Report from Dagstuhl Seminar 14062

The Pacemaker Challenge: Developing Certifiable Medical Devices

Edited by

Dominique Méry[1], Bernhard Schätz[2], and Alan Wassyng[3]

1 **LORIA – Nancy, FR**
2 **fortiss GmbH – München, DE**
3 **McMaster University – Hamilton, CA,** wassyng@mcmaster.ca

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14062 "The Pacemaker Challenge: Developing Certifiable Medical Devices". The aim of the seminar was to bring together leading researchers and industrial partners of this field; the seminary ended up with 24 participants from 8 countries: Canada, Denmark, France, The Unites States, Germany, United Kingdom, Brazil. Through a series of presentations, discussions, and working group meetings, the seminar attempted to get a general view of the field of medical devices and certification issues through the pacemaker challenge. The seminar brought together, on the one hand, researchers from the different notations and various tools. The main outcome of the seminar is the exchange of information between different groups and the project of a book.

Seminar February 2–7, 2014 – http://www.dagstuhl.de/14062
1998 ACM Subject Classification D.2 Software Engineering, K.4 Computers and Society
Keywords and phrases Embedded systems, Real-time systems, Medical devices, Model-driven development, Software certification, Validation & verification, Formal methods
Digital Object Identifier 10.4230/DagRep.4.2.17

1 Executive Summary

Dominique Méry
Bernhard Schätz
Alan Wassyng

Pacemakers are typical examples of those medical devices, like insulin pumps, that help save lives when they operate correctly and safely, but may cause grievous harm when they fail. State-of-the art safety standards like IEC 61508 highly recommend (semi-)formal methods for the specification, design, and development of those devices. The Pacemaker Formal Methods Challenge, the first challenge issued by the North American Software Certification Consortium, is hosted by the Software Quality Research Lab at McMaster University, Canada. The challenge is based on a pacemaker specification offered by Boston Scientific, and is part of the verification Grand Challenges which is an international, long-term research programme that seeks to create a substantial and useful body of code that has been verified to the highest standards of rigour and accuracy. The Pacemaker case-study attracted substantial participation during different events in the research community such as workshops at FM2008, FM2009, FHIES 2011, FHIES 2012 and the student competition at ICSE2009 (SCORE).

The Pacemaker Challenge: Developing Certifiable Medical Devices, *Dagstuhl Reports*, Vol. 4, Issue 2, pp. 17–37
Editors: Dominique Méry, Bernhard Schätz, and Alan Wassyng
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Currently there are more than 10 world-class research institutes and universities that take part in the challenge, and are using different approaches. Today, there is a wide range of approaches in the formal methods community to specify and develop high integrity systems. Many of these formal approaches do not work well on industrial level applications, and so the state of the practice is remarkably deficient, even in the case of systems that require certification according to the highest safety levels. The purpose of this five days seminar was to bring together researchers, regulators, as well as practitioners in the medical field to discuss and compare different approaches for the development of certifiable medical software, and further the state of practice. Listed below are research topics related to development of medical software to be covered in the seminar:

- Certification: How can formal methods help in the process of certification of embedded medical software? What standards are in current use and in what measure do they cover model based development? How do we address safety, security and privacy now that these implantable devices are equipped with Wi-Fi, Bluetooth and other wireless networking technologies? How do unspecified environmental assumptions affect the final product?
- Model-based Development: How can established methods for model based development help the building of implantable medical devices? What kind of models (e.g. controlled biological process, hardware platform, safety function) are needed for designing and certifying safety critical medical systems?
- Medical-domain specific aspects: What are the most important specific non-functional aspects that need to be considered while developing implantable medical devices? How can biological and medical aspects be integrated in the development process?
- Tooling: What is the current state of the art and practice concerning tools for formal specification that would be useful in the medical device domain?
- Pragmatics: What is the fitness of different methods for transfer into practice? What do we need to do to ensure that the regulators and workforce are adequately informed of methods and tools that are useful/indispensable in this domain?

As major results of this Dagstuhl Seminar, two publications are targeted at all three relevant sectors researchers, regulators and manufacturers.

The first outcome is a comparison of the different approaches to the Pacemaker Challenge, to be available as a Dagstuhl publication. To achieve such a comparison, the organizers have prepared a catalogue of criteria according to which the approaches are compared. This catalogue was available in advance of the seminar, so presenters can provide a rationale for their classification according to the catalogue, and participants can discuss those classifications.

As a further, more formal result, a joint publication most preferably in the form of a book on the use of rigorous methods for the development of software-intensive medical devices with the pacemaker as a common example will be produced, with the organizers and editors, and all invited research groups as co-authors. Commitment to the participation in this publication will be made a prerequisite for participation in the seminar for members of the research groups having participated in the challenge.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 Validation via Haskell

Andrew Butterfield (Trinity College Dublin, IE)

Joint work of Butterfield, Andrew; O'Cathain, Art
URL https://bitbucket.org/andrewbutterfield/hpacemaker

Validating a complex formal model, in order to show that it captures the intent of the original requirements author, is very important, but can be very difficult. The difficulty is often increased by the obscure nature of the formalism used. Here we discuss an approach to aid validation that involves generating output from the formal model that mimics the presentation used in the original informal documents. We discuss past work in Flash Memory modelling, and then discuss using Haskell as a modelling language applied to the PACEMAKER challenge. We also present results of addressing a challenge made at the seminar to find a bug in the Pacemaker Specification.

3.2 Innovation and Quality Management

Martin Daumer (Trium Analysis Online GmbH, DE)

URL http://www.actibelt.com
URL http://www.thehumanmotioninstitute.org
URL http://www.trium.de

Innovation and quality management should ideally positively influence each other. but there are cases where they seem to pull in opposite directions, in particular in highly regulated fields like medical device development and clinical research. The major effort to explore, validate, document and integrate innovations or even just improvements in existing products may be used as an argument to continue with the status quo, in particular in price sensitive areas. The effort related to the maintainance and continued development of QM systems, the preparation and management of audits is considerable. Important questions are: are currently used QM systems and audit procedures improving the quality of the core processes? What is the evidence that the quality and safety of the products is increasing? What are the best procedures and structures to allow for a partnership between innovation and quality management? We describe and discuss case studies and solutions from ongoing developments related to the introduction of mobile accelerometry in clinical trials and clinical practice.

