engine and the specified CAD system to generate the mentioned resulting documents (Section VIII). Furthermore, it transfers all necessary CAD system specific information, particularly the identification number of a part, from the CAD system to the inference engine. This information is required by the inference engine to identify the parts in the CAD system. Thus, the established communication is bidirectional, i.e., from the inference engine to the CAD system and vice versa.

As illustrated in Fig. 7, in the context of projects with LWN, we used the developed framework for the design of the two above discussed use cases (ascent assemblies and box-type booms) as well as for houses on stilts (an illustrative case for an open-house day/event demonstration).

X. THE KBE SYSTEM IN PRACTICE AND ITS BENEFITS

In this section, we demonstrate the practicability of the software and highlight the biggest advantages for the engineers of LWN when using the developed KBE system.

We first illustrate the design and functionality of the KBE system along the example of a platform. We focus here on the steps an engineer has to take to obtain a CAD model, costs, BOM and production drawings. Fig. 8 gives an overview of the general procedure.

As illustrated in the images in the top left box of Fig. 8, the engineer specifies the geometry (e.g., length and angles) and the functionalities (e.g., entry and fixing areas) of the platform. By pressing the "preview" button in the GUI, all data are submitted to the KBE system. In the background, the inference engine starts working: Using the standardized part set and the rule base, the developed algorithm calculates all parameters to generate a 3D CAD model and stores it in a tree. A connection to the CAD system is established and the data is transferred to the CAD system. The design engineer can now follow how each part is loaded and positioned in the CAD model, until the complete assembly is generated. The costs are automatically calculated and shown in the user interface (as illustrated in the left middle box of Fig. 8).

The design engineer then inspects the generated CAD model, and may detect some inaccuracies. An example is shown in the right box of Fig. 8: In Step 1 (current state), the engineer attached the platform to a crane (indicated by the yellow cube) and detected a gap between the crane and the platform. To correct the mistake s/he recalculates the necessary parameters to reach the target state, as illustrated in step 2. These parameters are then re-specified in the GUI (e.g., the length of one side of the platform). When pressing the button "change preview" in the GUI, the inference engine starts the calculation to adapt the previously generated CAD model and costs. By iterating these steps, the design engineer can easily and time-efficiently correct inaccuracies and mistakes in the 3D CAD model and reach a suitable solution to fit the customer's requirements.

Finally, once the CAD model is ready, the engineer only has to press one more button (starting the required methods in the inference engine) to generate the production drawings and BOM (shown in the bottom left box of Fig. 8).

The software is not only adequately designed for its target groups, i.e., design engineers, but also offers flexibility in the design. This means that the standardized part set and the rule base together with the developed application enabled the standardization of the engineering process of nearly all ascent assemblies and box-type booms. In case the generation of an assembly would not be possible using the software, it still offers the possibility to automatically load the required parts of the standardized part set into the CAD system, where the design engineer can then manually generate the assembly.

The major benefit of the developed KBE applications lies in the time savings encountered in the engineering design process, and thus in lower costs and faster time-to-market. By automating the creation of new assemblies and the adaptation of existing ones, the complexity of design processes is well reduced and a significant speed-up is achieved. The engineering of ascent assemblies of an LWN offshore crane used to require up to 150 hours. Employing the here proposed software, these efforts can be cut down to 10 to 20 percent, i.e., to 15 to 30 hours. Also in case of the

