Integrated Tool Chain for Model-based Design of Cyber-Physical Systems: The INTO-CPS Project

Peter Gorm Larsen
Department of Engineering
Aarhus University, Denmark
Email: pgl@eng.au.dk

John Fitzgerald
School of Computing Science
Newcastle University, UK
Email: john.fitzgerald@ncl.ac.uk

Jim Woodcock
Department of Computer Science
University of York, UK
Email: jim.woodcock@york.ac.uk

Peter Fritzson
Department of Computer and Information Science
Linköping University, Sweden
Email: peter.fritzson@liu.se

Jörg Brauer
Verified Systems International
Bremen, Germany
Email: brauer@verified.de

Christian Kleijn
Controllab Products
Enschede, The Netherlands
Email: christian.kleijn@controllab.nl

Thierry Lecomte
Clearsy SAS
Aix en Provence, France
Email: thierry.lecomte@clearsy.com

Markus Pfeil
TWT Science & Innovation
Stuttgart, Germany
Email: markus.pfeil@twt-gmbh.de

Ole Green
Agro Intelligence
Aarhus, Denmark
Email: olg@agrointelli.com

Stylianos Basagiannis
United Technologies Research Centre
Cork, Ireland
Email: BasagiS@utrc.utc.com

Andrey Sadovykh
Softeam
Paris, France
Email: andrey.sadovykh@softeam.fr

***Abstract*—We describe INTO-CPS, a project that aims to realise the goal of integrated tool chains for the collaborative and multidisciplinary engineering of dependable Cyber-Physical Systems (CPSs). Challenges facing model-based CPS engineering are described, focussing on the semantic diversity of models, management of the large space of models and artefacts produced in CPS engineering, and the need to evaluate effectiveness in industrial settings. We outline the approach taken to each of these issues, particularly on the use of semantically integrated multi-models, links to architectural modelling, code generation and testing, and evaluation via industry-led studies. We describe progress on the development of a prototype tool chain from baseline tools, and discuss ongoing challenges and open research questions in this area.**

I. Introduction

Cyber-Physical Systems (CPSs) which closely integrate computing and physical elements have enormous potential to catalyse businesses and improve the quality of life, but present significant engineering challenges [1]. They are characterised by a complex architecture and a design process that necessarily involve diverse technical disciplines, formalisms and even cultures. The CPS engineer faces a large design space that is prohibitively expensive to explore with physical prototypes, while the need for dependability of the CPS as a whole means that there is a need for well-founded validation and verification techniques.

A common workflow for the model-based design of CPS, and the tools needed to support it, are currently missing [2], [3]. It is not surprising, therefore, that current need analyses and research agendas identify, among other things, the challenges of combining diverse modelling paradigms, the extension of collaborative modelling through the life cycle, and the need to provide firm semantic foundations for CPS design methods [4], [5], [6].

The vision of the INTO-CPS[1] consortium is that CPS engineers should be able to deploy a wide range of tools to support model-based design and analysis, rather than relying on a single “factotum”. The goal of our project is to develop an integrated “tool chain” that supports multidisciplinary, collaborative modelling of CPSs from requirements, through design, to realisation in hardware and software, enabling traceability through the development. We will integrate existing industry-strength baseline tools in their application domains, based centrally around Functional Mockup Interface (FMI)-compatible co-simulation [7]. The project focuses on the pragmatic integration of these tools, making extensions in areas where a need has been recognised. The tool chain will be underpinned by well-founded semantic foundations that ensures the results of analysis can be trusted.

The tool chain is intended to provide powerful analysis techniques for CPS models, including generation and static checking of FMI interfaces; model checking; Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) simulation,

[1] See http://into-cps.au.dk/.

supported by code generation. It will allow for both Test Automation (TA) and Design Space Exploration (DSE) of CPSs. The INTO-CPS technologies will be accompanied by method guidelines, lowering entry barriers for CPS development.

In order to validate the effectiveness of such a tool chain it is necessary to evaluate it on genuine and challenging industrial applications. In INTO-CPS, four case studies are carried out by industrial partners in the rail, automotive, agricultural and building automation sectors. These domains are highly diverse, but they all stand to benefit significantly from design techniques that support CPS. In addition, the project has an Industrial Follower Group (IFG) of currently more than 40 members who will be able to provide input on emerging methods and tools, as well as supplying challenge problems that complement the larger industrial case studies.

