ISSN 2186-7437

NII Shonan Meeting Report

No. 2014-14

Science and Practice of Engineering Trustworthy Cyber-Physical Systems (TCPS)

Fuyuki Ishikawa
Alexander Romanovsky

October 27–30, 2014

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Science and Practice of Engineering Trustworthy Cyber-Physical Systems (TCPS)

Organizers:
Fuyuki Ishikawa (National Institute of Informatics, Japan)
Alexander Romanovsky (Newcastle University, UK)

October 27–30, 2014

1 Overview of Shonan Meeting TCPS

1.1 Background

Human society and activities within have been depending more and more on software-intensive systems. Novel system paradigms have been proposed and actively developed, notably Cyber-Physical Systems (CPS). Envisioned systems expand target entities and processes handled by the systems, stepping into more depth of human activities as well as real world entities. There are emerging application areas such as automated driving and smart cities, while existing areas are also evolving with richer features, such as aviation, railways, business process management, navigation systems, etc. Visions for CPS include or extend a lot of variations of system paradigms, Systems of Systems, Ubiquitous Computing, Ambient Intelligence, and so on. Obviously, the increased impact on human activities and real world entities leads to strong demand for trustworthy systems. On the other hand, the result is unprecedented complexity, caused not only by expanded application features, but also by combined mechanisms for trustworthiness (self-adaptation, resilience, etc.). Construction and provision of trustworthy systems under complexity are absolutely the key challenges in system and software engineering.

The key to tackle the challenge is engineering methods for trustworthy systems. There is no doubt that foundational theories and technical components are essential as building blocks. Building blocks for trustworthy systems spread across verification algorithms, probabilistic analysis, fault models, self-adaptation mechanisms, and so on. The challenge on complexity requires further elaboration and integration of such blocks into engineering methods. Engineering methods define a systematic and reliable way for set of tasks to model, analyze and verify the system and its trustworthiness nature while mitigating the complexity. Recently, there have been yet more active efforts on engineering methods for trustworthy systems, on the basis of various approaches. Formal methods are one of the promising approaches and have accompanied active efforts not only by the academia but also by the industry. Each approach has unique features that are apparently different but essentially relevant to each other, focusing on modeling of the system, modeling of trustworthiness or faults, and their analysis and verification for complex systems, especially Cyber-Physical Systems.

1.2 Objective and Planning of the Meeting

In order to speed up the evolution of engineering methods for emerging complex Cyber-Physical Systems, it is absolutely necessary to promote active discussions beyond specific applications or specific engineering approaches. This Shonan Meeting aimed at providing this opportunity by inviting world-leading researchers on engineering methods for trustworthy Cyber-Physical Systems.

This meeting had two kinds of sessions. One is presentations and targeted discussions, where each researcher presents ideas and positions that then kicks off various directions of discussions. The other is sessions consisting of intensive follow-up discussions involving mixed groups of attendees. The meeting will use a dedicated method for conducting the intensive discussion (a variation of the world cafe method: www.theworldcafe.com/method.html).

1.3 Participants

- Yamine Ait Ameur, IRIT / INPT-ENSEEIHT France
- Toshiaki Aoki, JAIST Japan
- Keijiro Araki, Kyushu University Japan
- John S Fitzgerald, Newcastle University UK
- Kokichi Futatsugi, JAIST Japan
- John Knight, University of Virginia US
- Tsutomu Kobayashi, The University of Tokyo Japan
- Imre Kocsis, Budapest University of Technology and Economics (/ Quanopt Ltd.) Hungary
- Hironobu Kuruma, Hitachi, Ltd. Japan
- Peter Gorm Larsen, Aarhus University Denmark
- Mark Lawford, McMaster University Canada
- Thierry Lecomte, ClearSy France
- Alexei Iliasov, Newcastle University UK
- Shaoying Liu, Hosei University Japan
- Tom McCutcheon, UK Defence Science and Technology Laboratory / Newcastle University UK
- Dominique Mery, LORIA / Universite de Lorraine France
- Daichi Mizuguchi, Atelier Inc., Japan
- Shin Nakajima, National Institute of Informatics Japan

- Nguyen Thanh Hung, Hanoi University of Science and Technology Vietnam
- Andras Pataricza, Budapest University of Technology and Economics Hungary
- Inna Pereverzeva, Abo Akademi University Finland
- John Rushby, SRI International US
- Neeraj Kumar Singh, McMaster University Canada
- Carolyn Talcott, SRI International US
- Thai Son Hoang, Hitachi Ltd. Japan
- Elena Troubitsyna, Abo Akademi University Finland
- Alan Wassyng, McMaster University Canada
- Virginie Wiels, ONERA/DTIM France
- James Woodcock, University of York UK

1.4 Schedule

Oct 26 (Sun)

- 19:00-21:00 Welcome Dinner

Oct 27 (Mon)

- Breakfast
- Welcome and Overview
- Kick-off
- Introduction of Participants (1)
- Talks & Discussions: Roadmap/Vision
- Lunch
- Introduction of Participants (2)
- Talks & Discussions: Modeling (1)
- Talks & Discussions: Certificate/Assurance (1)
- Talks & Discussions: Tool (1)
- Dinner

Oct 28 (Tue)

- (Optional) Hike to Lookout Tower
- Breakfast
- Talks & Discussions: Tool (2)
- Talks & Discussions: Verification (1)
- Lunch
- Talks & Discussions: Modeling (2)
- Talks & Discussions: Domain/Application (1)
- World-cafe Discussions
- Dinner
- Night Session

Oct 29 (Wed)

- Breakfast
- Talks & Discussions: Certificate/Assurance (2)
- Talks & Discussions: Modeling (3)
- Lunch
- Excursion & Banquet

Oct 30 (Thu)

- (Optional) Hike to Lookout Tower
- Breakfast
- Talks & Discussions: Verification (2)
- Talks & Discussions: Domain/Application (2)
- Lunch
- Output Planning, Discussions, Closing

2 Summary of Discussion Session

A discussion session was conducted focusing on the following four topics:

- What are the essential differences between CPSs and traditional software systems?
- What are the technical challenges facing TCPS engineering?
- What are the non-technical challenges facing TCPS engineering?
- What are the gaps between academia and industry (research and practice) in engineering (so far/for future TCPS)?

Below is a brief summary of the issues identified.

What are the essential differences between CPSs and traditional software systems?

To understand these differences we need to first identify the essential properties of the systems we wish to develop. A CPS needs a cyber part and a physical part, and the design methods should consider both.

CPSs consist of collaborating computational elements controlling physical entities, which interact with humans and their environment. The typical layering is as follows: environment, physical systems, computer/controller, and human operator.

The main characteristics of physical systems are their continuous behaviour and stochastic nature. This is why the development of CPSs requires specialists in multiple engineering disciplines to be involved. These systems often use close interaction between the plant and the discrete event system, and their complexity can grow due to interaction among multiple CPSs and the need to deal with time.

These are some of the essential features of CPSs beyond the conventional control systems: context awareness, cognitive computation and autonomy.

What are the technical challenges facing TCPS engineering?

The technical challenges in TCPS engineering derive from the need to deal with system heterogeneity. TCSP engineering needs sound foundations to deal with modelling and analysing the parts (e.g. continuous, discrete) and their compositions. These systems are often so complex that the humans in the loop may not completely understand them, which calls for special engineering methods to develop systems that would assist humans as far as possible and make sure that the systems are always safe.

These systems have to be modelled/reasoned about together with their environments; the difficulties here are in choosing the right level of detail in representing the environment and in ensuring that all relevant elements of the environment are represented.

There is a wide range of CPSs starting from the social interaction systems and the Internet (Internet of Things) to safety-critical CPSs in medicine/surgery.

Typically the general properties of CPS systems can only be deduced from the local behaviours of their parts. The execution of these parts might have global effects and typically results in an emergent behaviour of the whole.