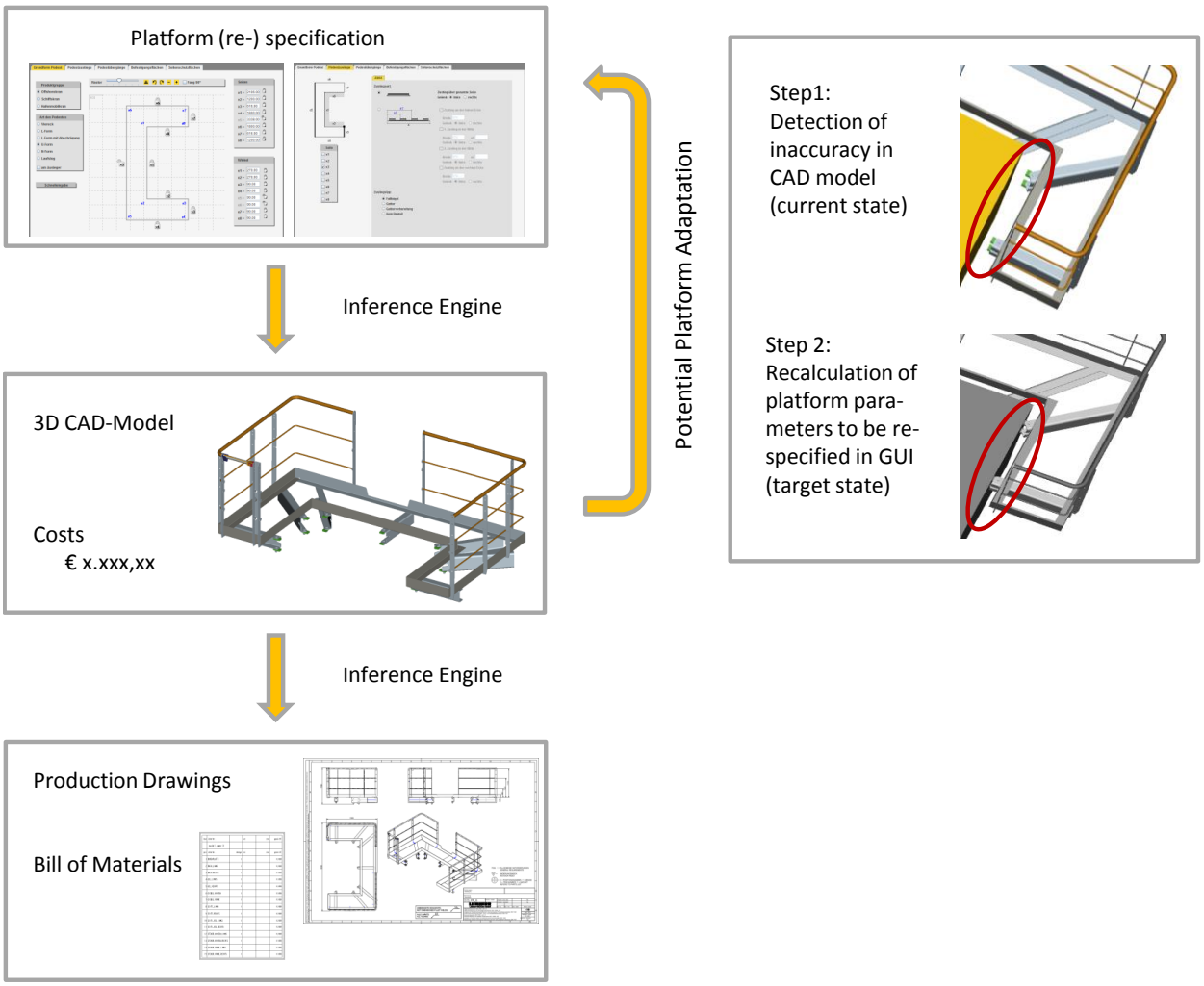


Figure 8. Automatic Platform Design Process, see Section X for details.

box-type booms, a significant speed-up in the development process could be observed, especially due to the integration of the structural analysis software into the KBE application.

XI. Conclusion and Future Work

For companies operating in markets where highly customized products are predominant (and thus requiring a high variety of products, assemblies and parts), it is crucial to find means of reducing design and manufacturing costs. This paper presented a KBE approach to delivering engineering design automation tools, which help solving such challenges, and demonstrated its practicality and efficiency by means of two use cases.