In this paper we provide an overview and update of INTO-CPS, including its objectives (Section II), the INTO-CPS technology including its baseline tools, technical foundations and and methodology (Section III), and industry case studies (Section IV). Finally Section V provides a summary of current progress and an indication of future plans.

II. Objectives

The project has five specific objectives:

1) **Build an open, well-founded tool chain for multidisciplinary model-based design of CPS that covers the full development life cycle of CPS.** The tool chain will support multiple modelling paradigms and will cover multiple development activities, including requirements modelling, analysis, simulation, validation, verification, and traceability of artefacts throughout all development activities across disciplinary boundaries.
2) **Provide a sound semantic basis for the tool chain.** We will produce mathematical foundations to support CPS multi-modelling and to underpin the tool chain. This will include semantics for FMI co-simulation, as well as SysML, discrete-event and continuous-time paradigms.
3) **Provide practical methods in the form of guidelines and patterns that support the tool chain.** The INTO-CPS methodology will be developed to ensure that adoption of the tool chain is cost-effective, providing industrial users with pragmatic guidance to help them determine the best modelling technologies and patterns to meet their needs.
4) **Demonstrate in an industrial setting the effectiveness of the methods and tools in a variety of application domains.** Four complementary industry case studies have been selected from four distinct domains that currently experience pressure to develop reliable CPSs (automotive, agricultural, railways and building automation). The case studies will be used to drive the production of the tools and methods and evaluate them.
5) **Form an INTO-CPS Association to ensure that project results extend beyond the life of the project.** Membership of the Association will allow future case study owners access to information, training, and competitively priced licenses at various levels of support. Tool vendors will be offered services to help integrate their products into the tool chain.

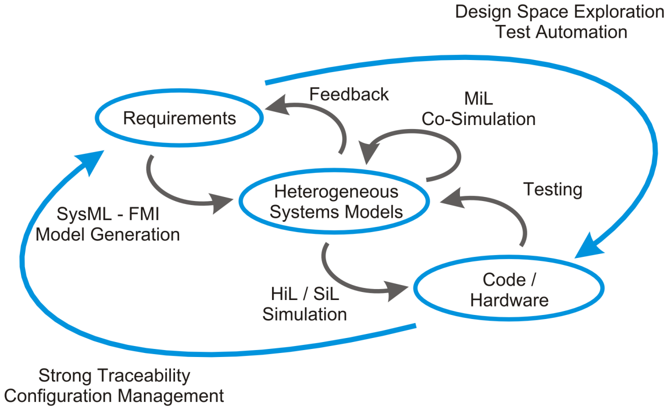

Fig. 1. Connections in the INTO-CPS tool chain

III. The INTO-CPS Technology

A. Workflow in the INTO-CPS Tool Suite

Figure 2 gives a graphic overview of the workflow supported by the tool chain. At the top level, the tool chain will allow requirements to be expressed using different views in SysML, supported by guidelines for capturing the requirements on a CPS. A SysML profile is being developed that allows the architecture of a CPS to be described, including both software, physical and networking elements. From the architectural model, an FMI interface can be generated, along with stub models to reduce effort in producing initial interfaces between constituent models. We export model descriptions for each of the constituent models that then subsequently can be imported by different simulation tools indicating the interfaces that are needed for the corresponding FMUs inspired by the work from HybridSim [8]. In addition the CPS SysML profile defined enable export from SysML about the simulation configuration.

Heterogeneous system models can then be built around this FMI interface, using the stub models as a starting point. A number of industry-strength tools will be connected here, permitting these heterogeneous “multi-models” to contain discrete-event models of software, continuous-time models of physical elements and the networks between them. The tool chain will permit static analysis of these multi-models, including model checking (of appropriate abstractions) and static analysis of the FMI interfaces. The constituent models can either be in the form of Discrete Event (DE) models or in the form of Continuous-Time (CT) models combined in different ways.

