Software Engineering for Smart Cyber-Physical Systems (SEsCPS 2018) – Workshop Report

Tomas Bures[1], Danny Weyns[2], Bradley Schmerl[3], John Fitzgerald[4] (workshop organizers)
Adina Aniculaesei[5], Christian Berger[6], João Cambeiro[7], Jan Carlson[8], Shafiul Azam Chowdhury[9], Marian Daun[10], Nianyu Li[11], Matthias Markthaler[12], Claudio Menghi[13], Birgit Penzenstadler[14], Aedan Pettit[15], Robert Pettit[16], Luca Sabatucci[17], Christos Tranoris[18], Hans Vangheluwe[19], Sebastian Voss[20], Edith Zavala[21] (contributing participants)

[1]Charles University Prague, Czech Republic, [2]Katholieke Universiteit Leuven, Belgium & Linnaeus University, Sweden, [3]Carnegie Mellon University, USA, [4]Newcastle University, UK, [5]TU Clausthal, Germany, [6]University of Gothenburg, Sweden, [7]NOVA LINCS, Portugal, [8]Mälardalen University, Sweden, [9]University of Texas at Arlington, USA, [10]University of Duisburg-Essen, Germany, [11]Peking University, China, [12]RWTH Aachen, Germany, [13]Chalmenrs University, Sweden, [14]California State University Long Beach, USA, [15]The College of Wooster, USA, [16]The Aerospace Corporation, USA, [17]ICAR-CNR, Italy, [18]University of Patras, Greece, [19]University of Antwerpen, Belgium, [20]fortiss GmbH, Germany, [21]Polytechnic University of Catalonia, Spain

bures@d3s.mff.cuni.cz, danny.weyns@kuleuven.be,
schmerl@cs.cmu.edu, john.fitzgerald@newcastle.ac.uk

ABSTRACT

Smart Cyber-Physical Systems (sCPS) are a novel kind of Cyber-Physical System engineered to take advantage of large-scale cooperation between devices, users and environment to achieve added value in the face of uncertainty and changing environments. Examples of sCPS include modern traffic systems, Industry 4.0 systems, systems for smart buildings, and smart energy grids. The uniting aspect of all these systems is that to achieve their high level of intelligence, adaptivity and ability to optimize and learn, they rely heavily on software. This makes them software-intensive systems, where software becomes their most complex part. Engineering sCPS thus becomes a recognized software engineering discipline, which, due to specifics of sCPS, can only partially rely on the existing body of knowledge in software engineering. In fact, it turns out that many of the traditional approaches to architecture modeling and software development fall short in their ability to cope with the high dynamicity and uncertainty of sCPS. This calls for innovative approaches that jointly reflect and address the specifics of such systems. This paper maps the discussions and results of the Fourth International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS 2018), which focuses on challenges and promising solutions in the area of software engineering for sCPS.

Keywords

Software engineering, cyber-physical systems

1. INTRODUCTION

The SEsCPS workshops series, traditionally a part of ICSE, aims to address the lack of software engineering techniques and methods tailored to the specifics of sCPS by identifying challenges, opportunities and use-cases of sCPS and by exploring novel software engineering approaches for building sCPS [1][2][3][4].

The workshop brings together academics and practitioners from several disciplines with overall objectives: (i) to increase the understanding of problems of Software Engineering (SE) for sCPS, (ii) to study the foundational principles for engineering sCPS, and (iii) to identify and define promising SE solutions for sCPS.

The topics of SEsCPS traditionally include areas of architectural modeling, qualities, assurances, etc., as well as exemplars, use-cases, and case-studies [5][6][7]. Based on the interests shown by the participants at the previous edition, SEsCPS'18 further brought forward the special themes of: (1) social aspects of sCPS, (2) diversity and cooperation in sCPS, and (3) analysis and enforcement of quality properties.

In this report, we summarize the discussions and findings of the 4th edition of the workshop, held on May 27th, 2018 in Gothenburg, Sweden in conjunction with ICSE 2018.