3.3 Design Space Exploration through Co-modelling and Co-Simulation: the Pacemaker Challenge

John S. Fitzgerald (Newcastle University, GB)

We have presented a study demonstrating that collaborative modelling and co-simulation can be used to explore design alternatives in the context of the pacemaker challenge problem.

Specifically, we showed the use of The Vienna Development Method (VDM) as a discrete-event formalism modelling the controller, coupled to a continuous-time model of the leads and heart environment represented in 20-sim. The modelling technology is formal but readily accessible, using pre-existing formal notations that each have their own simulator support. Our co-simulation tool links the two simulators, implementing a reconciled operational semantics, managing the passage of data, events, and the sometimes delicate progression of time between the two sides. Previous work on modelling the pacemaker controller in VDM concentrated only on the discrete-event model of behaviour, and on the generation of simulation traces using a coarse environment model. Our work reported here provided a stronger environment model built up from primitives. The possibilities for the exploration of design alternatives through co-simulation were illustrated by examining the requirement to change from synchronous to asynchronous pacing modes in the presence of noise (e.g. DDD to DOO, and AAI to AOO). A 20-sim heart model was constructed that allowed the modelling of alternative lead placements, and on the VDM side we modelled the normal controller operation, plus the mode change on noise detection. A feature of this approach is the ability in one model to observe effects of cyber or physical design decisions on system behaviour as a whole. For example, we have only examined one noise detection solution, largely in software and hence modelled in the discrete-event formalism. However, other methods such as filtering (modelled in the continuous time notation) can equally well be explored in exactly the same framework.

3.4 Formal Safety Analysis and Verification of a Family of Cardiac Pacemakers Using SCADE

Michaela Huhn (TU Clausthal, DE)

Joint work of Huhn, Michaela; Sara Bessling
Main reference M. Huhn, S. Bessling, "Enhancing product line development by safety requirements and verification," in Proc. of the 2nd Int'l Symp. on Foundations of Health Information Engineering and Systems (FHIES'12), LNCS, Vol. 7789, pp. 37–54, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-39088-3_3

We investigate the feature-oriented model-based development of a family of pacemakers in a formally founded modeling framework that fosters formal verification. We show how to formalize the findings of a safety analysis uniformly. Then we employ model checking for safety assurance and prove a number of functional properties to hold on the individual pacemaker variants. We extend the SCADE development framework, a tool suite for safety-critical systems based on a formally founded synchronous modeling language, by a transformational approach to product line design: As features are the main concept of functional decomposition in the product line approach, features also direct the safety analysis and the specification of system-level safety requirements: Hence, safety (design) constraints are allocated to features. VIATRA is employed to implement product resolution, i.e. the model graph transformation generating the individual products according the selected feature set. The behavior of components assocoated with a feature is formally modeled inScade language. System Theoretic Process Analysis (STPA) by N. Leveson is applied for hazard analysis. The findings, possible deviations and faults, but also the derived safety constraints are added to the models. Then formal verification techniques are employed in order to prove that the safety constraints are satisfied and thesystem level hazards are prevented. Functional safety is proven using SCADE Design Verifier. The combination of feature-oriented decomposition and STPA leads to a fine-grained safety analysis and is capable of uncovering unwanted interactions. The case study shows that formal methods and tools are ready for use within the software development of medium sized real-world dependable products.

3.5 Model-based Design of Pacemaker Software with Closed-loop Evaluation

Zhihao Jiang (University of Pennsylvania, US)

Joint work of Jiang, Zhihao; Pajic, Miroslav; Alur, Rajeev; Mangharam, Rahul
Main reference Z. Jiang, M. Pajic, R. Alur, R. Mangharam, "Closed-loop verification of medical devices with model abstraction and refinement," International Journal on Software Tools for Technology Transfer, Volume 16, Issue 2, pp. 191–213, April 2014.
URL http://dx.doi.org/10.1007/s10009-013-0289-7

Increasing complexity of the pacemaker software leads to increasing number of potential safety issues. The state-of-the-art pacemaker evaluation is based on open-loop testing which is not able to capture all the closed-loop behaviors. Furthermore, there is no formal techniques used during the development process to maintain the traceability of the requirements. To address these problems, we developed a model-based design framework for pacemaker software which can translate a verified pacemaker model to verified pacemaker implementation. A heart model is designed at different development stages (verification ⇒ simulation ⇒ testing) to evaluate the pacemaker safety/efficacy in closed-loop. The framework improves the confidence of the safety/efficacy of a pacemaker design which reduces the design efforts and increases the speed of certification. For more info please visit our website: http://medcps.org

3.6 Towards product-based certification of medical devices

Soeren Kemmann (Fraunhofer IESE – Kaiserslautern, DE)

The certification of medical devices is currently done by checking the compliance of the manufacturer to international safety standards. Those safety standards however do not focus on the medical device as a product and the irinherent product qualities, but are focused on prescribing rigorous processes. There is however (to the best knowledge of the author) no evidence that good processes lead to good products. Another issue regarding these process based standards is that sometimes innovations or innovative products to not fit into the prescribed process and its therefore hard to certify them. The usual case isthat first the standard is adapted/renewed and afterwards those products can be put on the market. This hinders to a large extend innovations. Another philosophy is that one should focus on the product and make the product safe, This of course requires argumentations and evidences of the product or itsdevelopment to show that the desired system property, safety, is fulfilled. There are already approaches addressing this, such as assurance cases, but they lack of guidelines and support for the manufacturer. We therefore propose an approach focused on arguing product qualities, butincluding guidance for the manufacturer how to develop those.