A Co-simulation Orchestration Engine (COE) is being developed by combining existing co-simulation solutions and scaling them to the CPS level, allowing these CPS multi-models to be evaluated through co-simulation. The COE will also allow real software and physical elements to participate

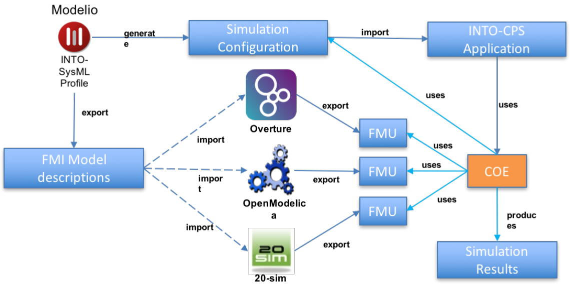

Fig. 2. The current INTO-CPS Tool Chain

in co-simulation alongside models, enabling both HiL and SiL simulation. Code generation from some of the baseline tools will help support automated HiL simulation.

The COE will also allow multiple co-simulations to be defined and executed, and the results collated and presented automatically. The tool chain will allow these multiple co-simulations to be defined via DSE or through TA based on test cases generated from the SysML requirement diagrams and using Linear Temporal Logic (LTL) formulas [9].

Currently a part od the INTO-CPS tool chain has been completed. This is illustrated by Figure 2. In the current tool chain the CPS SysML profile enables export of model descriptions for the constituent models that subsequently can be imported by the different baseline modelling and simulation tools. These can then import these model descriptions and after further work export the corresponding FMUs. In a similar fashion it is possible from the CPS SysML profile to export the overall composition of the constituent CPS components from a co-simulation perspective. This can then be explored by a user of the INTO-CPS Application which in turn makes use of the COE that uses the different FMUs exported for each of the constituent CPS components. Soon this will be extended with the envisaged DSE and TA capabilities as well as the SiL and HiL simulations.

B. Baseline Tools

The following list describes the existing baseline tools that are incorporated in the INTO-CPS tool suite:

- Modelio[2] is an open-source modelling environment supporting industry standards like UML and SysML [10]. INTO-CPS will make use of Modelio for high-level system architecture modelling using the SysML language and proposed extensions for CPS modelling.
- Overture[3] [11] is another open-source tool which supports modelling and analysis in the design of discrete computer-based systems using the VDM-RT notation [12]. This tool was used in the DESTECS[4] project for modelling and simulation of DE controllers [13].
- 20-sim[5] was used in DESTECS as the main tool for modelling and simulation of CT systems [14]. INTO-CPS will expand this use by incorporating results of systems engineering. The code generation and deployment capabilities of 20-sim will be used for HiL testing.
- OpenModelica[6] is an open-source Modelica-based modelling and simulation environment [15]. Modelica is an object-oriented, equation-based language to model complex CPSs. A large number of Modelica model libraries is available.
- Crescendo[7] is the co-simulation tool developed in the DESTECS project [16]. This tool enables the collaborative simulation of a DE controller modelled from the Overture tool, and a CT model of the physical plant from the 20-sim tool. The custom-built co-simulation interface was expanded to support DE models from Matlab/Simulink as well.
- TWT co-sim engine[8] is a framework for configuring and running co-simulations. The individual simulations each run in their own native tool or are supplied as FMUs. The simulations are connected via definition of signals to be exchanged. These signals are passed between the simulations using the co-sim router. Among the currently supported tools are Matlab/Simulink, Modelica (both OpenModelica and Dymola[9]), StarCCM+[10] and Qucs[11].
- RT-Tester[12] is a test automation tool for automatic test generation, test execution, and real-time test evaluation [9]. The RT-Tester Model Based Test Case and Test Data Generator supports model-based testing: automated generation of test cases, test data, and test procedures from UML/SysML models.

C. INTO-CPS Foundations

An integrated tool chain for CPS requires that evidence supplied by the different tools can be reconciled to produce coherent analysis results. Specifically, it must be possible to relate the outputs of the different tools in a way that provides them with unambiguous mathematical meaning. Different analysis tools are based on different notations; for example, a simulator may work at the level of a transition relation described using Structural Operational Semantics [17], whilst a program verifier may use an axiomatic Hoare logic [18]. Though distinguishable, these formalisms are related in that they provide a common foundation giving a global view of the world into which the different tool languages can be mapped and assigned meaning.