It is crucial to guarantee CPS trustworthiness for the guaranteed as well as non-guaranteed behaviour of the system. The safety, security and reliability risk analysis of CPSs require methods that go beyond the traditional hazard analysis.

CPS engineering will need to include methods for compositional verification and compositional certification.

What are the non-technical challenges facing TCPS engineering?

Perception of technology by the general public is crucial for deploying CPSs. Unfortunately this perception is often irrational; it is sometimes impossible to communicate probabilities and risks to the public, who often treat computer errors as normal events (glitches). We do need to better understand how people feel about new technologies and collecting data.

The development culture in companies needs to be changed, and we need to understand how to demonstrate advantages of new technologies (for example, how to switch to using formal methods). We do need a business case for formal methods.

TCPS development requires interdisciplinary approaches that rely on communication between various experts. Hopefully, the CS people will serve as a common ground as well as the ambassadors of new technologies. It is important to understand how to communicate decisions across domains.

In the area of certification and assurance the topics discussed were the liability of companies, the use of specification as contract, and the need to develop a standard of standards. One important issue is to understand when and why the life of a CPS ends.

The main challenges in education are as follows: CS engineering education does not cover continuous domain, and there is a need to educate the general public about new technologies.

What are the gaps between academia and industry (research and practice) in engineering (so far/for future TCPS)?

There is a cultural gulf between academia and industry, reified by the career requirements of both groups, but there is also a gulf within industry between research and production. There is a gap between different industries as well as between single product organisations and heterogeneous CPSs, where no one industry dominates. Sometimes companies are aware of the advantages of engagement but it is hard to implement it.

The goals are different as well. There is an academic need for academic excellence and the industrial need for commercial success. Academics are mainly into science; industry is into engineering. To add to this, there is inflexibility in both, academics and industry; for example, industries keep academics at a distance.

The successes achieved so far include governments supporting smaller joint projects (as in EU), large EU projects (e.g. the development of software for avionics has successfully mixed academics and industry –long term projects and large investment are key), moving PhDs from industry to academia and vice versa.

The ways forward are as follows. Relevant government policies should be developed, and the government should be lobbied for support. Unfortunately, ICT / CPS is not given the attention afforded to, e.g., high energy physics. In addition we need to lobby academia to promote the value of industrial engagement (the detail here is not trivial!) and the industry to show the cost implications of academic engagement. Cross-fertilisation between academic disciplines, and between academia and industry should be encouraged. There should be more academic presence on standards committees. Lastly, industry should be more involved in the teaching of CPS.

3 Overview of Talks

A generic model for system substitution

Yamine AIT-AMEUR, Guillaume BABIN and Marc PANTEL

The substitution of a system by another one occurs in several cases like adaptation, failures, resilience, reconfiguration, self-healing, etc. System substitution consists in replacing a running system by another one when a given condition holds. In this talk, we present a generic formal model for system substitution.

In our approach, a system is defined as being a transition system. Each state is characterised by a set of variables and transitions denote state changes. We consider that each system refines a global specification (another system). A set of systems, namely substitute systems, can be associated to a global specification. By system substitution, we mean the capability of a system to be replaced by another system, each of these two systems refine the same global specification. Preserving the properties of the original system is a key point to be addressed during substitution.

In this talk, we present a stepwise formal approach for system substitution. Substitute systems are formalised by Event-B machines, which refine a shared Event-B machine defining the global specification. State recovery is performed when a failure in the running system occurs. In that case, modes are changed and control is transferred to the selected substitute system. The transfer of the control shall 1) preserve safety of the properties expressed as invariants in the Event-B model and 2) identify the recovery state in the substitute system. Proof obligations associated to this substitution operation are defined. They guarantee invariant preservation. A four steps methodology is defined and a case study is shown to illustrate the developed approach.

Practical Application of Model Checking and Testing to Automotive Operating System

Toshiaki Aoki, Japan Advanced Institute of Science and Technology, Japan

The safety and reliability of automotive systems are becoming a big concern in our daily life. Recently, a functional safety standard which specializes in automotive systems has been proposed by the ISO. In addition, electrical throttle systems have been inspected by NHTSA and NASA due to the unintended acceleration problems of Toyota's cars. In light of such recent circumstances, we are working on the verification of automotive operating systems to ensure the high quality of automotive operating systems. An operating system which we focus on is the one conforming to the OSEK/VDX standard. This presentation shows a case study that model checking, which is one of formal methods, is applied to a commercial OS named REL OS. REL OS is too complicated to convince us that it correctly performs for any application. We adopted exhaustive verification techniques to check REL OS. We have conducted exhaustive testing based on a design model which was exhaustively verified by model checking. As a result, we acquired the confidence that REL OS correctly performs for any application although no new bug was found since the model checking and testing were more exhaustive and reliable than the traditional methods. Such combined model checking and testing are appropriate to convince us of the correctness thanks to their exhaustive nature.

Towards Deployment of Formal Approaches to Software Industry in Japan

Keijiro Araki, Department of Advanced Information Technology, Kyushu University, Japan

In this talk I briefly talk about our activities in introducing formal methods to real system development in Japanese companies. As well as individual collaborations with companies, I have been serving as an active member of committees and working groups on reliable system development with formal methods at a governmental organization, SEC (Software Engineering Center), IPA (Information-technology Promotion Agence, Japan). As part of the above activities, we published a survey report on successful cases of application of formal methods to real system development, which shows key points in applying formal methods syccessfully. For examples, reigourous specifications are the common artifacts referred from many places many times during the development process, and the purpose and scope of application of formal methods are set up clearly, and so on. Finally I make a quick introduction to our research project supported by the Ministry of Education on practically applicable formal methods.

Systems of Systems and Cyber-Physical Systems

John S Fitzgerald, Newcastle University, UK

Many of the most exciting applications of cyber-physical system (CPS) engineering involve the effective integration of units that are independently owned or managed. They thus have some features in common with "systems of systems" (SoSs) in that reliance is placed on them to offer a collective emergent service, even though the constituent systems are free to evolve or operate autonomously. In this presentation, we discuss the potential for model-based engineering of SoSs, and the extent to which lessons learned in that environment might be carried over to CPSs.

SoS Engineering (SoSE) is a branch of systems engineering that focuses particularly on the challenges posed by operational and managerial independence, distribution, evolution and emergence. The work that we describe on model-based SoSE (undertaken in the COMPASS project: www.compass-research.eu) has approached three specific challenges: coping with independence by taking contractual approaches to modelling constituent system interfaces; the verification of emergence by providing techniques for design space exploration based on simulation, and static analysis of SoS models composed from formal descriptions of contractual interfaces; and the need for semantics that cope with the heterogeneity of constituent systems and their interface models by providing an extensible semantic framework using the Unifying Theories of Programming.

The outcomes of our work have included sets of modelling guidelines and patterns in SysML, allowing domain-specific modelling frameworks to be developed. To date, this has been done in areas that include content streaming for home audio/video systems, and systematic fault modelling. The formal basis required to allow verification of emergence has been embodied in a modelling language CML – the first developed specifically for SoS engineering and with a UTP semantics. Tools realise the analyses defined in CML. Industry case studies have been used to evaluate the readiness for industry deployment of the model-based SoSE techniques developed to date. Much SoSE research has concentrated on digital SoSs – those whose properties of interest are largely described in computational terms. As we move to considering the design of dependable CPSs, the properties of interest range over both digital and continuous physical and human factors domains. The goal should be to enable exploration of the design space in a way that allows trade-offs to be made over both domains.