3.7 The Meaning of PACEMAKER: Formal Semantics for Chapter 5 of PACEMAKER System Specification with BLESS

Brian Larson (Kansas State University, US)

Joint work of Larson, Brian; Hatcliff, John; Chalin, Patrice
Main reference BLESS: Formal Specification and Verification of Behaviors for Embedded Systems with Software, Brian R. Larson, Patrice Chalin, and John Hatcliff, NASA Formal Methods, 2013.
URL https://bless.santoslab.org/

During the editing process at Boston Scientific to transform a company-confidential system specification into the publicly-released *PACEMAKER System Specification* this contributor sought temporal logics that could define pacemaker timing. Having surveyed dozens of temporal logics, finding none suitable for the pacemaker timing of PACEMAKER, this contributor tried extending first-order predicate calculus with simple temporal operators. All features of PACEMAKER were described declaratively using Assertions; the document text was edited to be transliterations of the Assertions into natural language. A very simple extension of first-order predicate calculus was found to be sufficient to define all pacing behavior in PACEMAKER:@ fixes the moment when its predicate is evaluated.[1]

For predicate q and time s, $(q@s \equiv qs)$ The fundamental safety property of PACEMAKER is Lower Rate Limit (LRL).

```
<<LRL:x: --Lower Rate Limit exists t:T --there was a moment in x-l..x
--within the previous LRL interval that (n@t or p@t) >>
--with a pace or non-refractory sense
```

defines a predicate `LRL` applied to parameter `x`, where `n` and `p` are names of ports, and that`n@t` means and event occurred on port `n` at time`t` and `p@t` means and event occurred on port `p`at time `t`. The most recent heartbeat occurred within the last LRL interval can be expressed as `<<LRL(now)>>`, where `now`refers to the present instant.*All* functions in Chapter 5 of PACEMAKER were defined using this temporallogic, and individually compared with the natural language text. No other temporal logic known to seminar participants could similarly capture the semantics of PACEMAKER.

3.8 The Pacemaker Challenge Hardware 2.0

Mark Lawford (McMaster University – Hamilton, CA)

Joint work of Lawford, Mark; Larson, Brian; Wassyng, Alan

We provide some background on the Pacemaker Challenge Hardware reference platform and make the case that it is important for people working on Certification to take into consideration the complete system, including hardware. To date however, Formal Methods researchers have tended to "cherry pick" the problems that show their method in the best light, typically at the requirements level, while embedded systems students have focused on working code that lacks formal specifications. We describe some of the limitation with the

[1] added `q^i` for periodic threads post Spec

current PIC18 based platform that have led to the current situation, namely that it used an RS-232 serial interface and separate power supply rather than a USB connection, required a separate device programmer, had limited open source C compiler support, and required a significant amount of low level driver code to make the system work.We discuss how these limitations can be addressed by redesigning the hardware. Alternatives hardware platforms are analysed and then details on a proposed new Pacemaker Challenge Hardware Reference platform are provided.

3.9 Architecture Centric Modeling – Models of Views, Refinement and Integration

Zhiming Liu (Birmingham City University, GB)

Joint work of Liu, Zhiming; He, Jifeng; Li, Xiaoshan; Stolz, Volker; Zhan, Naijun; Dong, Ruzhen; Ke, Wei; Faber, Johannes
Main reference R. Dong, J. Faber, W. Ke, Z. Liu, "rCOS: Defining Meanings of Component-Based Software Architectures," in Proc. of the International Training School on Software Engineering held at ICTAC 2013, Advanced Lectures on "Unifying Theories of Programming and Formal Engineering Methods," LNCS, Vol. 8050, pp. 1-66, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-39721-9_1

Engineering a complex application software system requires models of different aspects of the system architecture, and different view points of different users. These applications include such as web applications for the cloud, internet of things and cyber-physical systems (CPS). Our presentation at the seminar discusses how models for separation of concerns, refinement, and integration are treated in the rigorous model driven development method. We argue for the need of a unified semantic theory that to enable the consistency use of different logics, techniques and tools for requirements analysis, design, verification and validation. The theory and techniques support the development of tools and methods of software design too. In relation to the case study of the Pacemaker Challenge, we discuss differences between system requirements and software requirements. We in particular show how models system architecture in terms of components and their interfaces can be used to formulate Parnas' Four Variable Model to define the boundary of environment and the software program. The refinement relation is then applied to develop the models of the environment, i.e. the heart, and the Pacemaker control program at different levels of abstraction. We also used the case study to demonstrate the need of a unified semantic theory to support the use of different logics is prosed, such Duration Calculus (DC), LTL with bounded temporal operators, and clocked based logics used with timed automata. The proposed approach is based on our ongoing research of the rCOS Model-Driven Method and the transformational approach to design and verification of real-time and fault-tolerant systems.

3.10 The Pacemaker Challenge – Criteria Catalogue – Development of Medical Device Software System using Event B

Dominique Méry (LORIA – Nancy, FR)

Joint work of Méry, Dominique; Singh, Neeraj Kumar
URL http://eb2all.loria.fr

We summarize our contribution to the Pacemaker Challenge including a list of references of our results and works. Our goal is to integrate the use of formal techniques for designing the software for medical systems. This integrationmay help to identify the possible flaws in the existing system, increasing the quality, and provide some safety assurances to certification standards considering verification and validation approach. We have both academia and practitioners as the target readers. During our work, we met some industrial people (Sorin, Paris); they were looking for some solutions, which exactly match to our developed solution. On the other hand, our developed solutions can also help to students to understand the development of complex systems, like pacemaker. With respect to the technology readiness level of our approach, it may satisfy the TRL4 level (http://www.lbl.gov/dir/assets/docs/TRL%20guide.pdf, page 9).

We spent two yearsto obtain the solution (Event B models) including tools development. In our work, we consider the operating modes of a cardiac pacemaker. Particularly, we verify the pacing and sensing behaviour of cardiac pacemaker including *hysteresis*, and *rate modulation*. We consider the software and the interaction with the physicians. The environment has been partly modelled. We have modelled the biological environment (the heart) for a cardiac pacemaker or ICD. The heart model is based on electrocardiography analysis, which models the heart system at the cellular level. The main objective of this heart model is to provide a biological environment (the heart) for formalizing a closed-loop system (a combined model of a cardiac pacemaker and the heart) to verify the system requirements at an early stage of the system development. Event-B modelling language provides the classical techniques for verifying formal models using refinement mechanism, which is a very powerful feature for developing a formal model progressively. Moreover, Event-B is supported by a very powerful platform Rodin providing a proof obligations generator, editor of models, automatic provers, interactive proof assistant and a tool for animating and model checking models. The Event-B formalism is based on a set-theoretical predicate calculus and structures for organising and expressing models namely contexts for stating static properties over data and machines for expressing transitions or actions called events over state variables. The formalism is rigorous, and it allows to express the safety properties and invariant properties. It is a formalism that supports the correct-by-construction paradigm using the refinement or simulation as a mechanism for structuring the development. The notation is simple but the development of models requires both skills in modelling and in proving. Discrete maths are very difficult notions but Event B is as simple as techniques for developing models in fluid mechanics for instance. These engineers in fluid mechanics are trained in maths and physics and we think that Event B is accessible as long as you play with abstractions. The developed Event B models have been verified and validated using tools of the RODIN platform. Pacing and sensing behaviours of each operating modes under the specified time intervals including features like hysteresis and rate modulation. We ensure safety and invariance properties by construction. We have used both model checker and theorem prover (Rodin) in our work. Engineers can play with the animator and the model checker. They can also discharge many proof obligations as long as they are experts in modelling. The validation is operated through the use of a plugin relating the animation of the model and a view of the heart. It is called areal-time validation in our approach. For communicating with the physicians and medical experts, we have used Flash animation and programming to animate the proved formal model of Event-B. Moreover, we have also implemented the cardiac operating modes in SCJ (Safety Critical Java), and ECG signal interface in Java for developing the simulation. Synthetic ECG signal is generated for simulation purpose. The pacemaker challenge is a very critical application for illustrating and for improving scientific results and especially the discovery of new waysto structure models in Event B. It helps to state questions relating formal techniques and clinical questions. We got many interactions with scientific and industrial partners related to this topics. It is probably a very good illustration that we are addressing real societal questions even with ethical issues.