[2] http://www.modelio.org/
[3] http://overturetool.org/
[4] http://destecs.org/
[5] http://www.20sim.com/
[6] https://www.openmodelica.org/
[7] http://crescendotool.org/
[8] http://www.twt-gmbh.de/produkte/co-simulationen/co-simulation-framework.html/
[9] http://www.3ds.com/products-services/catia/capabilities/systems-engineering/modelica-systems-simulation/dymola
[10] http://www.cd-adapco.com/products/star-ccm
[11] http://qucs.sourceforge.net/
[12] http://www.verified.de/products/rt-tester/

CPSs are inherently complex due to the necessary combination of the cyber and physical worlds. The engineering of trustworthy networked CPSs requires compositional modelling and analysis techniques that deal with the four dimensions of computational, physical, human, and regulatory requirements. Within each of these dimensions, there are many different modelling issues and cutting across them are common conceptual concerns, such as distribution, concurrency, and time. For example, computational models can be synchronous or asynchronous; the physical world can be divided between co-existing physical dynamics in a time continuum; human agents can have competing objectives and motives; and the system may have to conform to several different regulatory requirements. Making sense of these diverse concerns, and ensuring interoperability and communication between their components, is a major scientific and engineering challenge.

Our chosen meta-modelling notation for giving semantics to different concepts and paradigms is Unifying Theories of Programming [19]. Our technique is to isolate important language features, and give them a denotational semantics; algebraic, axiomatic, and operational semantics can then be proved sound against this model. This allows different languages and paradigms to be linked together in a coherent way. We use formal links to specify the interfaces between heterogeneous modelling concepts. These concepts can then be assembled to form the semantics for different modelling languages and further links allow heterogeneous modelling using different languages. This compositional approach also leads to compositional analysis techniques.

D. Methodology

Methods for model construction, analysis and maintenance bridge the gap between semantic foundations and the functionality of tool chains. In multi-model-based engineering for CPSs, such methods need to overcome several particular challenges. First, the multi-model construction entails collaboration between different, possibly distributed, disciplines within the same enterprise; methods are needed to facilitate both DE and CT model construction and integration. Second, techniques are needed to master the exploration of a complex design space in which both cyber and physical elements vary. Third, traceability – a challenge even in conventional systems development – is particularly demanding because of the broad range of DE and CT artefacts that are to be managed in the design set.

Model Construction

In previous work on multi-modelling for embedded systems, we outlined strategies for simple multi-model construction, broadly characterised as “DE-first”, “CT-first”, or “contract-first” [20]. The choice of a strategy is governed by factors including the availability of DE/CT expertise and model libraries, and whether multi-model construction is taking place *ab initio* or by evolution of existing multi-models. For CPSs composed of multiple independently owned and managed systems, we expect to see a distributed process of model construction, including the negotiation of interfaces, in which the negotiating parties providing constituent systems will wish to expose only limited information about their constituent systems. Methods for multi-model construction need to be developed to take account of this process [21]. We further expect to develop methods as sets of guidelines and specific patterns for structuring multi-models, focussing on features that introduce complexity, such as the modelling of faults, and error detection and recovery mechanisms [16].

Design Space Exploration

DSE is the activity of evaluating multi-models representing design alternatives at key decision points, in order to reach a solution that satisfies requirements, for example in terms of specific performance characteristics. In DSE, ranges of multi-model design parameters are selected and co-simulations are run under these settings. Results are stored for each simulation and can be analysed. DSE results typically report upon multiple objectives such as speed, accuracy and energy consumed. A ranking function can be applied to evaluate designs, though this is can be simplistic; another approach is to compute a non-dominated set of designs to determine the Pareto Optimal front [22].

Ensuring Traceability

CPS engineering requires traceability among a wide range of artefacts, including requirements, models, multi-models, analysis results, test plans and test results, generated code and physical system designs. The need for CPS dependability and the capacity to manage tool evolution both imply that there should be a trail linking development artefacts by semantic design relations; maintaining these relations allows the ramifications of design choices and changes to be assessed. The maintenance of traceability documentation can be labour-intensive and is often dropped under pressure [23]. While many tools support basic traceability links, none of them yet do so automatically [24]. In INTO-CPS, the tool chain will allow design artefacts to be stored, organised, and retrieved across different tools using Open Services for Lifecycle Collaboration[13]. Records of the provenance of artefacts can be used at a later stage as evidence in documenting the adequacy of a design.