For the engineering of dependable SoSs, advances are required in foundations, methods, and tools. Form our experience in model-based SoSE, we would expect that semantic frameworks must be expanded to encompass the continuous phenomena in order to support multidisciplinary modelling. Methods research should encompass approaches to model construction, contract negotiation and renegotiation, and investigate the extent to contractual approaches apply to physical phenomena. Other open issues include the modelling and analysis of operational compensation for cyber or physical faults, and issues around assurance and certification of CPSs.

Generate & Check Method for Verifying Transition Systems in CafeOBJ

Kokichi Futatsugi, Reseach Center for Software Verification (RCSV) / Japan Advanced Institute of Science and Technology, Japan

A interactive theorem proving method for verification of transition systems is presented.

The state space of a transition system is defined as a quotient set (i.e. a set of equivalence classes) of terms of a top most sort State, and the transitions are defined with conditional rewrite rules over the quotient set. A property to be verified is either (1) an invariant (i.e. a state predicate that is valid for all reachable states) or (2) a (p leads-to q) property for two state predicates p and q. Where (p leads-to q) means that from any reachable state s with (p(s) = true) the system will get into a state t with (q(t) = true) no matter what transition sequence is taken.

Verification is achieved by developing proof scores in CafeOBJ. Sufficient verification conditions are formalized for verifying invariants and (p leads-to q) properties. For each verification condition, a proof score is constructed to (1) generate a finite set of state patterns that covers all possible infinite states and (2) check validity of the verification condition for all the covering state patterns by reductions.

Real World Types and Their Application

John Knight and Jian Xiang, University of Virginia, USA

Cyber-physical systems, especially embedded systems, interact with real-world entities in order to sense (and usually affect) the real world. Software in such systems should obey rules derived from the real-world context as well as from the machine context. However, the relationship between real-world entities and programs is quite complex. We introduce the notions of real-world types and the correspondence model. Real-world types are derived from real-world entities and document real-world entities in programs. They include all of the relevant attributes of a real-world entity together with appropriate rules for using those entities in expressions, as parameters, etc. The velocity of a moving object, for example, is more than a number. Velocity includes direction (since it is a vector), a reference frame and a unit of measurement, and the associated real-world type includes all of these attributes. The correspondence model documents and leverages the relationship between real-world entities and programs, and thereby allows programmers to take advantage of the relationship between real-world entities and programs. In essence, the correspondence model documents the link between the real-world context and the logic of the software that interact with the real world, including differences between the real-world and machine representations of real-world entities. For each real-world type, the correspondence model documents: (a) all of the relevant semantics of associated real-world entities; and (b) how those entities are represented within the machine. The correspondence model also includes rules derived from the real world including, for example, rules derived from physics. Real-world types allow programs to be written with variables that have types strongly aligned with the real-world entities that they represent. Using a variable with a real-world type for velocity allows all of the semantic details listed above to be associated with the variable together with the precision of the values available within the machine, i.e., the difference between the machine-accessible value and the actual value in the real world. For any given type, the correspondence model also includes a precise natural-language explication of the meaning of the type in the real world.5 Using real-world types and the correspondence model, programmers are able to enforce real-world rules in programs in a systematic way, thereby enabling a new class of fault detection. Real-world types have been illustrated in an implementation for Java and evaluated by applying the implementation to a set of open-source Java projects. In the evaluation numerous defects in the software were detected statically by real-world-type checking.

Refinement Engineering? – Systematic Refinement Planning of Formal Models

Tsutomu Kobayashi and Fuyuki Ishikawa, The University of Tokyo and National Institute of Informatics, Japan

In recent years software assurance has become increasingly important. Moreover, according to large complexity of software systems, complexity of software specifications has grown. Under this situation, approaches using stepwise refinement to gradually construct and verify a system specification is attracting wide attention. Among them, Event-B is promising due to flexible refinement mechanism that allows gradually adding aspects and functions of system to abstract models. Although the flexibility of refinement gives developers many possibilities of refinement, we found that intuitively planned refinement often fails to mitigate complexity or be consistent. However, existing studies on Event-B do not explicitly focus on how to choose what elements of the system should be included in each refinement step. We formalize the problem based on notions of artifact (statements of functionality and property) and phenomenon (constituents of statements). We view refinement planning as finding sequences of introduction of artifacts, while resolving dependencies between artifacts and phenomena. Based on the view, we constructed a method to derive refinement plans that mitigate complexity of modeling and prevent inconsistent models. Our method also accepts interactive filtering using human decision so that developers can reflect their preferences. Through case studies using several examples we succeeded to derive refinement plans that mitigate complexities and are consistent. The method could be used in a iterative and interactive manner to derive a limited number of desirable plans. By analyzing structure of examples and generated plans, we found that our method is effective to derive plans of complex example. We also discuss possible applications of our method, such as refactoring of refinement, evaluating existing plans and guidelines, and studying understandability or verifiability of proof for large problems.

Resilience Assurance in Cyber-Physical Clouds

Imre Kocsis, Dept. of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary

We are more and more experiencing the convergence of cyber-physical and cloud computing systems, as both fields increasingly mature to become established engineering disciplines. This gives rise to Cyber-Physical Clouds: systems that incorporate cyber-physical parts as well as utilize cloud services. However, this new systemic category introduces novel challenges of design for resilience. The talk identifies an important set of these and proposes measurement-driven resilience design strategies to address them.

In a wide range of cyber-physical scenarios – from smart cities to swarms of autonomous vehicles – the computational needs of data processing, analytics, decision and control support far outstrip what can be reasonably expected from field devices. Additionally, as the “plant” – the physical world – changes, the need arises for the computational backend to change, too; e.g. the resource needs and the necessary service assembly of a smart city operation will be radically different on “normal days” and in case of disasters. These characteristics justify and require using cloud services in order to be able to seamlessly scale as well as reconfigure the backend.

A logical extension of this basic cyber-physical + cloud pattern is the field devices to become parts of the cloud as “leasable” resource and service ensembles. Especially with recent advances in embedded and real-time virtualization, drawing field devices under the control of a cloud and/or providing cloud-like services over them is becoming technologically feasible (see e.g. [1]). Virtually the same trend can be observed in the telco domain with the push towards “carrier clouds” [2] – a cloud type that can be expected to become deeply intertwined with cyber-physical clouds in the future.

A fundamental aspect of CPS trustworthiness is timeliness in taking control decisions and actions as well as in communicating with human actors (arguably, a design problems that we haven’t mastered fully yet – see e.g. [3]). At the same time, in general purpose cloud computing the time-wise instability and population-wise heterogeneity of resource and service performance is a well-known and established threat on application performance (see e.g. [4]). The talk introduces these threats and argues that new, “cloud metrology” driven approaches are necessary to properly assess the threats and their effect on cloud-hosted applications. This, in turn, can drive the design of structural defenses and runtime dependability mechanisms inside the cloud as well as on the cloud-CPS boundary.

Dependability benchmarking is a significant element in this vision, translating to the need for “scale model” CPS + cloud – put simply, “CPS in the loop” and “cloud in the loop” – experimental environments. To this end, we have developed “cloud on cloud” capabilities for and integrated field devices to the Apache Virtual Computing Lab (VCL) educational cloud platform [5]. VCL enables the virtualization of classic computing labs through remote access to virtual machines instantiated from golden images of lab exercises, web-based reservations and intelligent backend scheduling mechanisms. We extended VCL with “cyber-physical reservations”, where a reservation is not only a set of freshly initialized virtual machines, but full cloud instances and field devices connected to these.

References

[1] Craciunas, S. S., Haas, A., Kirsch, C. M., Payer, H., Róck, H., Rottmann, A., Sengupta, R. (2010). Information-Acquisition-as-a-Service for Cyber-Physical Cloud Computing. In Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. USENIX Association.

