Softw Syst Model (2016) 15:933–935
DOI 10.1007/s10270-015-0510-0

GUEST EDITORIAL

Theme issue on Integrated Formal Methods

Einar Broch Johnsen[1] · **Luigia Petre**[2]

Received: 1 November 2015 / Revised: 11 November 2015 / Published online: 8 December 2015
© Springer-Verlag Berlin Heidelberg 2015

This theme issue of the *Software and Systems Modeling* journal is dedicated to the topic of *Integrated Formal Methods.* Formal methods allow the modeling and analysis of various aspects of a system. Modeling languages differ in the system aspects they target; numerous techniques address model analysis in these languages, specialized for different kinds of properties. Thus, applying formal methods may involve the modeling of different aspects of a system through different formal paradigms. Correspondingly, different analysis techniques will be integrated to examine differently modeled system views, different kinds of properties, or simply in order to cope with the sheer complexity of the system.

In recent years there has been a great deal of interest addressing the scalability of such hybrid or integrated formal models and analysis techniques. In addition, several formal methods have matured to the point where they can be deployed on industrial-scale applications, in big part due to their increasing tool support. The aim of this theme issue is to provide a resource that describes the state of the art in integrated formal methods and to outline a roadmap that addresses key challenges in this area.

The theme issue grew out of an open call for high-quality submissions on *Integrated Formal Methods*. Some of the theme issue papers are extended, and thoroughly revised papers initially published in the proceedings of **iFM 2013: the 10th International Conference on Integrated Formal Methods, Turku, Finland, June 10–14, 2013**. This is one of the major conferences in Formal Methods, organized approximately every one and a half years. In the last decade, the integration of formal methods has attracted such a significant interest in the research community that many other conferences adopt it. However, it is the focus of the iFM conference to select the most interesting and technically relevant papers on the topic. The 2013 edition collected 25 reviewed papers and 4 invited papers. To celebrate the 10th edition of this conference, the Software and Systems Modeling journal kindly agreed to host a theme issue on the Integrated Formal Methods topic. We are much indebted to the Editor-in-Chief Bernhard Rumpe and to the Assistant Editor Martin Schindler for helping us prepare this theme issue.

Out of 41 articles originally submitted to the theme issue, a total of 11 articles were accepted; of these, shorter versions of 5 articles were initially published in the iFM 2013 proceedings. We have grouped the articles appearing in this theme issue as follows: The first two articles address model-checking verification methods; the following two articles propose formal verification frameworks with respect to certain properties; we continue with two papers on formal modeling and analyzing cryptographic protocols; we then move on to three papers studying refinement together with progress properties analysis, symbolic execution, and simulation methods; and we end up with two more general articles, one on integrating distribution and control and the other on integrating formal verification with software product lines.

– **An overview of model checking practices on verification of PLC software, by Tolga Ovatman, Atakan Aral, Davut Polat, and Ali Osman Ünver**: In this paper, the authors address *verification methods for programmable logic controllers* (PLC), in particular model-checking techniques. PLC software is often the result of refining formal models, and so, applying a model-

✉ Einar Broch Johnsen
einarj@ifi.uio.no

Luigia Petre
lpetre@abo.fi

[1] Department of Informatics, University of Oslo, Oslo, Norway

[2] Faculty of Science and Engineering, Åbo Akademi University, Turku, Finland

checking technique to a correct-by-construction model requires some analysis with respect to the type of the verification technique used. The authors provide an overview of several model-checking approaches for analyzing PLC properties.