References

1 Dominique Méry and Neeraj Kumar Singh. Formalization of Heart Models Based on the Conduction of Electrical Impulses and Cellular Automata. In *Foundations of Health Informatics Engineering and Systems.*

2 Dominique Méry and Neeraj Kumar Singh. Medical Protocol Diagnosis Using Formal Methods.

3 Dominique Méry and Neeraj Kumar Singh. Pacemaker's Functional Behaviors in Event-B. Research report, MOSEL – INRIA Lorraine – LORIA, 2009.

4 Dominique Méry and Neeraj Kumar Singh. EB2C : A Tool for Event-B to C Conversion Support, September 2010. Poster and Tool Demo submission, and published in a CNR Technical Report.

5 Dominique Méry and Neeraj Kumar Singh. Functional Behavior of a Cardiac Pacing System. *International Journal of Discrete Event Control Systems (IJDECS)*, December 2010.

6 Dominique Méry and Neeraj Kumar Singh. Real-Time Animation for Formal Specification. In Marc Aiguier, Francis Bretaudeau, and Daniel Krob, editors, *Complex Systems Design & Management 2010*, pp. 49–60, Paris, France, October 2010. Springer.

7 Dominique Méry and Neeraj Kumar Singh. Technical Report on Formal Development of Two-Electrode Cardiac Pacing System. Research report, MOSEL – LORIA, February 2010.

8 Dominique Méry and Neeraj Kumar Singh. Trustable Formal Specification for Software Certification. In T. Margaria and B. Ste, editors, *4th International Symposium On Leveraging Applications of Formal Methods – ISOLA 2010*, volume 6416 of *Lecture Notes in Computer Science*, pp. 312–326, Heraklion, Crete, Greece, October 2010. Springer.

9 Dominique Méry and Neeraj Kumar Singh. A generic framework: from modeling to code. *Innovations in Systems and Software Engineering (ISSE)*, pp. 1–9, September 2011.

10 Dominique Méry and Neeraj Kumar Singh. Automatic Code Generation from Event-B Models. In *SoICT 2011*, Hanoi, Viet Nam, October 2011. Hanoi University, ACM ICPS.

11 Dominique Méry and Neeraj Kumar Singh. EB2J : Code Generation from Event-B to Java. In *SBMF – Brazilian Symposium on Formal Methods*, São Paulo, Brazil, September 2011. CBSoft – Brazilian Conference on Software: Theory and Practice.

12 Dominique Méry and Neeraj Kumar Singh. Formal Development and Automatic Code Generation : Cardiac Pacemaker. In *International Conference on Computers and Advanced Technology in Education (ICCATE, 2011)*, Beijing, China, December 2011.

13 Dominique Méry and Neeraj Kumar Singh. Medical Protocol Diagnosis using Formal Methods. In Zhiming Liu and Alan Wassyng, editors, *International Symposium on Foundations of Health Information Engineering and Systems (FHIES, 2011)*, Johannesburg, South Africa, August 2011.

14 Dominique Méry and Neeraj Kumar Singh. Technical Report on Formalisation of the Heart using Analysis of Conduction Time and Velocity of the Electrocardiography and Cellular-Automata. Technical report, MOSEL – LORIA, August 2011.

15 Dominique Méry and Neeraj Kumar Singh. Technical Report on Interpretation of the Electrocardiogram (ECG) Signal using Formal Methods. Technical report, MOSEL – LORIA, 2011.

16 Dominique Méry and Neeraj Kumar Singh. Critical systems development methodology using formal techniques. In *3rd International Symposium on Information and Communication Technology – SoICT 2012*, pp. 3–12, Ha Long, Viet Nam, August 2012. ACM.

17 Dominique Méry and Neeraj Kumar Singh. Event B. In Jean-Louis Boulanger, editor, *Mise en oeuvre de la méthode B*, Informatique et Systèmes d'Informations. HERMES, April 2013.

18 Dominique Méry and Neeraj Kumar Singh. Formal Specification of Medical Systems by Proof-Based Refinement. *ACM Trans. in Embedded Computing Systems*, 12(1):15, Jan. 2013.

19 Dominique Méry and Neeraj Kumar Singh. Ideal Mode Selection of a Cardiac Pacing System. In Vincent G. Duffy, editor, *4th Int'l Conf. – Digital Human Modeling and applications in Health, Safety, Ergonomics and Risk Management – DHM 2013 (HCI Int'l 2013)*, vol. 8025 of *LNCS*, pp. 258–267, Las Vegas, United States, July 2013. Springer.

20 Neeraj Kumar Singh. *Using Event-B for Critical Device Software Systems.* Springer, 2013.

3.11 Modeling Pacemaker with mbeddr

Zaur Molotnikov (fortiss GmbH, DE)

Joint work of Zaur Molotnikov; Daniel Ratiu; Markus Völter; Thomas Böhm
URL http://mbeddr.fortiss.org/

We have presented preliminary results demonstrating the use of language engineering techniques applied to the Pacemaker Challenge. For this we used a combination of mbeddr and CBMC, free and open-source technologies. Our is a part of mbeddr.

