Int J Softw Tools Technol Transfer (2009) 11:355–357
DOI 10.1007/s10009-009-0121-6

EDITORIAL

Preface

Tiziana Margaria · Mieke Massink

Published online: 20 October 2009

In a press release on 4 February 2008, the Association for Computing Machinery announced that the 2007 A. M. Turing Award would be given to three of the pioneers of Symbolic Model-Checking; Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis. The Press release continues by several statements that motivate the assignment of the prize such as "Their innovations transformed this approach from a theoretical technique to a highly effective verification technology that enables computer hardware and software engineers to find errors efficiently in complex system designs" and the statement by ACM President Stuart Feldman saying that "This is a great example of an industry-transforming technology arising from highly theoretical research". Needless to say that the whole Formal Methods community and in particular the *International Workshop on Formal Methods for Industrial Critical Systems* (FMICS) considers the prize as a great recognition of their incessant efforts on theoretical research and on technology transfer of the resulting techniques to industrial reality.

Besides the enormous progress that has been made over the last decades in finding ways to efficiently encode the often huge state space that models of industrial problems give rise to, there has been much work on extending the technique to allow also for the analysis of quantitative aspects of system models. Examples of these extensions are the development of real-time model checking techniques resulting in tools like UPPAAL [10] and KRONOS [3], but more recently, also performance and dependability aspects can be analysed by probabilistic and stochastic model checkers such as PRISM [9] and MRMC [8].

The latter techniques even led to an interesting collaboration between two fields of research that, up to very recently, were completely separated: formal methods and performance analysis. Each field used its own models, theory and tools. The reciprocal interest in each other's approaches has now led to an improvement of analysis methods in both fields, e.g. an approach on the reduction of the state space based on optimal lumping of states has been shown to coincide with strong Markovian bisimulation and the algorithms involved in stochastic model checking have been able to take profit of many efficient algorithms developed in the field of performance analysis.

The extension of model checking covering also aspects of performance analysis has also another advantage: it allows for the integrated analysis of both qualitative and quantitative aspects based on the same underlying system model. This is an important step towards maintaining formal links between different "views" on the system under analysis.

Although model checking can be used to analyse many more system aspects nowadays, there remain an abundance of new challenges to tackle in the future. Among those we would like to mention recently proposed techniques to allow for transient analysis of systems composed of huge numbers of equal components sharing resources. These techniques are based on the insight that process algebraic specifications of such systems could be transformed into sets of ordinary differential equations, abstracting from the identity of the individual components and considering the number of components that find themselves to be in a particular state as a continuous rather than a discrete quantity [7].

Techniques that can be used to address properties of systems composed of many identical components that collabo-

T. Margaria (✉)
Chair Service and Software Engineering,
Universität Potsdam, Potsdam, Germany
e-mail: margaria@cs.uni-potsdam.de

M. Massink
CNR-ISTI, Pisa, Italy
e-mail: mieke.massink@isti.cnr.it

rate in some way would also be needed for the analysis of industrial systems. As examples we like to mention the analysis of gossip protocols. These are protocols based on many small processes distributed in a dynamic network that can be used to spread information through the network. Questions regarding the effectiveness, efficiency and reliability of such protocols are far from easy to answer with current formal methods techniques [16].

Similar kinds of challenges can be found in application areas such as the analysis of intelligent swarms. In the technical sense, the term 'swarm' refers to a large grouping of simple components working together to achieve some goal and produce significant results. The use of swarm technologies has become of interest in a number of application domains such as medical sciences, bioinformatics and space applications [14].

A further challenging field of application of formal methods is that of human–computer interaction. Most services will ultimately be accessed by human users, but also in the emerging field of pervasive computing implicit human interaction with a surrounding continuously changing system poses many unresolved design issues, where the interplay of multiple humans and systems may easily lead to critical situations. The application of formal methods, including model-checking techniques, has been proposed here in the past (see e.g. [6]), but raised less attention than expected, despite interesting results, such as for example the work by John Rushby [15] on the analysis of mode-confusion by pilots of aircraft with automatic flight control systems.

To remain in the realm of industrial applications, future challenges for the application of model-checking approaches can also be found in the ongoing development of the Service Oriented Computing paradigm as ever more industries seek to offer their software products in the form of services that can be accessed over the Internet and which can be offered in an uniform and integrated way with products of other companies. Such a view requires a compositional approach to the creation of products and services that can guarantee their quality. Challenges such as composability of services, behavioural and semantic conformance of processes, dynamic and adaptive processes that can respond to dynamic changes of the environment in which they function and the composition of quality of service constraints are only some of the many problems that need to be addressed if the Service Oriented Paradigm is going to be a success [5,11].

In this context, it should not surprise that two of the four contributions of this special section are indeed dedicated to the use of model-checking techniques in industry. The first contribution, by de la Cámara et al. [4], discusses how model checking can be made suitable for the verification of C-code that uses external functionality provided by an operating system via APIs.

The second contribution, by Wijs et al. [17], uses model checking to address scheduling problems and apply the technique to a Clinical Chemical Analyser used to automatically analyse patient samples with the aim to improve the throughput of the system by finding optimal schedules.

The third contribution, by Raffelt et al. [13], discusses automatic techniques to obtain behavioural models of existing legacy software for which no formal specification of their behaviour is available. The technique is based on automata learning where finite state automata can be constructed based on the observation of the system's behaviour in its reaction to test sequences.