IV. Industrial Case Studies

To evaluate INTO-CPS technology in a variety of value chains, four industry-led case studies have been formulated in diverse domains: railways, agricultural, building automation and automotive. Even if the target products of these case studies have been designed and developed previously, they are all different by nature with different technical objectives. We here describe each case study, its innovative CPS angle, and the motivation of each company for undertaking the evaluation.

[13] http://open-services.net/

A. Railways

ClearSy[14], a company with strong technical capabilities in model-based design, particularly for critical transport systems. In railway signalling, an *interlocking* is an arrangement of signal apparatus that prevents conflicting movements through an arrangement of tracks such as junctions or crossings. Based on the status of the railway system as seen from sensors and on its short-term history, the interlocking computes the status of the actuators (switches, signals). This computation is determined by signalling safety rules that depend on different countries, but also by various optimisation issues. A central interlocking can deal with a complete line, all decisions being made globally. However the distance between devices distributed along the tracks and the interlocking system may lead to a significant delay to update devices' status. So there is room for an alternate solution: a distributed interlocking in which a train/metro line is divided into overlapping interlocked zones, each zone being controlled by an interlocking. Such interlockings would be smaller as fewer local devices have to be taken into account – a local decision could be taken in shorter time and would result in potentially quicker train transfers.

The target is the ability to model specific situations accurately, e.g., where trains are at different altitudes and where train movement could result in oscillations after braking. The main property to check is the absence of collision during the co-simulations. In order to check that distances are sufficient to ensure safety, several scenarios have to be considered including maximum descending slope, train weight, braking capability, acceleration and speed.

B. Agriculture

In this case study, led by Agro Intelligence, a company with expertise in smart agricultural machinery, will develop a CPS around Robotti, an autonomous robot platform that applies a variety of soil treatments by means of different liftable implements mounted on central bars. Robotti is differentially driven by tracks or wheel modules. It can navigate pre-loaded routes over a field which may feature slope changes affecting robot motion. Additionally, the robot must deal with obstacles, which may include those that are small enough to be passed over by the robot provided it raises its implements (e.g., birds' nests), and those requiring a full stop (e.g. a human ahead of the machine). The business goal underpinning the study is the ability to produce a radically innovative product while improving the development process. For example by modelling the dynamic behaviour of the mechanical parts and interactions of the controllers, this is expected to reduce the number of prototypes by 30-60%.

C. Building Automation

This case study, led by United Technologies Research Center (UTRC)[15] is based on a CPS supporting Heating, Ventilation and Air-Conditioning (HVAC) in a building [25]. HVAC systems keep a building's climate within a specified range by tuning and controlling heat and total air ventilation. Such applications exist already, of course, but can be expensive to develop due to the complex environment in which they operate, integrating different manufacturer's components which may be sensing and making control decisions in a common complex fluid dynamic environment. To this end, interoperability between heterogeneous systems is an important factor. The case study will explore the provision of additional intelligence to the HVAC CPS so it will be able to adapt to the use of a building in an energy-friendly fashion. The study will examine compliance with relevant standards for equipment safety and performance, and provision of evidence for successful certification of the controllers of the air handling unit based on the EN15232 [26] standard. The outcome of the study is modelling solutions that will be applicable to a wide class of buildings and HVAC-controlled spaces.

D. Automotive

This case study, led by TWT[16], will develop a CPS supporting intelligent mobility assistance for electric and hybrid electric vehicles to help with their adoption by vehicle owners, thereby encouraging adoption of fuel saving strategies. The CPS will provide drivers with choices of driving modes depending on the route, desired optimum concerning range, fuel consumption or comfort. This kind of application exists already, but it is very expensive to test such solutions using real vehicles, so co-simulation offers a means of exploring aspects of a complex automotive system behaviour and systematically finding and analysing faulty behaviours at an early stage. A great challenge will be to support the exploration of an enormous design space that includes non-technical dimensions such as driver comfort. A target is evidence supporting certification to ISO 26262, which governs functional safety of automotive electric/electronic systems for all development phases and from system level to hardware/software [27]: this will provide a challenge for the traceability functionality.