[2] Taleb, T. (2014). Toward carrier cloud: Potential, challenges, and solutions. IEEE Wireless Communications, 21(3), 80–91. doi:10.1109/MWC.2014.6845052

[3] Lee, E. A. (2010). CPS foundations. In Proceedings of the 47th Design Automation Conference on – DAC ’10 (pp. 737–742). New York, New York, USA: ACM Press. doi:10.1145/1837274.1837462

[4] Li, Z., OBrien, L., Cai, R., & Zhang, H. (2012). Towards a Taxonomy of Performance Evaluation of Commercial Cloud Services. In 2012 IEEE Fifth International Conference on Cloud Computing (pp. 344–351). IEEE. doi:10.1109/CLOUD.2012.74

[5] Vouk, M. A., Rindos, A., Averitt, S. F., Bass, J., Bugaev, M., Kurth, A., ..., Valenzisi, M. (2009). Using VCL technology to implement distributed reconfigurable data centers and computational services for educational institutions. IBM Journal of Research and Development, 53(4), 2:1–2:18. doi:10.1147/JRD.2009.5429056

Introducing Formal Methods into Industry

Hironobu Kuruma, Yokohama Research Laboratory, Hitachi, Ltd., Japan

This talk reports insights from our experience to introduce formal methods to the industry.

Collaborative Modelling and Co-simulation: Tools and techniques for Designing Embedded Systems

Peter Gorm Larsen, Aarhus University, Denmark

The embedded systems market is a lively place, and there is growing demand for rapid innovation of products that make exploit new materials, sensors and computing hardware, often through clever and complex software. In this context, developers have to form creative teams out of disparate disciplines but the semantic gaps between disciplines cost time and money because misunderstandings are often only detected when the physical product is built and software fails to control it properly. How can model-based development work if these teams of specialist engineers describe different parts of the product and its environment in very different ways, and can formal techniques help? We have been developing practical methods for collaborative creation of "co-models" composed of discrete-event models of control devices/software and continuous-time models of the controlled devices and the environment, bridging gaps between software and other engineering disciplines. Reconciled operational semantics permit co-models to be "co-simulated", allowing us to explore the design spaces of physics and software together, so that we can trade off alternatives on such bases as performance, energy consumption and cost before committing to a solution.

This talk gives introduction and discussion about the new Crescendo toolset linking Overture-VDM and 20-sim tools. The principles of co-modelling and co-simulation are discussed as well as the experience gained in developing Crescendo, and with its industry application in the DESTECS project (www.destecs.org).

TCPS@McSCert: If at first you don't succeed . . .

Mark Lawford, McMaster University Canada

In this talk I present ongoing work at the McMaster centre for Software Certification (McSCert) on Trusted Cyber Physical Systems (TCPS) with emphasis on how we have developed a successful collaborative research program with industry partners. McSCert is focused on how we can develop and certify software intensive systems such as TCPS using product based evidence. With our partners in the Automotive, Medical, and Nuclear industries, we work on methods and tools for development and certification of software intensive systems and apply these methods to case studies. Case studies such as the scale model of an automotive Adaptive Cruise Control (ACC) provide insights into the benefits and limitations of formal models and the importance of validating the formal (mathematical) models against the actual system implementation to make sure that the models are fit for their intended purpose. As an example, we describe a formal model of the ACC system that was used to "prove" collision freedom of a system implementation but that when implemented exhibits mode thrashing causing extreme braking and acceleration under a particular driving scenario when the lead vehicle is traveling at precisely the requested open lane cruise maximum speed set by the ACC user.

Tools such as the Tabular Expression Toolbox for Matlab/Simulink are described as a way of making the latest research tools – theorem provers and SMT solvers - easily useable by practitioners in widely used development tools, while facilitating industry adoption of academic techniques such tabular methods. We describe a long term industrial interaction on the formal verification of real-time systems that through 15 years of theoretical and applied research eventually resulted in a tools supported that is now being used by our industry partners for the Formal Specification and Verification of safety critical PLCs using IEC 61131 style function blocks. This work highlights the considerable gaps that often exist between current academic research and production ready development techniques. The project also illustrates the benefits of cultivating long term relationships with industry partners.

In the remainder of the talk we provide a more detailed description of developing practical methods and tools to improve software engineering of control systems developed using Model Based Design. In order to improve the development process at an automotive partner we made use of the acausal, symbolic modeling tool MapleSim to model the physical part of an automotive CPS product line for advanced hybrid electric drivetrains. The symbolic tools were used to create models of the different drivetrain variants and then, with the help of the underlying computer algebra system, we are able to extract "calibrations" in the form of transfer functions that are used in a drivetrain configuration hardware hiding module to enable reuse of the rest of the hybrid control processor software. The computer algebra operations used to generate the calibrations are verified by automatically generating proof obligations for the PVS theorem prover as part of the forward development process. This eliminates the need for a time consuming and error prone manual derivation of the system equations and resulting transfer functions.

We end the talk by highlighting recent theoretical work that provides necessary and sufficient conditions for the existence of a software implementation given relational system requirements and input and output hardware descriptions in Parnas and Madey's 4 variable model.

More Readable / Usable / Integrated Formal Methods

Thierry Lecomte, ClearSy System Engineering

Formal methods have been introduced in the industry in order to increase the level of confidence of resulting products, through a more-or-less mechanized process. For example, assertion-based program proof was introduced in the railways in the early 90's to replace code peer review. Program proof was then partly replaced by refinement-based proof (B) for the development of safety critical software. However a formal model of a software is not guarantying by itself the safety of the plant being controlled, given that it is specification could be faulty. System-level formal modelling (Event-B) provides means to deduce and to prove software specification but analysis is performed against a selection of (small) number of dimensions: no completeness can be claimed. Formal model animation allows for playing scenarii and check the behaviour but only in a finite number of cases. Hence safety of the real system cannot be formally and totally ensured. Moreover the complexity of the industrial systems, including the degraded modes of operation, leads to huge formal models that are still difficult to handle by a single human mind.

(Towards) Design Engineering with Rodin/Event-B

Alexei Iliasov, Newcastle University UK

This talk reports our approach to Design Engineering with Rodin/Event-B for support of explorative modeling, scalability, prototyping and integration.

Modelling a Smart Traffic Light System Using SOFL – Experience and Challenges

Shaoying Liu, Hosei University Japan

A smart traffic light system is a typical cyber physical system (CPS) in which operations of heterogeneous components and their communications through networks are involved. Compared to conventional traffic light systems in which the time for displaying each of the red, green and orange lights is fixed regardless of traffic situations, the smart traffic light system is able to dynamically change the time for displaying traffic lights to ensure a smooth traffic flow in the area of junctions. This research and talk explain how the Structured Object-oriented Formal Language (SOFL) can be used to model such a smart traffic light system. Due to the complexity, the principle of "divide and conquer" is applied to allow us to focus on each sub-system responsible for monitoring vehicles, making display time plan, and controlling the traffic lights, respectively. Our experience suggests that a CPS can be naturally and comprehensibly modeled using data flow diagrams (DFD), but to achieve precise models, appropriate operational semantics and communication mechanisms for the DFD must be provided. It seems challenging if we try to use a single notation or method for the modelling of CPS as a whole but effective if the principle of "divide and conquer" is applied properly.

(T)CPS - A user perspective

Tom McCutcheon, UK Defence Science and Technology Laboratory / Newcastle University, UK

This talk discusses key aspects of Trustworthy CPS including non-technical issues.