The fourth contribution, by Mikáč and Caspi [12], illustrates on a case study how the industrially successful SCADE/Lustre environment can be enhanced to support a refinement-based development style. The Flush platform additionally integrates a number of formal verification techniques (including model checking, but also abstract interpretation and theorem proving), that are helpful in proving properties of the system and of the refinements.

The tenth edition of the FMICS workshop, from which the articles in this special section have been selected and subsequently extended, was held in 2005 in Lisbon and was co-located with the ESEC/FSE conference. It also marked the 10 year existence of the FMICS working group which provided a good occasion for a reflection on the results obtained and on the new and remaining open problems that form the research challenges for the industrial application of formal methods in the decade to come. The start of such a discussion was provided by Jonathan Bowen and Michael Hinchey, who, exactly 10 years after the publication of their much cited opinion paper "Ten Commandments of Formal Methods" [1], presented their follow-up paper "Ten Commandments Revisited—A Ten-Year Perspective on the Industrial Application of Formal Methods" as an invited presentation at the FMICS05 workshop [2]. The paper re-examines the original ten requirements (or "commandments") that formal developers should consider and discusses their validity in the light of a further decade of industrial best practice. Many of the commandments are shown to be as relevant to the current status of formal methods as they were 10 years ago. This is not to say that no progress has been made, on the contrary. For example, the issue of the selection of the appropriate notation for the specification of the problem at hand has seen enormous developments. It has become clear that no single notation will be suitable to address all aspects of a complex system. This immediately raised the problem of ways to combine different notations in more or less closely coupled and in more or less formal ways. Directly related to the choice of notation is also the kind of concepts that can be addressed and the type of analysis and verification that can be applied.

The above discussed issues are only a few of the many future challenges that the FM community is facing in the next decade. A brief preface to this special issue would not be the right place to pretend an exhaustive overview. We hope nevertheless that you will find the articles in this issue interesting and that it will stimulate you to join the future FMICS workshop issues to share ideas with other researchers and practitioners from industry and research institutions around the world.

We thank all the members of the programme committee and all the additional referees for their, sometimes repeated, evaluation of the submitted papers.

References

1. Bowen, J.P., Hinchey, M.G.: Ten commandments of formal methods. Computer **28**(4), 56–63 (1995)
2. Bowen, J.P., Hinchey, M.G.: Ten commandments revisited: a ten-year perspective on the industrial application of formal methods. In: FMICS '05: Proceedings of the 10th International Workshop on Formal Methods for Industrial Critical Systems, pp. 8–16. ACM Press, New York (2005)
3. Daws, C., Olivero, A., Tripakis, S., Yovine, S.: The tool KRONOS. In: Hybrid Systems III: Verification and Control. LNCS, vol. 1066, pp. 208–219. Springer, Berlin (1995)
4. de la Cámara, P., del Mar Gallardo, M., Merino, P., Sanán, D.: Checking the reliability of socket based communication software. Int. J. Softw. Tools Technol. Transf (this issue)
5. EU-IST-3-016004-IP-09 Sensoria Project. http://sensoria.fast.de/
6. Harrison, M., Timbleby, H.: The role of formal methods in human-computer interaction. In: Formal Methods in Human–Computer Interaction, pp. 1–8. Cambridge University Press, New York (1990)
7. Hillston, J.: Fluid flow approximation of pepa models. In: Proceedings of Second International Conference on Quantitative Evaluation of Systems (QEST'05), pp. 33–42. IEEE Computer Society Press, Washington, DC (2005)
8. Katoen, J.-P., Khattri, M., Zapreev, I.S.: A Markov reward model checker. In: QEST '05: Proceedings of the Second International Conference on the Quantitative Evaluation of Systems (QEST'05) on the Quantitative Evaluation of Systems, pp. 243. IEEE Computer Society, Washington, DC (2005)
9. Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM: Probabilistic symbolic model checker. In: Computer Performance Evaluation/TOOLS. LNCS, vol. 2324, pp. 200–204 (2002)
10. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a Nutshell. Int. J. Softw. Tools Technol. Transf. **1**(1–2), 134–152 (1997)
11. Margaria, T. (ed.): Cover features: service orientation. IEEE Comput. **40**(11), 33–80 (2007)
12. Mikáč, J., Caspi, P.: Flush: an example of development by refinements in SCADE/Lustre. Int. J. Softw. Tools Technol. Transf (this issue)
13. Raffelt, H., Steffen, B., Berg, T., Margaria, T.: LearnLib: a framework for extrapolating behavioral models. Int. J. Softw. Tools Technol. Transf (this issue)
14. Rouff, C.A., Hinchey, M.G., Truszkowski, W.F., Rash, J.L.: Experiences applying formal approaches in the development of swarm-based space exploration. Int. J. Softw. Tools Technol. Transf. **8**(6), 587–603 (2006)
15. Rushby, J.: Using model checking to help discover mode confusions and other automation surprises. Reliab. Eng. Syst. Safety **75**(2), 167–177 (2002)
16. Voulgaris, S., van Steen, M.: An epidemic protocol for managing routing tables in very large peer-to-peer networks. In: Self-managing distributed systems. LNCS, vol. 2867, pp. 299–308. http://www.springerlink.com/content/cwymm0ae0p2r5v4f/ (2003)
17. Wijs, A.J., van de Pol, J.C., Bortnik, E.M.: Solving scheduling problems by untimed model checking. Int. J. Softw. Tools Technol. Transf (this issue)