2. WORKSHOP STRUCTURE

The workshop attracted 14 submissions, out of which 2 were accepted as full papers and 6 as position and future-trends papers. In total, around 20 participants attended the workshop. The workshop started with a keynote. The rest of the morning was devoted to short presentations of accepted papers, grouped in three themes as overviewed in the next section. The whole afternoon of the workshop was devoted to discussion in breakout groups, where participants discussed topics of SE for smart CPS that emerged from the paper presentations and discussions in the morning. A plenary report session and outlook on the future concluded the workshop.

3. KEYNOTE

The keynote was delivered by Hans Vangheluwe (University of Antwerpen, Belgium), who focused in his talk on the topic of multi-paradigm modelling. The main problem here is that the networking of multi-physics (mechanical, electical, hydraulic, biochemical, ...) with computational systems (control systems, signal processing, logical inferencing, planning ...) processes, interacting with often uncertain environments, with human actors, leads to hitherto unseen level of complexity of Cyber-Physical Systems. To date, no unifying theory nor systematic design methods, techniques and tools exist for such systems. Individual (mechanical, electrical, network or software) engineering disciplines only offer partial solutions. Multi-paradigm Modelling (MPM) proposes to model every part and aspect of such complex systems explicitly, at the most appropriate level(s) of abstraction, using the most appropriate modelling formalism(s). This includes the explicit modelling of the often complex engineering workflows. Modelling language engineering, including model transformation, and the study of their semantics, are used to realize MPM. MPM is seen as an effective answer to the challenges of designing CPS. The talk by Hans Vangheluwe introduced some of the challenges of collaborative development of CPS as well as possible multi-paradigm modelling solutions such as (in-)consistency management and co-simulation.

4. WORKSHOP THEMES

The workshop presentations provided a cross-cutting view of the software engineering challenges related to sCPS and potential approaches to address the challenges. The presentations were organized into the three themes overviewed below.

4.1 Modeling and Validation

The first theme of the workshop was concerned with modeling and validation of sCPS. An important aspect of sCPS is their inherent complexity, which comes from the typically large-scale collaboration. This calls for techniques that support modeling and validation that ensure that these systems comply with their requirements. This topic was targeted by three talks. Sebastian Voss presented methods for design space exploration with the help of SAT in collaborative systems. An important aspect here was the ability to back-report errors on the level of domain requirements. Marian Daun focused on a semi-automated approach to foster the validation of collaborative networks of sCPS. The emphasis in this work was on modeling these networks on instance and type level to help identify errors that can be discovered only at particular level of abstraction (instance/type). Nianyu Li elaborated on validation of early statistical requirements in sCPS. This provides valuable early insights through statistical techniques

4.2 Trustworthiness

The second theme discussed at the workshop focused on trustworthiness as one of the key properties of safety-critical sCPS. In the first talk of this session Robert Pettit and Aedan Pettit elaborated on the feasibility of automatically detecting and recovering from single event upsets in CPS. This was focused on micro satellites where the sCPS has to be able to cope with transient hardware faults stemming from cosmic radiation. Another aspect of trustworthiness was introduced by Christian Berger and Birgit Penzenstadler who focused on using blockchains for safety-critical systems. They described where it is beneficial to use the blockchain in software engineering of sCPS. This includes such areas as resource access frameworks, libraries of certified and trustworthy software assets and contracts and agreements in changes during software evolution.

4.3 Reference Problems

The third theme covered by the talks of the workshop focused on reference problems. The availability of reference problems constitutes a very important topic as till now, there are not many generally usable exemplars and benchmarks for experimentation with sCPS. In this respect, Christos Tranoris introduced a case study on an automotive vertical domain with 5G networking. João Cambeiro described a building automation case study. Shafiul Azam Chowdhury presented a curated corpus of Simulink models for model-based empirical studies. Lastly, Luca Sabatucci introduced a self-adaptation exemplar of shipboard power system reconfiguration.

5. OPEN RESEARCH TOPICS

The whole afternoon of the workshop was allocated to breakout groups that focused on selected topics that emerged from the morning presentations. In total, there were four groups, each focusing on one of these topics selected for discussion: “What does it mean for a multi-paradigm model to be good?”, “How to define context of models?”, “How to organize autonomy and trust in distributed CPS?”, “What are the characteristics of good sCPS exemplars?” In the rest of the section, we report on the findings of each breakout group in turn.