We have modelled and functionally verified on the C code level two pacing modes: VVI and DDD. It turns out, that using language engineering technologies and code level verification it is possible to create verified subsystems. The resulting artifact, C code, is all: executable, functionally verified, lightweight, deployable. We validate the latter two by deploying the DDD pacing logics to Adruino platform, and performing testing, which shows adequate performance and behavior.

The future work is to be done, however, before the method can be applied in practice. The verification is to be characterized, as code transformation and CBMC might introduce problems in it. The model-checking-based process is made more applicable with language-engineering, but additional work is still to be done to ensure scalability to bigger systems/more subsystems of a pacemaker.

References

1 E. Clarke, D. Kroening, and F. Lerda. *A tool for checking ANSI-C programs.* In Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2004), volume 2988 of Lecture Notes in Computer Science, pp. 168–176. Springer, 2004.

2 D. Ratiu, M. Völter, B. Kolb, and B. Schätz. *Using Language Engineering to Lift Languages and Analyses at the Domain Level.* In 5th International Symposium, NASA Formal Methods, 2013.

3 S. Q. R. L. Boston Scientific. *PACEMAKER System Specification* http://sqrl.mcmaster.ca/pacemaker.htm, 2007.

4 M. Voelter. *Language and IDE Development, Modularization and Composition with MPS.* In Generative & Transformational Techniques in Software Engineering, GTTSE 2011, LNCS. Springer, 2011.

5 M. Voelter, S. Benz, C. Dietrich, B. Engelmann, M. Helander, L. Kats, E. Visser, and G. Wachsmuth. *DSL Engineering – Designing, Implementing and Using Domain-Specific Languages.* CreateSpace Publishing Platform, 2013.

6 M. Voelter, D. Ratiu, B. Kolb, and B. Schätz. *mbeddr: Instantiating a Language Workbench in the Embedded Software Domain.* Journal of Automated Software Engineering, 2013.

7 M. Voelter, D. Ratiu, B. Schätz, and B. Kolb. *mbeddr: an extensible C-based programming language and IDE for embedded systems.* In Proc. of Conference on Systems, Programming, and Applications: Software for Humanity, SPLASH '12, 2012.

3.12 Model-Based Design and Medical Devices

Pieter J. Mosterman (The MathWorks Inc. – Natick, US)

Joint work of Mosterman, Pieter J.; Zander, Justyna; Denckla, Ben; Hamon, Gregoire; Ghidella, Jason; Prabhu; Erkkinen, Tom; Schieferdecker, Ina

Main reference P. J. Mosterman, J. Zander, G. Hamon, B. Denckla, "A computational model of time for stiff hybrid systems applied to control synthesis," Control Engineering Practice, Vol. 20, Issue 1, pp. 2–13, January 2012.

URL http://dx.doi.org/10.1016/j.conengprac.2011.04.013

Model-Based Design has been successfully used in Aerospace and Automotive domains. Recently it is gaining interest from the medical devices community. Based on a high-level design process, this presentation illustrates where computational models and tools help improve design and test. Furthermore, it is argued that using computational models as deliverables between design stages requires formalizing the computational semantics of numerical algorithms in an execution engine.

References

1 Jason Ghidella and Pieter J. Mosterman. Requirements-Based Testing in Aircraft Control Design. In *AIAA Modeling and Simulations Technologies Conference and Exhibit 2005*, Paper ID AIAA 2005-5886, San Francisco, CA, August 2005.

2 Pieter J. Mosterman, Jason Ghidella, and Jon Friedman. Model-Based Design for System Integration. In *Proceedings of the 2nd CDEN Int'l Conf. on Design Education, Innovation, and Practice*, pages TB-3-1 through TB-3-10, Kananaskis, Canada, July 2005.

3 Pieter J. Mosterman, Sameer Prabhu, and Tom Erkkinen. An Industrial Embedded Control System Design Process. In *Proceedings of The Inaugural CDEN Design Conference*, pages 02B6-1 through 02B6-11, Montreal, Canada, September 2004.

4 Pieter J. Mosterman and Justyna Zander. Advancing model-based design by modeling approximations of computational semantics. In *Proceedings of the 4th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pp. 3–7, Zürich, Switzerland, September 2011.

5 Pieter J. Mosterman, Justyna Zander, Gregoire Hamon, and Ben Denckla. Towards Computational Hybrid System Semantics for Time-Based Block Diagrams. *Proc. of the 3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, pp. 376–385, Zaragoza, Spain, 2009.

6 Pieter J. Mosterman, Justyna Zander, Gregoire Hamon, and Ben Denckla. A computational model of time for stiff hybrid systems applied to control synthesis. *Control Engineering Practice*, 20(1):2–13, 2012.

7 Gabriela Nicolescu and Pieter J. Mosterman, editors. *Model-Based Design for Embedded Systems.* Computational Analysis, Synthesis, and Design of Dynamic Systems. CRC Press, Boca Raton, FL, 2009.

8 Justyna Zander, Ina Schieferdecker, and Pieter J. Mosterman, editors. *Model-Based Testing for Embedded Systems.* Computational Analysis, Synthesis, and Design of Dynamic Systems. CRC Press, Boca Raton, FL, 2011.

9 Justyna Zander, Pieter J. Mosterman, Gregoire Hamon, and Ben Denckla. On the structure of time in computational semantics of a variable-step solver for hybrid behavior analysis. *Proceedings of the 18th World Congress of the International Federation of Automatic Control*, pp. 9581-9586, Milan, Italy, 2011.

3.13 The Pacemaker Grand Challenge – From Specification to Hardware

Marcel Oliveira (Federal University of Rio Grande do Norte, BR)

Main reference A. O. Gomes, M. V. M. Oliveira, "Formal Development of a Cardiac Pacemaker: From Specification to Code," in Proc. of the 13th Brazilian Symposium on Formal Methods (SBMF'10), LNCS, Vol. 6527, pp. 210–225, Springer, 2010.
URL http://dx.doi.org/10.1007/978-3-642-19829-8_14

As a contribution to the International Grand Challenge project on Verified Software, which aimed to stimulate the creation of newtheories and tools to be applied on industrial-scale problems, we presented a formal model of the pulse generator (PG) of a cardiac pacemaker using the Z notation. Later on, we translated this specification into the Perfect Developer language, from which we automatically generated C# executable code. More recently, we have targeted the Arduino Micro Controller Board. For that, using Perfect Developer, we automatically generated C++ executable code, with which, unfortunately, Arduino could not cope. With some changes in the generated code, we were able to execute the PG in the Arduino board. This execution, however, presented further problems. This talk seeks to promote a discussion on the validity of the approach as well as on the possible solution to overcome our current problems with the running prototype.