Incremental proof-based development of medical systems Perspective for Medical Devices and Medical Domain

Dominique Méry
LORIA,Université de Lorraine

Cyber-physical systems integrate physical components that can be described by classical applied mathematics in the continuous domain and digital components that can be described by discrete state changes in the discrete domain. These systems can interact with human beings through many modalities and it leads to consider extra-knowledges for instance from the medical domain in the case of medical devices. In previous works [1,7], we consider the problem of the quality of service in digital television broadcasting and especially the evaluation of more than twenty relevant parameters and associated methods which have been specified by ETSI[1] for DVB[2]. Using the *Event-B* [2] modelling language and the refinement, we have derived a hierarchy among these parameters and validated the acyclic structure of the hierarchy made up of these parameters. It has a very important consequence on the ability to design systems-on-chip that can be related to some specific analysis of parameters and to the synthesis of new parameters to make assessment of quality of service in an easier way. The global hierarchy has allowed to confirm the intuition of domain experts. Both concepts as invariant and refinement play a central role in the organisation of the system under construction and help to validate psycho-cognitive studies. However, the integration of discrete and continuous models in the refinement process is not addressed in an operative way and we have to take into account features that are related to the medical domain as for instance the human-in-loop modelling. In fact, embedded systems or cyber-physical systems invade our daily life and among these systems, medical devices constitute a very critical class of software-based systems which may aid people with disabilities or health problems like bradycardia or inoperative heart. A French company, Carmat, confirmed Monday, September 8th, have implanted a second patient on the artificial heart developed by the company that plans to continue tests on two others persons. These systems are clearly hybrid systems, since they contain computing and physical elements. The class of medical devices interfere with biological elements which are, in a first approximation, considered as physical elements. Modelling hybrid systems require the use of so-called hybrid models [8] and model checking tools [9] are developed for analysing these models. Alur and Dill [3] developed a theory for adding time in automata and later they have developed techniques for analysing timed automata modelling systems. It is then clear that there are formalisms for expressing discrete and continuous aspects of a system made up of diverse components. Previously, action systems have been generalized by Back et all [4] to express continuous behaviours too, but refinement was not addressed. Recently, Hybrid *Event-B* [5, 6] extends the scope of *Event-B* to deal with hybrid systems and cyber-physical systems. A. Plätzer develops verification techniques [11] and tools [12] for cyber-physical systems; he addresses the definition of refinement [10] but it remains to make the refinement operative and operational, especially in *Event-B* extensions. An alternative approach [13] does not extend the *Event-B* modelling language and it proposes the use of Matlab for complementing the Rodin toolbox. In this case, the main advantage is to use the classical *Event-B* approach without new features like in Hybrid *Event-B* , requiring specific tools developments as well as a special point on the refinement in action.

[1] ETSI stands the European Telecommunications Standards Institute, which produces globally-applicable standards for Information and Communications Technologies (ICT), including fixed, mobile, radio, converged, broadcast and internet technologies (see www.etsi.org)

[2] DVB stands the Digital Video Broadcasting Project (DVB) which is an industry-led consortium of over 200 broadcasters, manufacturers, network operators, software developers, regulators and others from around the world committed to designing open interoperable technical standards for the global delivery of digital media and broadcast services (see www.dvb.org)

References

[1] D. Abraham, D. Cansell, P. Ditsch, D. Méry, and C. Proch. Synthesis of the QoS for digital TV services. In *First International Workshop on Incentive Based Computing - IBC'05*, Amsterdam, The Netherlands, 2005.

[2] J.-R. Abrial. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press, 2010.

[3] Rajeev Aur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[4] R.-J. Back, L. Petre, and I. Porres. Generalizing action systems to hybrid systems. In Mathai Joseph, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 1926 of *Lecture Notes in Computer Science*, pages 202–213. Springer Berlin Heidelberg, 2000.

[5] Richard Banach, Michael Butler, Shengchao Qin, and Huibiao Zhue. Core hybrid event-b ii: Multiple cooperating hybrid event-b machines. Under submission, 2014.

[6] Richard Banacha, Michael Butler, Shengchao Qin, Nitika Verma, and Huibiao Zhue. Core hybrid event-b i: Single hybrid event-b machines. Under submission, 2014.

[7] Dominique Cansell, Dominique Méry, and Cyril Proch. System-on-chip design by proof-based refinement. *International Journal on Software Tools for Technology Transfer (STTT)*, 11:217–238, 03 2009.

[8] ThomasA. Henzinger. The theory of hybrid automata. In M.Kemal Inan and RobertP. Kurshan, editors, *Verification of Digital and Hybrid Systems*, volume 170 of *NATO ASI Series*, pages 265–292. Springer Berlin Heidelberg, 2000.

[9] ThomasA. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. Hytech: A model checker for hybrid systems. In Orna Grumberg, editor, *Computer Aided Verification*, volume 1254 of *Lecture Notes in Computer Science*, pages 460–463. Springer Berlin Heidelberg, 1997.

[10] Stefan Mitsch, Jan-David Quesel, and André Platzer. Refactoring, refinement, and reasoning - A logical characterization for hybrid systems. In *FM 2014: Formal Methods - 19th International Symposium, Singapore, May 12-16, 2014. Proceedings*, pages 481–496, 2014.

[11] André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.* Springer, Heidelberg, 2010.

[12] The KeYmaera Project. Keymaera: A hybrid theorem prover for hybrid systems. http://symbolaris.com/info/KeYmaera.html.

[13] Wen Su, Jean-Raymond Abrial, and Huibiao Zhu. Formalizing hybrid systems with event-b and the rodin platform. *Sci. Comput. Program.*, 94:164–202, 2014.

A translator from SysML to B method for efficient software development

Daichi Mizuguchi, Atelier Corporation. Japan

Today, automotive industry has to make more and more effort to take measures against safety concerns of in-vehicle electronic systems as their functions are getting more active and intelligent. In fact, traffic accidents have happened due to malfunctions of safety-related electronic systems.

To counter these situations, an international standard: ISO 26262 ”Road vehicles - Functional safety” has been issued in 2012. Now automotive companies and suppliers are struggling to comply this standard.

The standard aims to give guidelines and best practices for development and evaluation of safety-related systems to reduce their risk as low as reasonably practicable. For that purpose, it requires that, along the overall development lifecycle of an electronic system, more rigorous design and verification techniques shall be used under more rigorous management as the level of risk gets higher. Because of this rigor, the development cost inevitably increases for functional safety. Efficient method is needed to minimize the increasing amount in the development cost.

Now, formal method is one of our company’s solutions. In fact, Japanese automotive industry is interested in formal methods to cut cost while maintaining the quality of software.

The standard requires that software development shall follow the V-model process, and each work product such as software safety requirements specification, architecture design and unit design shall be consistent with each other with the maintained traceability. But in reality, to maintain the consistency is very hard because rework between requirements, design and implementation often happens and a work product may be changed without considering the impact to other work products. To make the development process more efficient for functional safety, the left-hand-side of the V-model - requirements, architecture design and unit design and implementation - needs to be repeated without collapse.

To cope with this problem, the B method can be useful. In the B method, the consistency between design and implementation is ensured with mathematical proof of the consistency between abstract machines and refinement models/implementation models.

Further, the B method can be more practical via a combination with a modeling language such as SysML. SysML contains a variety of diagrams such as BDD (Block Definition Diagram), IBD (Internal Block Diagram) and MSD (Message Sequence Diagram). BDD defines the building blocks of a system. IBD defines what are interchanged between the blocks. And MSD defines the procedures to realize required functions.

We can see correspondence between BDD/IBD and abstract machines, and correspondence between MSD and refinement models/implementation models. Based on these correspondences, our company has started to build a translation tool to convert from SysML models to the B method models (Figure 1).