5.1 Good Multi-Paradigm Models

The first breakout group discussed the question of what it means for a multi-paradigm model to be good. This issue of the quality of models becomes significantly more important as sCPS typically consist of multiple models which mutually interact and need to be aligned. Obviously, a flaw in one model influences the quality of other models.

The group started by identifying a definition of a multi-paradigm model, which is a collection of abstractions, languages and processes along with viewpoints and transformations. The group pointed out the relation between meta-model and models in the sense that quality of a meta-model can only be assessed by answering the question whether good concrete models can be derived from it. Further, the group identified the criteria for evaluating how “good” models are. These are: scope, domain, purpose, tooling, evolvability, and costs.

Evolvability becomes an especially important criterion because it is directly connected to the “smartness” of sCPS. This evolvability is needed since sCPS are designed to cope with uncertainty and a successor system will, hence, need to be specified and developed with certain aspects left open since they cannot be fully anticipated. As such, a good multi-paradigm meta-model needs to be “smart” in the sense that it must be evolvable over time.

5.2 Context of Models

The second breakout group investigated the question of what the context of a model is and how to define such a context. This is an important aspect of sCPS because the context of the model has important implications on the interpretation of the model, (automatic) reasoning about the model and the use of the model in general. Also, it is important for correct evolution of the model as well as its reuse and replacement.

The group outlined the context of a model as assumptions that are connected with creation and use of the model. This comprises the system, environment, requirements and modeling language. Such a context of a model comes from multiple sources. These especially comprise the knowledge of a designer (domain knowledge and expertise, very often tacit), empirically collected and automatically derived observations, and (physical) laws.

Apart from often being tacit, a typical problem with model context is that it is never complete as it depends on the purpose of its use. Another important challenge is achieving and maintaining its consistency. Having systematic means and a process for specifying the assumptions of the model context explicitly can foster consistency. This means support for informal and formal specification and means for explicitly modeling varying degrees of (un)certainty about knowledge: exact, probabilistic, fuzzy, partial.

5.3 Autonomy and Trust in Distributed CPS

The third breakout group focused on the question of how to organize autonomy and trust in distributed CPS. The basic premise here was the apparent conflict among autonomy, trust and smartness, which are all crucial aspects of smart CPS. An sCPS has to be able to make “smart” decisions by itself, yet it has to be trustworthy. This means the sCPS has to be able to reason about confidence it has in itself and in other systems. Based on this confidence, it should enable and scale autonomy in a way that it remains trustworthy.

There are multiple levels of autonomy and trust that need to be covered when designing such smart trustworthy CPS. These in particular are: instance level (a single system), system of systems level (collaborative decision-making), organizational level. This leads to context-aware behavioral rules and obligations.

When pushing the smartness beyond pre-defined rules, sCPS need to be able to learn. This introduces further uncertainty and complicates establishing trust. In this sense, the learning has to be carefully designed to be based on penalties that can be translated into the physical world (be risk and domain specific). This then makes it easier to define, monitor and analyze the behavior of systems that exploit learning, which eventually makes it easier to establish trust.

All in all, realizing such smart trustworthy systems requires better understanding about the autonomous system domain and ability to correctly express risk and uncertainty and connect them to well defined boundaries in which a system is permitted to learn.

5.4 Characteristics of Good sCPS Exemplars

The fourth breakout group focused on identification of characteristics of good sCPS exemplars. To date, there are very few exemplars and reference problems of smart CPS. This is mostly connected to the relative youth and immaturity of the field, making it all the more important to identify good exemplars and their characteristics.

To improve this situation, the group sought to give examples and guidelines for what a good exemplar should look like, such that it is easier for the community to come up with exemplars that can be used by others to experiment with their ideas and most importantly to allow reproducibility and comparability of results.