3.14 Testing and Operational Evidence of Safety-Critical Software: when is it enough for Certification?

Francesca Saglietti (Universität Erlangen-Nürnberg, DE)

Joint work of Saglietti, Francesca; Lill, Raimar; Meitner, Matthias; Oster, Norbert; Pinte, Florin; Söhnlein, Sven

The talk addressed the importance of explicit test coverage demands for safety-critical software, where the term "coverage" can be taken to refer to the degree to which the behavioral multiplicity or the usage profile are captured during testing. The presentation started by highlighting potential limitations of software verification and validation processes relying on purely formal techniques and stressed the importance of complementing such approaches by extensive and measurable testing and operational evidence. After a comparison of coverage demands posed by different safety standards, the talk focused on research work supporting structural testing by automatizing as far as possible the underlying test data generation process. The use of genetic algorithms revealed to offer useful heuristics for the solution of multi-objective optimization problems involving both the maximization of testing coverage and the minimization of testing effort. Different testing environments were reported to have been automatically optimized in the light of these conflicting targets: among them are the integration testing of synchronously interacting software components as well as the interoperation testing of autonomously cooperating robotic entities.The final part of the talk was devoted to the quantitative evaluation of operational evidence gained with proven-in-use software. In a real-world automotive application involving a software-based gearbox controller, statistical sampling theory revealed to provide a practicable instrument for the extraction of conservative reliability estimates. In order to avoid functionally incomplete operational experience, structural and reliability testing targets were combined into a novel testing strategy aiming at the generation of statistically independent and operationally representative test scenarios capable of covering the data flow induced by component invocations.

3.15 Model-Based Engineering for Medical Device Software

Bernhard Schaetz (fortiss GmbH – München, DE)

Joint work of Schaetz, Bernhard; Ratiu, Daniel; Gareis, Stephanie; Chen, Wenwen

Models can contribute to a high-quality development process of embedded software by

1. providing dedicated and concise views of the systems and environment under consideration
2. enable analysis techniques to front-load quality assurance
3. enable synthesis techniques to automate development steps and support design-space exploration.

To assess the usefulness of the model-based approach for medical software, several techniques recommended in standards for safety-critical software-intensive systems (e.g., IEB 61508) were applied to the pacemaker challenge using the AutoFocus3 (AF3) development approach and tool. AF3 provides dedicated views (e.g., textual requirements, template-based property descriptions, component-based system specification, platform description), strong analysis (e.g., non-determinism analysis, verification of properties), and synthesis techniques (e.g., test case generation, deployment generation). Using these techniques, a complete development process was carried out, starting from the textual requirements based on the Boston Scientific Specification and leading to a running implementation on the MSCert prototyping hardware combined with a implementation of the physical heart model on a PIC18F5420 microprocessor. For all the produced artifacts – requirements specification, pacemaker system and software design, logical context/heart model, pacemaker implementation, physical heart model implementation – and performed steps – requirements structuring and formalization, conformance verification, soundness analysis, MiL and HiL verification, requirements-based testing, conformance testing, code generation and deployment of pacemaker software and heart model – we demonstrate how these are supported by or automated by the AF3 approach and how this contributes to a development process according to IEC 61204.

3.16 Development of Medical Device Software Systems

Neeraj Kumar Singh (McMaster University – Hamilton, CA)

Joint work of Singh, Neeraj Kumar; Méry, Dominique

Formal techniques are not well integrated in the software development life-cycle of medical device software systems. We propose a development life-cycle to develop the medical systems using formal techniques from requirements analysis to code generation. In this context, we have provided a chain of tools support to realize the rigorous process development considering the safety assessment approaches. Moreover, we also address the necessity of an environment model and real-time animator to bridge a gap between the stakeholders. Our approach is to design a system in a progressive fashion using refinements. Each refinement level introduces the new concrete behaviours considering some safety properties to make sure the correctness of desired functional behaviours. To evaluate our proposed life-cycle and associated tools, we use the Grand Challenge cardiac pacemaker.

References

1 Neeraj Kumar Singh *Using Event-B for Critical Device Software Systems*, Springer, ISBN: 978-1-4471-5259-0, I-XVIII, 1–326, 2013.

2 D. Méry and N. K. Singh *Event-B*, Jean-Louis Boulanger. Mise en oeuvre de la méthode B, HERMES, Apr. 2013, Informatique et Systèmes d'Informations, ISBN : 978-2-7462-3810-7.

3 D. Méry and N. K. Singh *A Generic Framework: from Modeling to Code*, Journal of Innovations in Systems and Software Engineering, Springer London, 1–9, 2011.

4 D. Méry and N. K. Singh *Formal Specification of Medical Systems by Proof-Based Refinement*, ACM Trans. on Embedded Computing Systems, Vol-12(1), 15:1–15:25, Jan. 2013.

5 D. Méry and N. K. Singh *Functional behavior of a cardiac pacing system*, International Journal of Discrete Event Control System, Vol-1, 129–149, January 2011.

6 D. Méry and N. K. Singh *Ideal Mode Selection of a Cardiac Pacing System*, 15th Int'l Conf. on Human-Computer Interaction (HCII 2013), Las Vegas, Nevada, USA, 21–26 July 2013.

7 N. K. Singh, Andy Wellings and Ana Cavalcanti *The Cardiac Pacemaker Case Study and its Implementation in Safety-Critical Java and Ravenscar Ada*, Proceedings of the 10th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2012), ACM, 62–71, October 2012.

8 D. Méry and N. K. Singh *Closed-loop modeling of Cardiac Pacemaker and Heart*, International Symposium on Foundations of Health Information Engineering and Systems (FHIES 2012), Springer LNCS, Vol-7789, 151–166, 2013.

9 D. Méry and N. K. Singh *Critical Systems Development Methodology using Formal Techniques*, Proc. of the 2012 Symp. on Information and Communication Technology, SoICT 2012, Hanoi, Vietnam, ACM, ACM International Conference Proceeding Series, 3–12, 2012.