This tools can help the following:

- Quick iteration is made possible between abstract machines (software architecture) and refinements (software detail design) without losing the

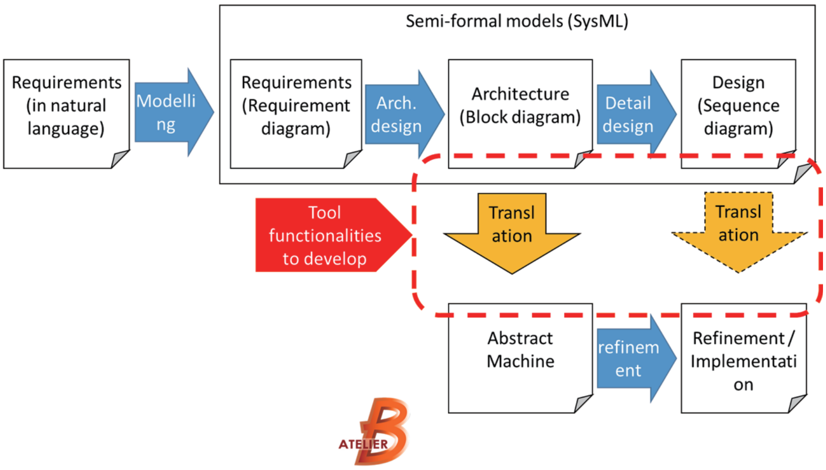
Semi-formal models (SysML)
Requirements (in natural language)
Modelling
Requirements (Requirement diagram)
Arch. design
Architecture (Block diagram)
Detail design
Design (Sequence diagram)
Tool functionalities to develop
Translation
Translation
Abstract Machine
refinement
Refinement / Implementation
ATELIER B

Figure 1: Trial development of a support tool for describing embedded software requirements specification in formal language

consistency.

- SysML models can be verified through the B method.

Then the following merits are expected in the software development lifecycle especially for functional safety:

- Review cost is reduced.
- Repetition/correction cost is reduced.
- Engineers' Motivation for design activity is increased.

Model Checking of Energy Consumption Behavior using Real-Time Maude

Shin NAKAJIMA, National Institute of Informatics

Energy consumption is one of the primary non-functional concerns, especially for application programs running on systems that have a limited battery capacity. Model-based analysis methods are introduced to supplement the current practice of runtime profiler techniques. We view that analyzing the energy consumption is considered as a duration-bounded cost constraint problem, and the problem can be encoded in logic formula. The energy consumption behavior is represented in terms of the power consumption automaton, which is a kind of weighted timed automata. In addition, a variant of linear temporal logic with freeze quantifiers is introduced. We argue that the problem is solved by model checking of such logic formulas with respect to the automaton. As the model checking is un-decidable in general, we introduce some restrictions on the patterns of the logic formulas. With appropriate abstraction techniques, we sketch how the automatic analysis is conducted using Real-Time Maude.

References

[1] S. Nakajima. Model-based Power Consumption Analysis of Smartphone Applications, In Proc. ACES-MB 2013, November 2013.

[2] S. Nakajima. Everlasting Challenges with the OBJ Language Family, In Proc. SAS 2014, pp.478-493, April 2014.

[3] S. Nakajima and M. Toyoshima. Behavioral Contracts for Energy Consumption, Ada User Journal, to appear, December 2014.

[4] S. Nakajima. Model Checking of Energy Consumption Behavior, In Proc. 1st CSDM Asia, to appear, December 2014.

Component-based verification using incremental design and invariants

Saddek Bensalem, Marius Bozga, Joseph Sifakis: Verimag Laboratory, Grenoble, France Thanh-Hung Nguyen: Department of Software Engineering, Hanoi University of Science and Technology, Vietnam

We propose invariant-based techniques for the efficient verification of safety and deadlock-freedom properties of component-based systems. Components and their interactions are described in the BIP language. Global invariants of composite components are obtained by combining local invariants of their constituent components with interaction invariants that take interactions into account. We study new techniques for computing interaction invariants. Some of these techniques are incremental, i.e., interaction invariants of a composite hierarchically structured component are computed by reusing invariants of its constituents. We formalize incremental construction of components in the BIP language as the process of building progressively complex.

Towards Dynamic Dependable Open Cyber-Physical Systems

András Pataricza, Budapest University of Technology and Economics

Our surrounding world is undergoing drastic changes due to the evolution of information technology:

- On the one hand, ubiquitous communication services facilitate the widespread use of transducers deployed in the physical environment as information sources and actuators.
- On the other hand, utility-like computing in the form of computational clouds offers practically unlimited power to perform complex and intelligent processing of data.
- Finally, the availability of knowledge and intelligence via the Internet facilitates on-demand synthesis of solutions to complex problems.

The presentation focuses on the new, emerging systemic class of dynamic, dependable, open cyber-physical systems. These are characterized by a) the use of multipurpose sensors and actuators already deployed into the environment and ready to be integrated into an application; b) the use of local servers turning the dataflow to and from the sensors/transducers into standard web services; and c) the run-time, on-demand synthesis of applications from solution patterns that are executed on XaaS offerings. The Internet-connected nature of the computational "backend" facilitates on-demand (semantic) integration of Internet-based services and deep knowledge integration.

Taking Model-Driven Design (MDD), the best practice of critical software development as a starting point, an integrated architecture and design approach are proposed and a demonstrative pilot application is showcased.

The approach uses a traditional Configuration Management Database (CMDB) in order to keep inventory of the available resources. This resource database tracks physical availability and properties as well as current and historical reservations by the individual applications. The way to express the capabilities of the different transducers is based on the W3C Semantic Sensor Network Ontology and offers a uniform abstract interface for integration via the Sensor Observation Service Model.

The synthesis of a particular application starts with a high-level, abstract description of the problem (CIM). Design patterns are used as core building blocks for the platform specific solution – taking into account the availability of transducers based on the abstract sensor related models. This paradigm also allows the integration of externally available high-level algorithms in a transparent way (such as offered commercially e.g. by the IBM Internet of Things Foundation and Wolfram).

Finally, the talk addresses the problem of self-* properties – as self-configuration, self-healing, self-optimization and self-protection – in order to assure trustworthiness. The core idea here is that the availability of cheap resources (e.g. in the cloud) and professional "algorithms as a service" offer novel opportunities for reusing traditional techniques as redundancy-based fault tolerance and design for diversity in this new setting.

Formal Development and Quantitative Assessment of a Resilient Multi-Robotic System

Inna Pereverzeva, Abo Akademi University/Turku Centre for Computer Science

Development and assessment of resilience – a property of a system to remain dependable despite changes – of complex multi-robotic systems constitute a significant engineering challenge. Decentralised architecture, asynchronous communication, component failures puts high scalability and expressiveness demands on the techniques for reasoning about resilience of multi-robotic systems. In this work we present an integrated approach to development and assessment of resilient multi-robotic systems. Our approach combines two formal techniques – refinement and probabilistic model checking – to achieve scalability and expressiveness required for reasoning about resilience of multi-robotic systems.

We demonstrate how to rigorously specify and verify essential properties of resilience mechanisms of multi-robotic systems in Event-B and derive a detailed formal system specification by refinement. Our refinement steps unfold the system architecture and introduce the required resilience mechanisms. In our case study – a multi-robotic cleaning system this corresponds to specifying the behaviour of cleaning robots and supervising base stations both in nominal conditions and in the presence of failures. When a detailed logical architecture is derived by refinement, we augment the obtained model with the probabilistic information required to conduct probabilistic resilience assessment. The automated support provided by the PRISM model checker allows us to calculate the probability of goal reachability in the presence of robot failures and compare different reconfiguration strategies for selected architectures.

Trustworthy (Self) Assembly of Systems

John Rushby, Computer Science Laboratory, SRI International

Currently, trustworthy cyber-physical systems (TCPS) such as those deployed in aviation and rail are developed in a very controlled manner. In the case of civil aircraft, the system requirements (SR) for functions to be implemented in software are developed by systems engineers and subjected to safety analysis that is intended to ensure that the complete system will operate effectively and safely, if the SRs are implemented correctly. Software development begins by constructing High Level Requirements (HLR) that serve as the top level for software development (the SR may be stated in the form of constraints, whereas the HLR will be more operational). The HLR are elaborated and refined through intermediate levels of specification and implementation until they yield executable object code (EOC). The task of software assurance is to show that the HLR are equivalent to the SR and that the EOC is correct with respect to the HLR.