The group started with identifying what an exemplar can be. Generally, an exemplar is an instance of an archetypical example that others could use to guide or evaluate their approaches. In the field of software engineering for sCPS, an exemplar can be any of the following:

- A “stretch” **use case** that pushes that state of the art forward
- A **challenge problem** that facilitates competitive work
- A **benchmark** that provides test cases for comparing approaches
- A **testbed** that provides the infrastructure to perform research
- A **dataset** that can be used to train algorithms, compare to ground truth, etc
- A **library** that can provide reuse of models or code
- A set of **patterns** and/or **antipatterns** of how to tackle a problem
- A description of a **process** of tackling a problem (e.g. modeling process)

The group further identified some examples of different categories of exemplars. A good example of the “challenge problem” is the SAT competition [8], which has a number of defined inputs and expected outputs and teams compete within well-defined rules. An example of a good benchmark are the SPEC benchmark suites [9], which allow reproducible comparison of different VMs, libraries and their configurations. A good example of data exemplar is the urban observatory data set [10] which comprises of data for various sensors at different geographical locations.

The important guidelines one can derive from these categories and examples is that a good exemplar should clearly state what it is intended for. In particular an exemplar should answers questions such as: Is it describing a problem or a solution? Who would use this exemplar? Further, an exemplar should be: (re)usable, (preferably) open, curated, and canonical (i.e. non-redundant in data or models).

6. ACKNOWLEDGEMENTS

The SEsCPS 2018 workshop is a collective endeavor, as such, the authors would like to express their gratitude to those who have participated in its organization. This comprises in particular the ICSE 2018 Workshops Co-Chairs, in particular Dimitra Giannakopoulou and Paul Grünbacher, and the SEsCPS 2018 Program Committee comprised of Steffen Becker, Stefan Biffl, Wolfgang Böhm, Johann Bourcier, Radu Calinescu, Jan Carlson, Sagar Chaki, Rogerio De Lemos, Antonio Filieri, Ilias Gerostathopoulos, Holger Giese, Rodolfo E. Haber, Gabor Karsai, Mark Klein, Luka Lednicki, Henry Muccini, Torvald Mårtensson, Bashar Nuseibeh, Patrizio Pelliccione, Rick Rabiser, Wolfgang Renz, Ina Schieferdecker, Lionel Seinturier, Vitor E. Silva Souza, Ingo Stierand, Jagadish Suryadevara, Bedir Tekinerdogan, Catia Trubiani, Christos Tsigkanos, Petr Tuma, Andreas Vogelsang, Thorsten Weyer.

7. REFERENCES

[1] NSF, Cyber Physical Systems, NSF 15-541 http://www.nsf.gov/pubs/2015/nsf15541/nsf15541.pdf

[2] B. K. Kim and P. R. Kumar, “Cyber–Physical Systems: A Perspective at the Centennial”, Proceedings of the IEEE, vol. 100, no. Special Centennial, 2012.

[3] E. A. Lee, “Cyber Physical Systems: Design Challenges”, 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing, 2008.

[4] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, “Cyber-Physical Systems: A New Frontier,” IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2008.

[5] T. Bures, D. Weyns, S. Biffl, M. Daun, T. Gabor, D. Garlan, I. Gerostathopoulos, C. Julien, F. Krikava, R. Mordinyi, “Software Engineering for Smart Cyber-Physical Systems – Towards a Research Agenda,” Software Engineering Notes, November 2015.

[6] T. Bures, D. Weyns, B. Schmerl, E. Tovar, E. Boden, T. Gabor, I. Gerostathopoulos, Pr. Gupta, E. Kang, A. Knauss, P. Patel, A. Rashid, I. Ruchkin, R. Sukkerd, C. Tsigkanos: Software Engineering for Smart Cyber-Physical Systems: Challenges and Promising Solutions. ACM SIGSOFT Software Engineering Notes 42(2): 19-24, 2017

[7] T. Bures, D. Weyns, B. Schmerl, J. Fitzgerald: Software Engineering for Smart Cyber-Physical Systems: Models, System-Environment Boundary, and Social Aspects. ACM SIGSOFT Software Engineering Notes 43: 42-44, 2018

[8] SAT Competition 2018, http://sat2018.forsyte.tuwien.ac.at/

[9] The Standard Performance Evaluation Corporation (SPEC), SPECjbb 2015, https://www.spec.org/jbb2015/

[10] Urban observatory, http://urbanobservatory.ac.uk/