The steps of the KBE system from input to output were thoroughly explained and illustrated with numerous small examples taken from the two use cases. From the beginning, the described application was developed towards a generic framework, which was also described in detail. While the framework was so far mainly used for the presented use cases of ascent assemblies and box-type booms, it is not limited to these tasks. An adaptation and extension towards other assemblies and components, as well as other fields (e.g., construction industry) is currently under way with promising first results.

Consequently, the software framework is designed as a flexible infrastructure towards realizing projects similar to the presented use cases also for other types of complex assemblies and in other industry sectors. Furthermore it is prepared to not only support variant but also adaptive design methodologies and to deliver added-value not only in the design and development phase but all along the process chain of integrated virtual product creation. Overall, a focus was put on keeping the framework as CAD-system independent as possible.

The software framework supports the development of advanced product configurator user interfaces, which are tightly linked to CAD systems via the core component of the framework, i.e., the inference engine. This engine operates on a fully fledged intermediate tree representation for

lossless bidirectional data exchange between framework and CAD systems. Generally, bottom-up and top-down design procedures can be facilitated. This flexibility allows us to go beyond the boundaries of product configuration as the pure application of this paradigm is often too simplistic to cope with the complexities in today's engineering projects. As an example, the generation of new parts was discussed.

The shown user interfaces employ innovative concepts of user guidance supported by interactive sketches and wizards. Modular designs (i.e., standardized components) and both simple and programmed rules as well as constraints are used to formalize the critical knowledge, then automatically ensuring satisfaction of industry-wide and company-internal norms. A diversity of differently complex CAD templates can thereto be imported and used in the implemented approach.

If the design process of an assembly is based on a repetitive logic, it is possible to automate its generation. Furthermore, if a part set exists, which contains all necessary parts of an assembly type, and if the design know-how can be modeled in a rule system, an automatic design from scratch is possible.

The main challenge is to identify these repetitive design processes as well as determine and capture the engineering knowledge hidden behind these processes. By way of operational use of the presented methodology and KBE application in its engineering department, LWN gained valuable insight in the automation of engineering design processes, and further builds on this experience in future application areas.

The main benefit of the KBE applications is the significantly faster realization of the design process. For certain assemblies, this speed-up saved up to 90 percent of the design time. These savings in terms of time and thus also cost were realized with the presented application through the following features:

- Storage of the expert knowledge in a rule-base, together with a standardized part set,
- Reproducibility of all created assemblies,
- Enabling iterative engineering,
- Integration of structural analysis, and
- Production-suitable CAD models (i.e., models, characterized by feasible dimensions, tolerances and adequate material attributes for manufacturing them [38]).

Overall, through the automation of the repetitive part of the design processes of ascent assemblies and box-type booms the engineers of LWN nowadays save a significant amount of design time. This time can be used for creative, value-creating activities instead, such as preparing several design variants for customer specific requirements (e.g., ascent layouts) when bidding for an offer. Furthermore, the faster development of assemblies also allows LWN to react faster to changes in the market, as well as in the customers' requirements. Both of these aspects yield a competitive advantage for LWN.

XII. ACKNOWLEDGEMENTS

This paper discussed results and findings of a research project within the K-Project "Integrated Decision Support Systems for Industrial Processes (ProDSS)", financed through the Austrian funding program COMET (COMpetence centers for Excellent Technologies).

XIII. REFERENCES

[1] G. Frank, "An expert system for design-process automation in a CAD environment," in 8th International Conference on Systems (ICONS), 2013, pp. 179-184.

[2] W. J. C. Verhagen, P. Bermell-Garcia, R. E. C. van Dijk, and R. Curran, "A critical review of knowledge-based engineering: An identification of research challenges," Advanced Engineering Informatics, vol. 26, no. 1, pp. 5-15, Jan. 2012.

[3] Liebherr-Werk Nenzing GmbH. <http://www.liebherr.com/en-GB/35267.wfw> 2014.05.28.