The software assurance process (DO-178B/C in the case of aircraft) seems to be effective: there have been no accidents or incidents in aviation attributed to flaws in software development; there have, however, been several incidents attributed to flawed SR. More specifically, the flaws have been in parts of the SR that specify how the system is to operate; the parts that specify its safety requirements have been correct.

One way to protect against these kinds of flaws is to monitor the system against its safety requirements at runtime. Such monitors can be very simple and formally verified; with modern advances in automated synthesis based on EF-SMT solving, they could even be generated automatically. We can therefore have great confidence that they are nonfaulty, which can be expressed numerically as a "probability of perfection." The perfection of the monitor is conditionally independent of the reliability of the operational system with respect to its safety properties. Hence, the monitor provides a multiplicative increase in the safety of the system [4].

Safety monitors can be seen as moving some of the system assurance required for certification from pre-deployment into runtime. In this talk, I propose that other migrations of safety assurance functions can enable certification of TCPS that are more flexible than those employed today: for example, adaptive systems (current safety-critical systems are static), and open systems that dynamically connect with each other to form trustworthy Systems of Systems (current safety-critical systems are predefined). Adaptation- or connect-time functions may involve automated verification that the composition of local safety properties ensures and is ensured by the safety requirement of the composition, and the automated synthesis of monitors and wrappers (wrappers serve both to adapt the behavior of systems and to control or restrict their interfaces). At a more advanced (and currently speculative) level of self-assembly, we can imagine the automated construction of a system assurance case from those of its components: this would require automated interpretation of assurance cases and (the more speculative part) automated exploration of hazards in a combined system.

The purpose of automation in assembling these systems is not to replace human skill and judgment but to amplify and leverage it. Already there are projects that explore these directions (e.g., DEOS in Japan [5], AMADEOS [1] and SM@RT [6] in Europe, and ONISTT [3] and the Evidential Tool Bus [2] in USA), and there are exciting use-cases (e.g., the "operating room of the future" [7]).

This vision, with its use of automated deduction, synthesis, verification, and safety analysis at run time, goes a long way beyond "Service-Oriented Architectures" and the "Internet of Things" and, if realized, would enable trustworthy interoperation and assembly of CPS. The research program required to realize it will generate many productive capabilities along the way.

References

[1] AMADEOS Project. *Basic SoS Concepts, Glossary and Preliminary Conceptual Model*, June 2014. http://amadeos-project.eu/documents/public-deliverables/.

[2] Simon Cruanes, Grégoire Hamon, Sam Owre, and Natarajan Shankar. Tool integration with the Evidential Tool Bus. In Roberto Giacobazzi, Josh Berdine, and Isabella Mastroeni, editors, *Verification, Model Checking, and Abstract Interpretation (VMCAI), 14th International Conference*, volume 7737 of *Lecture Notes in Computer Science*, pages 275–294, Rome, Italy, January 2013. Springer-Verlag.

[3] Reginald Ford, David Hanz, Daniel Elenius, and Mark Johnson. Purpose-aware interoperability: The ONISTT ontologies and analyzer. In *Simulation Interoperability Workshop*, number 07F-SIW-088. Simulation Interoperability Standards Organization, 2007.

[4] Bev Littlewood and John Rushby. Reasoning about the reliability of diverse two-channel systems in which one channel is "possibly perfect". *IEEE Transactions on Software Engineering*, 38(5):1178–1194, September/October 2012.

[5] Mario Tokoro. *Open Systems Dependability—Dependability Engineering for Ever-Changing Systems*. CRC Press, 2013.

[6] Mario Trapp and Daniel Schneider. Safety assurance of open adaptive systems—a survey. In Nelly Bencomo, Robert France, Betty H.C. Cheng, and Uwe Assmann, editors, *Models@Run.Time: Foundations, Applications, and Roadmaps*, volume 8378 of *Lecture Notes in Computer Science*, pages 279–318. Springer-Verlag, 2014.

[7] Susan F. Whitehead and Julian M. Goldman. Getting connected for patient safety: How medical device “plug-and-play” interoperability can make a difference. *Patient Safety and Quality Healthcare*, January/February 2008. Available at http://www.psqh.com/janfeb08/connected.html.

A Perspective on Environment Modelling forVerifying Cyber-Physical Systems

Neeraj Kumar Singh, McMaster Centre for Software Certification, McMaster University Hamilton, Ontario, Canada

Analyzing requirements is a major challenge in the area of safety-critical software, where requirements quality is also an important issue to build a dependable cyber-physical system. Most of the time, any project fails due to lack of understanding of user needs, functional and non-functional system requirements, inadequate methods and tools, and inconsistent system specifications. This often results poor quality of system requirements. Based on our experience and knowledge, an environment model has been recognized to be a promising approach to support the requirements engineering to validate the system specification. It is crucial to get an approval and feedback at an early stage of the system development to guarantee the completeness and correctness of the requirements. In this talk, we propose a novel technique for analyzing requirements using environment modelling, where environment model and system model based on available requirements form a closed-loop system to trace the unidentified and hidden requirements. Moreover, the environment model also assists in the construction, clarification, and validation of the given system requirements.

Models for Design and Reasoning about Cyberphysical agents

Carolyn Talcott, SRI International

The latest sensor,actuator, and wireless communication technologies make it feasible to build systems that can operate in challenging environments, but the foundations needed to support the design of such systems are not well developed. Existing foundations provide primitives that are too strong, such as transactions. We are specifially interested in models and principles for designing and building open distributed systems consisting of multiple cyber-physical agents, where a coherent global view is unattainable and timely consensus is impossible. Such agents attempt to contribute to a system goal by making local decisions to sense and effect their environment based on local information.

We review progress including a communication framework and a preliminary formal model for exploring ideas, featuring

- communication via sharing of partially ordered knowledge,
- making explicit the physical state as well as the cyber perception of this state, and
- the use of a notion of soft constraints for specifying goals and guiding agent behavior

We begin with a discussion of desiderata for new foundations and conclude with a discussion of some challenges such as

- asynchronous communication primitives that preserve some anonymity

but support runtime building of trust

- design and reasoning principles for trustworthy CPS

Developing Systems Guided by Safety and Liveness Requirements

Thai Son Hoang, Yokohama Research Laboratory, Hitachi Ltd., Japan

Developing systems satisfying their desirable properties is a non-trivial task. Formal methods have been seen as a possible solution to the problem. Given the increasing complexity of systems, many formal methods adopt refinement techniques, where systems are developed step-by-step in a property preserving manner. In this way, a system's details are gradually introduced into its design within a hierarchical development.

System properties are often categorised into two classes: safety and liveness. A safety property ensures that undesirable behaviours will never happen during system executions. A liveness property guarantees that eventually desirable behaviours will happen. Ideally, systems should be developed in such a way that they satisfy both their safety and liveness requirements. Although safety properties are often considered the more important ones, we argue that having live systems is also important. A system that is safe but not live can be useless. For example, consider an elevator system that does not move. Such an elevator system is safe (nobody gets hurt), yet worthless.

We present Unit-B [4], a formal method inspired by Event-B [1] and UNITY [3]. Unit-B aims at the step-wise design of software systems satisfying safety and liveness properties. The method features the novel notion of coarse and fine schedules, a generalisation of weak and strong fairness for specifying events' scheduling assumptions. Based on events schedules, we propose proof rules to reason about progress properties and a refinement order preserving both liveness and safety properties. An overview of the Unit-B method can be seen in Figure 2.