10 D. Méry and N. K. Singh *Automatic Code Generation from Event-B Models*, Proceedings of the 2011 Symposium on Information and Communication Technology, SoICT 2011, Hanoi, Vietnam, ACM, ACM International Conference Proceeding Series, 179–188, October 2011.

11 D. Méry and N. K. Singh *Formal Development and Automatic Code Generation : Cardiac Pacemaker*, International Conference on Computers and Advanced Technology in Education (ICCATE 2011), Beijing, China, 3–4 November 2011 (Appear in a book EICE 2012, ASME Press, New york).

12 D. Méry and N. K. Singh *EB2J : Code Generation from Event-B to Java*, 14th Brazilian Symposium on Formal Methods (SBMF 2011), 26–30 September 2011, (*Short Paper*).

13 D. Méry and N. K. Singh *Formalisation of the Heart based on Conduction of Electrical Impulses and Cellular-Automata*, Int'l Symp. on Foundations of Health Information Engineering and Systems (FHIES 2011), Springer LNCS, Vol-7151, 140–159, 2012.

14 D. Méry and N. K. Singh *Medical Protocol Diagnosis using Formal Methods*, International Symposium on Foundations of Health Information Engineering and Systems (FHIES 2011), Springer LNCS, Vol-7151, 1-20, 2012.

15 D. Méry and N. K. Singh *Trustable Formal Specification for Software Certification*, in Proceeding ISoLA (2), Springer LNCS,Vol-6416, 312–326, 2010.

16 D. Méry and N. K. Singh *Real-time animation for formal specification*, in Proceeding Complex Systems Design & Management, Paris, 27–29 October, 2010.

17 D. Méry and N. K. Singh *EB2C : A Tool for Event-B to C Conversion Support*, Poster and Tool Demo submission, and published in a CNR Technical Report in SEFM 2010.

18 D. Méry and N. K. Singh *Technical Report on Formalisation of the Heart using Analysis of Conduction Time and Velocity of the Electrocardiography and Cellular-Automata*, Technical Report (http://hal.inria.fr/inria-00600339/en/), 2011.

19 D. Méry and N. K. Singh *Technical Report on Interpretation of the Electrocardiogram (ECG) Signal using Formal Methods*, Technical Report (http://hal.inria.fr/inria-00584177/en/), 2011.

20 D. Méry and N. K. Singh *Pacemaker's Functional Behavior in Event-B*, Technical Report (http://hal.inria.fr/inria-00419973/en/), 2009.

21 D. Méry and N. K. Singh *Formal Development of Two-Electrode Cardiac Pacing System* Technical Report, (https://hal.archives-ouvertes.fr/inria-00465061/en/), 2010).

3.17 Grand Challenge Problems for Education

Alan Wassyng (McMaster University – Hamilton, CA)

Joint work of Wassyng, Alan; Lawford, Mark

This talk provides a brief retrospect of the PACEMAKER Grand Challenge, anopinion as to how successful it has been, and how it has been used for teaching at McMaster University. The primary goal of the talk is to suggest that the PACEMAKER Challenge and other such future challenges can be used as influentialcase studies in education.

3.18 Timing Analysis of Pacemakers

Reinhard Wilhelm (Universität des Saarlandes, DE)

Modern high-performance processors introduce a large variability in the execution time of machine instructions. Determining worst-case program execution times (WCETs) is therefore a difficult problem. I present a solution to this problem using static program analysis incorporating an abstract model of the architecture. Such a static analysis of a program computes an invariant at each program point. This invariant describes all execution states that the HW may be in when program execution reaches thisprogram point. The invariant can be used to determine a reliable and precise upper bound on the execution time of the instruction at this program point and on the execution time of the containing basic block. Based on these bound a worst-case path is determined through the basic-block graph of the program. This approach and tools based on it are in routing use in the embedded-systems industry. They have been accepted and used for certification of several time-critical subsystems in modern airplanes. It seems that this technology has not a very high relevance for pacemakers since the real-time requirements are rather modest and the amount of computation is also quite limited.

References

1 Reinhard Wilhelm and Daniel Grund. *Computation takestime, but how much?*. Commun. ACM (CACM) 57(2):94–103 (2014).

4 Working Groups and Panel Discussions

We have reported the summary of a planned book that was emerging during our seminar. Works will go ahead after the seminar. By the end of the seminar, we obtain the following sketch for this book. The summary of the book is resulting from panel discussions and planned working groups attached to each chapter of the book. We are skectchig chapters planned for the book and we warn readers that some chapters are not yet completely defined.

1. Introduction: *what's it about? emphasis on evidence for certification, aimed at regulators, acceptable risk, not "no risk", focus on critical systems (Class III), Common Terminology/Glossary*
2. Pacemaker Challenge: *This section will recall the Pacemaker Challenge and outcomes.*
3. Certification Overview: *certification goal, Safety & Efficacy, Assumptions and problem statements/Objectives and Requirements (everywhere)*
4. Current Practice: *process based, good historical reasons for this, approaches to risk/hazard analysis, no evidence it works*
5. Process and Product: *the argument for this, reify safety/efficacy into product qualities/properties, certification checks EVIDENCE, FMEA – what's best for software?, EVIDENCE, how to cover everything? how to avoid systemic error?*
6. Rigorous Methods for Development of Safety-Critical Systems: *Summary on fomal methods related to certification and medical devices*
7. Producing Product Qualities and Evidence: *confidence vs evidence, benefit of mathematical rigour, v. high confidence in domain of applicability, Evidence Classes vs Development Phases, applicability conditions for evidence classes in each phase, Remember: focus is on CERTIFICATION, processes for devpt for certification vs. processes for certification*
8. Tools for Development of Safety-Critical Systems: *Introduction (not an overview; Types of tools (primarily model-based with simulation, model-checking, theorem proving etc. but also the operating system, compilers etc.); Roles of different involved stakeholders (developer, regulator, end user,) (different stakeholders have different issues, technical, business) (used for development of systems (not for certification authority in itself although requirements traceability may be useful); tool chain considerations (semantically connected tools); tools for producing models for different viewpoints (using separation of concerns and abstraction) of key parts of a system with a specific purpose of analysis is recommended (Model stuff, produce artifacts that can be used to gain confidence in the safety and correctness of the system, engineering reasoning between such dedicated models and the claim produced mush be provided; qualification/validation of tools (including what is done*

in other application domains) independent analysis tools that can increase confidence could be used with advantage even without certification/validated (In particular it is valuable to have multiple independent tools analysing the same aspects in order to increase likelihood of correctness; The state-of-the-art analysis tools here shall be used to incorporate safety and correctness of medical system); Concluding Remarks

9. New Roles for Regulators and Standards: *acceptable risk, not "no risk"; standards should focus on which qualities/properties to assure; state required levels of confidence; competency; certification should check if acceptable confidence; level attained; required knowledge: (i) for regulator; (ii) for developers.*
10. Conclusion

A shared collaborative space is provided to potential contributors among the participants of the seminar.