[4] S. Y. Nof, Springer Handbook of Automation. Dordrecht; New York: Springer, 2009.

[5] K. Ehrlenspiel, A. Kiewert, and U. Lindemann, Cost-Efficient Design. Berlin; New York: Springer; ASME Press, 2007.

[6] Committee on Engineering Design Theory and Methodology, Commission on Engineering and Technical Systems, and National Research Council, Improving Engineering Design: Designing for Competitive Advantage. National Academies Press, 1991.

[7] M. M. Andreasen and L. Hein, Integrated Product Development. Bedford, UK: IFS (Publ.), 1987.

[8] D. Jansen, Ed., Handbuch der Electronic-Design-Automation (Handbook of Electronic Design Automation). München; Wien: Hanser, 2001.

[9] D. Macmillen, R. Camposano, D. Hill, and T. W. Williams, "An industrial view of electronic design automation," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 19, no. 12, pp. 1428-1448, 2000.

[10] G. La Rocca, "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design," Advanced Engineering Informatics, vol. 26, no. 2, pp. 159-179, Apr. 2012.

[11] C. B. Chapman and M. Pinfold, "The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure," Advances in Engineering Software, vol. 32, no. 12, pp. 903-912, Dec. 2001.

[12] M. N. Le, Y. S. Ong, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by adapting surrogates," Evolutionary Computation, vol. 21, no. 2, pp. 313-340, May 2013.

[13] Y. Lin, K. Shea, A. Johnson, J. Coultate, and J. Pears, "A method and software tool for automated gearbox synthesis," in 35th Design Automation Conference, Parts A and B, San Diego, California, 2009, vol. 5, pp. 111-121.

[14] A.-C. Zavoianu, G. Bramerdorfer, E. Lughofer, S. Silber, W. Amrhein, and E. P. Klement, "Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives," Engineering Applications of Artificial Intelligence, vol. 28, no. 8, pp. 1787-1794, 2013.

[15] P. Bermell-Garcia, W. J. C. Verhagen, S. Astwood, K. Krishnamurthy, J. L. Johnson, D. Ruiz, G. Scott, and R. Curran, "A framework for management of knowledge-based engineering applications as software services: Enabling personalization and codification," Advanced Engineering Informatics, vol. 26, no. 2, pp. 219-230, Apr. 2012.

[16] G. La Rocca and M. J. L. van Tooren, "Knowledge-based engineering to support aircraft multidisciplinary design and optimization," Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, vol. 224, no. 9, pp. 1041-1055, Sept. 2010.

[17] S. W. G. van der Elst, M. J. L. van Tooren, B. Vermeulen, C. L. Emberey, and N. R. Milton, "Application of a knowledge based design methodology to support fuselage panel design," The Aeronautical Journal, vol. 114, no. 1159, pp. 589-597, 2010.

[18] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design Optimization," Journal of Mechanical Design, vol. 129, no. 4, pp. 370-380, May 2006.

[19] G. Frank and C. Hillbrand, "Automatic support of standardization processes in design models," in 16th International Conference on Intelligent Engineering Systems (INES), 2012, pp. 393-398.

[20] G. Frank, C. Hillbrand, and M. Schwarz, "Simulation based automated design to cost of structurally complex products," in DS 70: Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, 2012, pp. 435-444.

[21] E. K. Antonsson and J. Cagan, Eds., Formal Engineering Design Synthesis. New York, NY, USA: Cambridge University Press, 2001.

[22] J. Clarkson and C. Eckert, Design Process Improvement: A Review of Current Practice. Springer London, 2010.

[23] G. Steinbichler, Methoden und Verfahren zur Optimierung der Bauteilentwicklung für die Spritzgießfertigung (Methods and procedures for optimizing the development of parts for injection-molding production processes). PhD Thesis, University Erlangen-Nuernberg, 2008.