- **Model-checking software library API usage rules, by Fu Song and Tayssir Touili**: Modern software often employs third-party software via APIs. To verify the correct usage of the API rules, the authors first model these rules using specialized temporal logic extensions and then inquire via adapted model checking whether the programs using libraries violate the API usage rules or not. The approach is tool-supported, avoids false alarms that may occur in more traditional approaches to model checking, and detects several previously unknown bugs in open-source programs.
- **A formal verification framework for static analysis, by Elvira Albert, Richard Bubel, Samir Genaim, Reiner Hähnle, Germán Puebla, and Guillermo Román-Díez**: In this paper, the authors integrate static analysis with formal verification in the following sense. The program information gathered during its static analysis is translated into specification contracts that contain enough information in order to be verified automatically. In this way, they do not verify the tools for static analysis, but the results of the tools. This framework is employed to produce verified resource guarantees with respect to sequential Java programs.
- **A framework for deadlock detection in core ABS, by Elena Giachino, Cosimo Laneve, and Michael Lienhardt**: In this paper, the authors integrate formal methods for the *detection of deadlocks* in a concurrent object-oriented language. The language has asynchronous method invocations and allows cooperative scheduling, dynamic creation of resources, and features recursion, and hence, detecting deadlocks is complex and non-scalable in general. The authors integrate a method for extracting abstract behavior—the contract—with methods for contract analysis, in particular for deadlock detection.
- **Automated anonymity verification of the ThreeBallot and VAV voting systems, by Murat Moran, James Heather, and Steve Schneider**: Secure voting protocols that do not employ cryptography are quite complex and thus not previously formally analyzed. By employing abstraction and other modeling mechanisms in the CSP formal method, the authors propose the first automated formal analysis of the anonymity properties for these protocols. In so doing, they uncover the ambiguity of one of the crucial assumptions for anonymity, namely the *short ballot assumption* and propose a solution for this.
- **Constructing and verifying a robust Mix Net using CSP, by Efstathios Stathakidis, David Williams, and James Heather**: In this paper, the cryptographic protocol Mix Net is modeled with the CSP formal method and verified with its FDR model checker. This improves the state of the art with respect to properties proved for Mix Net, as typically only safety properties are ensured; the authors prove liveness properties as well. Moreover, they prove the correct functioning of the protocol, under the assumption that a majority of the servers involved in the protocol are acting as they should.
- **The Unit-B method: refinement guided by progress concerns, by Simon Hudon, Thai Son Hoang, and Jonathan Ostroff**: In this paper, a new formal method, Unit-B, is proposed as the result of integrating two formal methods, Event-B and UNITY; instrumental in the approach is the concept of *coarse and fine scheduling*. The reason for the integration is to combine Event-B's ability to model safety properties with UNITY's feature for modeling liveness properties. Unit-B allows provable correctness by construction when both safety and liveness properties drive the development.
- **Integrating deductive verification and symbolic execution for abstract object creation in dynamic logic, by Stijn de Gouw, Frank de Boer, Wolfgang Ahrendt, and Richard Bubel**: Deductive verification in a weakest precondition calculus is integrated with symbolic execution, to overcome the backward processing order implicit in calculating weakest preconditions. The resulted assertion language framework promotes a forward reasoning style in the sense that the calculus behaves like a symbolic interpreter of the program to be verified. The proof rules are fully implemented in the KeY theorem prover for Java programs.
- **Contract-based verification of discrete-time multi-rate Simulink models, by Pontus Boström and Jonatan Wiik**: Model-based design of control systems is often achieved with Simulink; since control systems have high reliability requirements, verifying the Simulink models is necessary. In this paper, the authors propose a translation from Simulink models to sequential programs, which are then verified with respect to some initial contracts. Automation for generating proof obligations to prove contract satisfiability, automation for the proofs themselves, as well as the correctness of the verification process are also addressed.
- **Knowledge-based construction of distributed constrained systems, by Susanne Graf and Sophie Quinton**: This paper studies a formalism for distributing centralized specifications upon proving some properties for them. Proving properties is easier for the centralized version, so it should be done before the distribution takes place. However, the distribution should be made so that the proven properties still hold for the system as a collection of its distributed parts. The formalism of choice

in the paper is Petri Nets, but the methodology is more widely applicable.

– **Feature Nets: behavioural modelling of software product lines, by Radu Muschevici, José Proença, and Dave Clarke**: Feature Nets are a formalism integrating Petri Nets into software product line engineering, to model not only single software systems, but software product lines. In this way, one gets a modular framework able to verify that a particular software system in the product line respects its requirements. Importantly, Feature Nets provide a facility for the incremental construction of a single model that includes the various behaviors exhibited by the products in a software product line.

We are very happy with the end result of the Integrated Formal Methods theme issue and hope that the readers will enjoy the papers.

Einar Broch Johnsen, Oslo
Luigia Petre, Turku
October 30th, 2015

Einar Broch Johnsen is professor at the Department of Informatics at the University of Oslo. His main research interests are in object-oriented, concurrent, and distributed systems and in particular in developing formal models and analysis techniques to predict their behavior. His work in formal methods includes the development of modeling abstractions and their semantics, type systems, deductive verification, testing, and rapid prototyping. In recent years, he has worked on virtualized systems, including the formal modeling and analysis of resource management on the cloud. He is the coordinator of the European FP7 project Engineering Virtualized Services (Envisage). He has been PC chair of FMOODS 2007, iFM 2013, and ESOCC 2016, and General Chair of DisCoTec 2008 and FM 2015. Prof. Johnsen has published around 100 peer-reviewed research papers.

Luigia Petre is an associate professor of Computer Science in the Faculty of Science and Engineering at Åbo Akademi University (Turku, Finland). She got her PhD in Computer Science in 2005 on modeling techniques in formal methods. Her research interests include formal methods and their integration, wireless sensor networks, network architectures, meta-modeling, non-functional properties, and time-space dependent computing. She has supervised 11 MSc students, three PhD students and currently has one PhD student under supervision. She was granted funding from the Academy of Finland to lead a consortium research project named FResCo during 2013–2015 and has been coordinating NODES—the Nordic Network on Dependable Systems (financed by Nordforsk), concerned with deploying a dependability curriculum for the Nordic countries, during 2007–2012. She has organized a winter school and several conferences; most notably, she has actively participated in the International Conference on Integrated Formal Methods, organizing it twice in Turku (2002 and 2013), as a program committee member of this conference in 2002, 2004, 2005, 2007, 2012–2014, as a program committee chair in 2013 and as a member of the steering committee for this conference since 2014. Dr. Petre has edited two books and three special issues of international journals; she has published about 45 peer-reviewed articles.