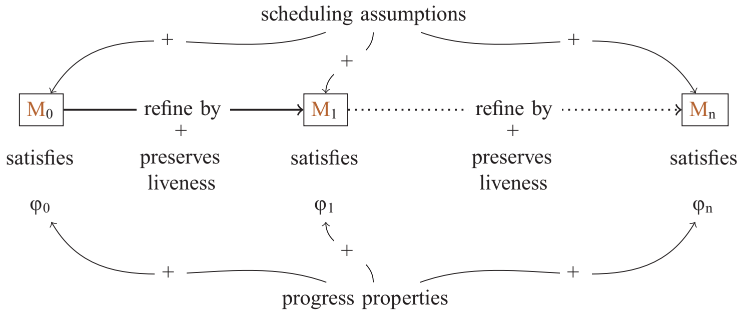

Figure 2: Overview of the Unit-B method

An Unit-B system is modelled by a transition system, where the state space is captured by variables v and the transitions are modelled by guarded events. Furthermore, Unit-B has additional assumptions on how the events should be scheduled. Using an Event-B-similar syntax, a Unit-B event has the following form:

```
e
  any t where
    G   // Guard
  during
    C   // Coarse-schedule
  upon
    F   // Fine-schedule
  then
    S   // Action
  end
```

where t are the event's indices, G is the event's guard, C is the event's coarse schedule, F is the event's fine schedule, and S is the event's action changing state variables v. The scheduling assumption of the event is specified by C and F as follows: if C holds continually and F becomes true infinitely often then event e is carried out infinitely often.

Properties of an Unit-B model includes both safety and liveness.

- **Invariance**, e.g., $\square I$, stating that a property (e.g., I) always holds. Invariance properties are proved using the standard (inductive) invariance principle.

- **Unless**, e.g., $P \,\mathbf{un}\, Q$, stating that if a property (e.g., P) holds then either it holds forever or it holds until another property (e.g., Q) holds. Unless properties such as $P \,\mathbf{un}\, Q$ are proved on per-event basis to guarantee that if the event starts in a state satisfying $P \land \neg Q$ then a state satisfying $P \lor Q$ is reached.

- **Progress**, e.g., $P \leadsto Q$, stating that if a property (e.g., P) holds then eventually another property (e.g., Q) holds. Techniques to reason about progress properties are mainly from UNITY including transient rule, ensure rule, and induction rule [3].

Refinement in Unit-B is also performed on a per-event basis. Consider an abstract event $\mathsf{e_a}$ and a corresponding concrete event $\mathsf{e_c}$ as follows:

$$\mathsf{e_a} \mathrel{\widehat{=}} \textsf{any } t \textsf{ where } G_a \textsf{ during } C_a \textsf{ upon } F_a \textsf{ then } S_a \textsf{ end}$$

$$\mathsf{e_c} \mathrel{\widehat{=}} \textsf{any } t \textsf{ where } G_c \textsf{ during } C_c \textsf{ upon } F_c \textsf{ then } S_c \textsf{ end}$$

Standard proof obligations for refinement preserving safety properties include guard and action strengthening. For preserving liveness properties through refinement, the following condition has to be proved:

$$(\square\, C_a \;\land\; \square\lozenge\, F_a) \quad\Rightarrow\quad \lozenge(\square\, C_c \;\land\; \square\lozenge\, F_c) \qquad \text{(REF\_LIVE)}$$

For practical reason, condition (REF_LIVE) can be broken down into the following conditions:

- **Coarse-schedule following**

$$C_a \land F_a \;\leadsto\; C_c \tag{C\_FLW}$$

- **Coarse-schedule stabilising**

$$C_c \;\textbf{un}\; \neg C_a \tag{C\_STB}$$

- **Fine-schedule following**

$$C_a \land F_a \;\leadsto\; F_c \tag{F\_FLW}$$

These conditions are either progress or unless properties, which can be proved within the Unit-B method.

In the future, we plan to extend the supporting Rodin platform [2] of Event-B for reasoning about Unit-B models.

References

[1] J-R. Abrial. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press, 2010.

[2] Jean-Raymond Abrial, Michael Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. Rodin: An Open Toolset for Modelling and Reasoning in Event-B. *Software Tools for Technology Transfer*, 12(6):447–466, November 2010.

[3] K. Mani Chandy and Jayadev Misra. *Parallel program design - a foundation.* Addison-Wesley, 1989.

[4] Simon Hudon and Thai Son Hoang. Systems design guided by progress concerns. In Einar Broch Johnsen and Luigia Petre, editors, *Integrated Formal Methods*, volume 7940 of *Lecture Notes in Computer Science*, pages 16–30, Turku, Finland, June 2013. Springer-Verlag. http://dx.doi.org/10.1007/978-3-642-38613-8_2.

Leveraging Impact of Formal Modelling in Development of CPS

Elena Troubitsyna, Abo Akademi University, Finland

CPS are complex heterogeneous systems that require a wide spectrum of methods and tools for their analysis. There is a strong need for integrating different approaches to modelling and analysis of CPS. Formal methods are indispensable for ensuring system trustworthiness. In my talk, I will present our work on integrated modelling and assessment of CPS. I will discuss how to integrate the results of safety analysis into formal system development by refinement in Event-B. I will also present our work on integrating probabilistic reasoning into Event-B modelling - an integration that enables quantitative assessment of such important properties as safety and reliability. Finally, I will discuss how to extract safety cases - a structured argument about system safety

- from Event-B models. In my talk, I will argue that the discipline of system engineering is still to emerge. To facilitate this process, the research efforts should be put into integrating formal modelling into the heterogeneous CPS engineering landscape.

New Standards for Trustworthy Cyber-Physical Systems

Alan Wassyng, McMaster University, Canada

Cyber-Physical Systems (CPS) are extremely complex safety-critical systems that combine physical and cyber interfaces and components. It is imperative that these systems be developed and certified to be safe, secure and reliable – hence the focus on Trustworthy Cyber-Physical Systems. Current safety-critical or high-integrity standards are primarily process based, and these standards have not proven to be effective enough even in the production of less complex safety-critical, software intensive systems. Assurance Cases have been gaining traction as a way of documenting claims about critical properties of a system together with evidence and an argument as to why the claims are valid. We have been exploring the use of Assurance Case Templates that can be used to drive development of a software-intensive critical system, as well as document a convincing argument regarding the trustworthiness of the resulting system. We believe that such a domain specific template could serve as a standard for development and certification of a CPS in a specific domain, and be much more effective than the standards we have now. To that end we discuss the role of standards, the shortcomings of current standards, the advantages of an assurance case template based standard, and essential components and concepts of such a template.

A few necessary steps between efficient Formal Methods and operational Formal Methods

Virginie Wiels, ONERA/DTIM France

This talk will highlight a few steps that should be done to speed up the industrial deployment of formal verification. These steps concern methodology, tools, certification and integration into an industrial process. They are illustrated on concrete examples, based on the experience we have in formal verification of critical embedded systems in the aeronautical domain.

Trustworthy Cyber-Physical Systems

James Woodcock, University of York UK

Cyber-Physical Systems (CPSs) integrate computation with physical processes: embedded computers monitor and control physical processes and feedback loops continuously influence computations. Their applications span many domains, from communication through healthcare to manufacturing and transportation. CPSs are distributed control systems that are networked, socio-technical, adaptive, predictive, and intelligent. Engineering trustworthy CPSs requires the formation of structured arguments as well as the production of evidence. To support these arguments and their evidence, the modelling and analysis techniques required for CPSs must be heterogeneous, and they will include continuous, discrete, concurrent, stochastic, and real-time models. This semantic heterogeneity is a significant scientific obstacle: no single tool or discipline covers all aspects of CPS design. Support is needed for diverse abstractions, vocabularies, and traditions and successful collaboration between diverse experts is critical. In this chapter, we set out a long-term vision to answer the question: ”How can we provide people with cyber-physical systems they can bet their lives on?” Trustworthiness for CPSs is truly a societal challenge