V. Concluding Remarks

We have described challenges facing model-based engineering for dependable CPSs, including the needs for collaborative modelling to embrace notational heterogeneity at a semantic level, methods to manage the complexity of the design space by supporting a traceable path from requirements to implementations, and integrated tool chains evaluated through industrial practice. INTO-CPS aims to address these challenges. Building on formal foundations, we are creating a family of interlinked tools supporting CPS development from requirements and architectural modelling formalised using SysML, via FMI interface definitions to multi-models. The tool chain is intended to permit static analysis of multi-models, as well as co-simulation, including co-simulation of models with implementations of cyber and/or physical elements. We

[14] http://www.clearsy.com

[15] http://www.utrc.utc.com

[16] https://www.twt-gmbh.de

aim to allow these co-simulations to be exploited in DSE and test automation. We are evaluating the framework using applications in the rail, agriculture, automotive and building automation domains.

INTO-CPS began in January 2015 and will last three years. Ten months into the project, the industrial case study owners have created first constituent models of important elements of their systems using the baseline technologies without FMI. In parallel, the first release package of semantic foundations, methods and tools is being developed. This package will be used by the industrial case studies over the next two years. Following an iterative approach, industry needs derived from the experimental application will be fed back as requirements to the technology providers for subsequent releases of the full methods and tools package. We expect that, with members of the follow group, we will develop a set of further pilot studies that provide starting points for new users of the package.

There are major benefits to flow from the goal of creating integrated tool chains for CPSs, including improved verification of CPS-level dependability properties, and early detection of bottlenecks and defects. Nevertheless, we expect to face many challenges in creating such a tool chain, not least ever-increasing performance demands for co-simulation. Nevertheless, however good co-simulation performance gets, the need remains for significant advances in broadening the range of multi-models that can be managed, and the providing stronger support to the CPS design process.

Acknowledgment

The work presented here is partially supported by the INTO-CPS project funded by the European Commission's Horizon 2020 programme under grant agreement number 664047. We are grateful to our colleagues for all their hard work to make the vision presented here a reality.

References

[1] M. Broy, "Engineering Cyber-Physical Systems: Challenges and Foundations," in *Complex Systems Design & Management*, M. Aiguier, Y. Caseau, D. Krob, and A. Rauzy, Eds. Springer Berlin Heidelberg, 2013, pp. 1–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34404-6_1

[2] J. Fitzgerald, C. Gamble, P. G. Larsen, K. Pierce, and J. Woodcock, "Cyber-Physical Systems design: Formal Foundations, Methods and Integrated Tool Chains," in *FormaliSE: FME Workshop on Formal Methods in Software Engineering*. Florence, Italy: ICSE 2015, May 2015.

[3] A. Bagnato, E. Brosse, I. R. Quadri, and A. Sadovykh, "INTO-CPS: An integrated "tool chain" for comprehensive model-based design of cyber-physical systems," in *Revue Génie Logiciel*, June 2015, pp. 31–35.

[4] P. Mosterman and J. Zander, "Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems," *Software & Systems Modeling*, pp. 1–12, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10270-015-0469-x

[5] B. Schätz, M. Törngreen, S. Bensalem, M. V. Cengarle, H. Pfeifer, J. A. McDermid, R. Passerone, and A. Sangiovanni-Vincentelli, "Cyber-Physical European Roadmap and Strategy: Research Agenda and Recommendations for Action," CyPhERS, Tech. Rep., February 2015. [Online]. Available: www.cyphers.eu

[6] E. L. Gide, "Embedded / cyber-physical systems artemis major challenges: 2014-2020, 2013 draft addendum to the artemis-sra 2011," ARTEMISIA, http://www.artemis-ju.eu/publications, Tech. Rep., december 2013.

[7] T. Blochwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, and A. Viel, "The Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models," in *Proceedings of the 9th International Modelica Conference*, Munich, Germany, September 2012.

[8] B. Wang and J. S. Baras, "HybridSim: A Modeling and Co-simulation Toolchain for Cyber-physical Systems," in *17th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2013, Delft, The Netherlands, October 30–November 1, 2013*. IEEE Computer Society, 2013, pp. 33–40.