5 Conclusions

It became obvious during this seminar that challenge problems like the PACEMAKER Challenge are invaluable for stimulating research and furthering the state of that research, facilitating collaborations, providing a focus for application of theory to a specific, practical problem, and for high quality case study material that could be used in education. A few lessons emerged for future such Grand Challenges: 1) A hardware platform that can be used uniformly by all collaborators is a real advantage. Even with the limitations of the PACEMAKER reference hardware, having that platform available (at reasonable cost) contributed to the success in getting so many groups to participate in the Challenge; 2) We need better defined rules for such challenges. That way we will be able to make better comparisons between competing approaches; 3) We need to be able to plan and support workshops and seminars for participants actively engaged in the Challenge (Dagstuhl is an incredibly effective vehicle for this kind of meeting); 4) Special issue publications are necessary to really disseminate the work. The upcoming book arising out of this Dagstuhl Seminar is a good example. 5) There should be a number of target audiences for each Challenge, and publications can target one or a number of those audiences. The fact that the upcoming book from this Seminar targets Medical Device Regulators, is an enterprising move. It has the potential to make a difference in a very practical way.

In conclusion, the presentations at this Dagstuhl Seminar, focused on improving the quality (safety, security and dependability) of medical device software, specifically the Pacemaker, were of very high quality. Even more useful to the community was the in-depth and extensive discussion that took place. The upcoming book is now of immediate concern, and the first milestone for examining drafts and revising direction, if necessary, is due at the end of June 2014.

The organizers wish to thank all the participants for their excellent contributions, and the staff and organizers of Dagstuhl for this wonderful opportunity.

6 Programme

Monday – February 3, 2014

09:00–10:30 Welcome, Overview & Introductions: What we want out of this seminar
10:30–11:00 Coffee
11:00–12:15 Brian Larson – The PACEMAKER Spec
12:15–13:30 Lunch
13:30–14:30 Martin Daumer – Certification affects innovation
14:30–15:30 Gunter Klebes – Quality assurance for certification
15:30–16:00 Coffee
16:00–17:00 Roland Mols answers domain questions
17:00–17:30 Dealing with the medical domain

Tuesday – February 4, 2014

09:00–10:30 Case Study 1: Artur Gomes & Marcel Oliveira
Case Study 2: Dominique Méry & Neeraj Singh
Case Study 3: Zhiaho Jiang
10:30–11:00 Coffee
11:00–12:00 Case Study 4: John Fitzgerald & Peter Gorm Larsen
Case Study 5 Brian Larson
12:00–13:30 Lunch
13:30–15:00 Case Study 6: Markus Völter & Zaur Molotnikov
Case Study 7 Michaela Huhn
Case Study 8 Daniel Ratiu & Bernhard Schätz
15:00–15:45 Discussion on Case Studies
15:45–16:00 Coffee
16:00–16:30 Francesca Saglietti – Testing and Certification
16:30–17:00 Soeren Kemmann – Standards for Medical Software
17:00–17:30 Reinhard Wilhelm – Verification of Non-Functional Aspects

Wednesday – February 5, 2014

09:00–9:30 Pieter Mosterman – Analysis & Design Tools for Medical Devices
09:30–10:00 Andrew Butterfield - Validation
10:00–10:30 Do we need to model the environment?
10:30–11:00 Coffee
11:00–11:30 Mark Lawford – Hardware for Challenge Problems
11:30–12:00 Alan Wassyng – Using Challenge Problems for Teaching
12:00–13:30 Lunch
13:30–21:00 Excursion

Thursday – February 6, 2014

09:00–9:20 Christian Prehofer – medical devices: some observations
09:20–10:30 Elaboration of topics
10:30–11:00 Coffee
11:00–12:00 Prioritization of topics
12:00–13:30 Lunch
13:30–14:30 Declaration of interest: topics & involvement
14:30–15:45 (Q&D) attempt: Topic specific points for elaboration
15:45–16:00 Coffee
16:00–17:30 Discussion on Topic specific points for elaboration

Friday – February 7, 2014

09:00–10:30 Discussion on Topic specific points for elaboration
10:30–11:00 Coffee
11:00–12:00 Summary on the project of book
12:00–13:30 Lunch

Participants

- Andrew Butterfield
Trinity College Dublin, IE
- Martin Daumer
Trium Analysis Online GmbH, DE
- John S. Fitzgerald
Newcastle University, GB
- Michaela Huhn
TU Clausthal, DE
- Zhihao Jiang
University of Pennsylvania, US
- Soeren Kemmann
Fraunhofer IESE – Kaiserslautern, DE
- Günther Klebes
sepp.med – Röttenbach, DE
- John Komp
University of Minnesota – Minneapolis, US
- Peter Gorm Larsen
Aarhus University, DK
- Brian Larson
Kansas State University, US
- Mark Lawford
McMaster Univ. – Hamilton, CA
- Zhiming Liu
Birmingham City University, GB
- Dominique Méry
LORIA – Nancy, FR
- Zaur Molotnikov
fortiss GmbH – München, DE
- Pieter J. Mosterman
The MathWorks Inc. – Natick, US
- Marcel Oliveira
Federal University of Rio Grande do Norte, BR
- Christian Prehofer
fortiss GmbH – München, DE
- Florian Prester
sepp.med – Röttenbach, DE
- Francesca Saglietti
Univ. Erlangen-Nürnberg, DE
- Bernhard Schätz
fortiss GmbH – München, DE
- Neeraj Kumar Singh
McMaster Univ. – Hamilton, CA
- Markus Völter
Völter Ingenieurbüro, DE
- Alan Wassyng
McMaster Univ. – Hamilton, CA
- Reinhard Wilhelm
Universität des Saarlandes, DE