[24] C. T. Leondes, Intelligent Knowledge-Based Systems Business and Technology in the New Millennium. London: Springer, 2004.

[25] C. van der Velden, C. Bil, and X. Xu, "Adaptable methodology for automation application development," Advanced Engineering Informatics, vol. 26, no. 2, pp. 231-250, Apr. 2012.

[26] M. Stokes, Managing Engineering Knowledge: MOKA – Methodology for Knowledge Based Engineering Applications. London: Professional Engineering Pub., 2001.

[27] E.-S. S. Aziz and C. Chassapis, "A decision-making framework model for design and manufacturing of mechanical transmission system development," Engineering with Computers, vol. 21, no. 2, pp. 164-176, Dec. 2005.

[28] R. Raffaeli, M. Mengoni, and M. Germani, "Improving the link between computer-assisted design and configuration tools for the design of mechanical products," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 27, no. 01, pp. 51-64, 2013.

[29] A. Brinkop, Variantenkonstruktion durch Auswertung der Abhängigkeiten zwischen den Konstruktionsbauteilen (Design of Variants by Analyzing the Dependencies among Parts Used in the Design). Sankt Augustin: Infix, 1999.

[30] R. Bourke, "Product configurators: Key enablers for mass customization." Midrange Enterprise, Aug. 2000.

[31] A. Brinkop, "Marktführer Produktkonfiguration (Market leaders of product configuration)," Artikel, Brinkop Consulting, Mai 2013.

[32] D. Sabin and R. Weigel, "Product configuration frameworks – A survey," IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 42-49, July 1998.

[33] S. Danjou, N. Lupa, and P. Koehler, "Approach for automated product modeling using knowledge-based design features," Computer-Aided Design & Applications, vol. 5, no. 5, pp. 622-629, 2008.

[34] N. R. Milton, Knowledge Acquisition in Practice: A Step-by-Step Guide. London: Springer, 2007.

[35] H. Adickes, J. Arnoscht, A. Bong, R. Deger, S. Hieber, R. Krappinger, M. Lenders, P. Post, M. Rauhut, M. Rother, J. Schelling, G. Schuh, and J. Schulz, "Lean Innovation – Auf dem Weg zur Systematik (Lean innovation – Towards a systematic approach)," in AWK Aachener Werkzeugmaschinen Kolloquium '08 - Aachener Perspektiven, Aachen, 2008.

[36] W. Kanok-Nukulchai, "Structural analysis," in Civil-Engineering, K. Horikawa, Ed. Oxford: EOLSS Publishers Co Ltd, 2009.

[37] ANSYS. <http://www.ansys.com> 2014.05.28

[38] H. Brockmeyer, A. Lucko, and F. Mantwil, "Entwicklung von wissensbasierten Assistenzen zur frühzeitigen Produktbeeinflussung am Beispiel des Karosseriebaus (Development of knowledge-based wizards for timely influencing product properties, using the example of body making)," in Die digitale Produktentwicklung, 2008, p. 107 f.

www.iariajournals.org

International Journal On Advances in Intelligent Systems

ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, ENERGY, COLLA, IMMM, INTELLI, SMART, DATA ANALYTICS

issn: 1942-2679

International Journal On Advances in Internet Technology

ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING, MOBILITY, WEB

issn: 1942-2652

International Journal On Advances in Life Sciences

eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO, SOTICS, GLOBAL HEALTH

issn: 1942-2660

International Journal On Advances in Networks and Services

ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION, VEHICULAR, INNOV

issn: 1942-2644

International Journal On Advances in Security

ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

issn: 1942-2636

International Journal On Advances in Software

ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, IMMM, MOBILITY, VEHICULAR, DATA ANALYTICS

issn: 1942-2628

International Journal On Advances in Systems and Measurements

ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL, INFOCOMP

issn: 1942-261x

International Journal On Advances in Telecommunications

AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA, COCORA, PESARO, INNOV

issn: 1942-2601