[9] J. Peleska, "Industrial-Strength Model-Based Testing - State of the Art and Current Challenges," *Electronic Proceedings in Theoretical Computer Science*, vol. abs/1303.1006, pp. 3–28, 2013.

[10] A. Bagnato, I. Quadri, E. Brosse, A. Sadovykh, L. S. Indrusiak, R. Paige, N. Audsley, I. Gray, D. S. Kolovos, N. Matragkas, M. Rossi, L. Baresi, M. Crippa, S. Genolini, S. Hansen, and G. Meisel-Blohm, *Handbook of Research on Embedded Systems Design*. Hershey, PA: Information Science Reference, 2014, ch. 8: MADES FP7 EU Project: Effective High Level SysML/MARTE Methodology for Real-Time and Embedded Avionics Systems, pp. 181–208.

[11] P. G. Larsen, N. Battle, M. Ferreira, J. Fitzgerald, K. Lausdahl, and M. Verhoef, "The Overture Initiative – Integrating Tools for VDM," *SIGSOFT Softw. Eng. Notes*, vol. 35, no. 1, pp. 1–6, January 2010. [Online]. Available: http://doi.acm.org/10.1145/1668862.1668864

[12] M. Verhoef, P. G. Larsen, and J. Hooman, "Modeling and Validating Distributed Embedded Real-Time Systems with VDM++," in *FM 2006: Formal Methods*, ser. Lecture Notes in Computer Science 4085, J. Misra, T. Nipkow, and E. Sekerinski, Eds. Springer-Verlag, 2006, pp. 147–162.

[13] J. F. Broenink, P. G. Larsen, M. Verhoef, C. Kleijn, D. Jovanovic, K. Pierce, and F. Wouters, "Design Support and Tooling for Dependable Embedded Control Software," in *Proceedings of Serene 2010 International Workshop on Software Engineering for Resilient Systems*. ACM, April 2010, pp. 77–82.

[14] C. Kleijn, "Modelling and Simulation of Fluid Power Systems with 20-sim," *Intl. Journal of Fluid Power*, vol. 7, no. 3, November 2006.

[15] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, January 2004.

[16] J. Fitzgerald, P. G. Larsen, and M. Verhoef, Eds., *Collaborative Design for Embedded Systems – Co-modelling and Co-simulation*. Springer, 2014. [Online]. Available: http://link.springer.com/book/10.1007/978-3-642-54118-6

[17] G. D. Plotkin, "A structural approach to operational semantics," Aarhus University, Tech. Rep. DAIMI FN-19, 1981.

[18] C. Hoare, "An axiomatic basis for computer programming," *Communications of the ACM*, vol. 12, no. 10, pp. 576–581, October 1969.

[19] T. Hoare and H. Jifeng, *Unifying Theories of Programming*. Prentice Hall, April 1998.

[20] J. Fitzgerald, P. G. Larsen, K. Pierce, and M. Verhoef, "A Formal Approach to Collaborative Modelling and Co-simulation for Embedded Systems," *Mathematical Structures in Computer Science*, vol. 23, no. 4, pp. 726–750, 2013.

[21] C. B. Nielsen, "Strengthening Collaboration, Integration and Modelling in System of Systems Engineering," Ph.D. dissertation, Department of Engineering., Aarhus University, June 2014. [Online]. Available: http://www.riverpublishers.com/pdf/ebook/RP_9788793237155.pdf

[22] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[23] M. Jarke, "Requirements tracing," vol. 41, no. 12, pp. 32–36, December 1998.

[24] P. Mäder, *Rule-based Maintenance of Post-requirements Traceability*, ser. MV Wissenschaft. MV-Verlag, 2010.

[25] J. Fitzgerald, C. Gamble, R. Payne, P. G. Larsen, S. Basagiannis, and A. E.-D. Mady, "Collaborative Model-based Systems Engineering for Cyber-Physical Systems – a Case Study in Building Automation," in *INCOSE 2016*, Edinburgh, Scotland, July 2016.

[26] "Energy Performance of Buildings – Impact of Building Automation, Controls and Building Management," European Standards, Tech. Rep., 2012.

[27] "Road Vehicles - Functional Safety - Part 6: Product development at the software level," International Organization for Standardization, Tech. Rep., 2011, iCS 43.040.10.