Softw Syst Model (2011) 10:283–286
DOI 10.1007/s10270-010-0171-y

GUEST EDITORIAL

Guest editorial to the theme issue on non-functional system properties in domain specific modeling languages

Marko Bošković · Dragan Gašević · Claus Pahl · Bernhard Schätz

Published online: 11 August 2010

Complexity of software system has been recognized as the major cause of difficulties in making software system development an engineering discipline. As identified by F. Brooks (F. Brooks, *The Mythical Man-Month*, Addison Wesley, 1995), complexity of software system consists of the *essential complexity*, the complexity of problems that are part of requirements placed upon software systems, and *accidental complexity* caused by the use of current software engineering methods and technologies.

Current trends in the use and development of information technology raise the both dimensions of complexity. Potentials recognized in software systems increase the number of their applications, and thus, causing the number and size of problems to grow. Increasing demands for software systems also push different vendors in producing implementation technologies, which promise to have solutions for all problems and such technologies require significant knowledge from software developers, often not very helpful in solving domain problems.

As a solution to constant increase of complexity, model driven engineering (MDE) arises as one of the most prominent approaches. Based on ideas of *direct representation*, where solutions to problems are specified within the problem domain using domain specific terms and models, and *automation*, where implementations of software systems are (semi)automatically generated from domain specific models, MDE promotes (software) system development by problem-domain experts (e.g. home automation, insurance, performance, reliability) and not implementation experts.

M. Bošković (✉) · D. Gašević
School of Computing and Information Systems,
Athabasca University, 1 University Drive,
Athabasca, AB T9S 3A3, Canada
e-mail: marko.boskovic@athabascau.ca

M. Bošković · D. Gašević
School of Interactive Arts and Technology,
Simon Fraser University Surrey, 13450 102 Avenue,
Surrey, BC V3T 5X3, Canada

D. Gašević
e-mail: dgasevic@acm.org

C. Pahl
School of Computing, Dublin City University,
Glasevin, Dublin 9, Ireland
e-mail: cpahl@computing.dcu.ie

B. Schätz
Fortiss GmbH, Guerickestraße 25, 80805 Munich, Germany
e-mail: schaetz@fortiss.org

1 Domain specific modeling languages

Going through the history of programming languages, different attempts have been made to move away from implementation problems caused by large amount of expertise needed for implementation of specified solutions. The second generation of languages introduced assemblers, textual machine instructions, and abstracted programs from binary representations, used at the time. After the complexity of execution platforms and application domains became too overwhelming, software engineers started exploiting the third generation of languages, such as Cobol, Fortran and C. The third generation of languages improved developer friendliness for program development and enabled support for programmer-defined data structures. Finally, the fourth generation of programming languages is often understood to comprise domain specific languages (DSLs). That is,

languages dedicated for the development of software systems of particular application domains. DSLs support specification of systems in terms familiar to experts of those systems' application domain. From these specifications, implementations are generated automatically.

MDE broadens spectrum of domains offered by DSLs, by introducing domain specific modeling languages (DSMLs). Besides modeling languages for functional specification of software systems, DSMLs include modeling languages and estimation methods for so-called non-functional system properties (NFPs), such as timeliness, cost, reliability, availability, safety, security among others. NFPs have been recognized to be as important as functional, because, often despite the fact that the functional requirements of software systems are met, not meeting requirements can lead to unusable software systems. Furthermore, the definition of functional and non-functional requirements often varies from domain to domain. For example, timeliness is often considered functional requirement in automotive systems, while in domains such as insurance systems it is non-functional. Finally, integration of MDE makes a step forward to making software development a true engineering discipline, because it facilitates prediction of different properties of systems before they are actually implemented and deployed.

2 Non-functional system properties in DSMLs

Domain specific modeling languages in MDE enable different roles in software engineering process to focus on their domains of expertise. However, during software systems development, collaboration between different roles is necessary. Because knowledge of participants in software development process can significantly differ, collaborative software development with DSMLs mandates sophisticated methods of integration. Owing to variety of domains, and NFPs estimation methods, this challenging task needs involvement of different communities. Some of identified issues are the following:

- *Principles and methods for specification and analysis of interrelations between functional and non-functional system properties.*
  Functional and non-functional properties are recognized as orthogonal dimensions in software development. Investigation of methods and principles for specification and analysis of different interrelations between them is necessary for enabling software systems with best tradeoffs.
- *Principles and methods for specification and quantification of interrelationships between different non-functional system properties.*
  Different non-functional system properties are analyzed with different models and methods. Development of principles and methods for quantification of interrelations between results of different modeling formalisms is also needed for obtaining best tradeoffs.
- *Methods and principles for specification of different concerns in different languages.*
  Different concerns are usually analyzed with their own formalisms. However, often there is a need for specification of input data for analysis in the basic model and returning the results of analysis. Presenting all data in the one model can make models hard to comprehend. Furthermore, specification of all information required for the analysis of different concerns would require experts to know the semantics of those concerns. Therefore, methods and principles for specifying *annotations*, finding appropriate *level of information required* from different design participants, and *presenting analysis results* to different roles in design process are an important prerequisite.
- *Alignment of terminology in the design process.*
  Mutual understanding of the presented information is essential in collaborative design processes. Different communities often have a different understanding of the same terms. For example, in business administration performance means productivity of the process, and in software engineering it is a characteristic of a system. Having the aligned terminology, misunderstandings between different participants in the process is removed.
- *Alignment of analysis of non-functional system properties with certification authorities and legal regulations.*
  Generally more external, certification and legal authorities are also participants in the design process. To be able to produce a system that can be legally certified, analysis results must be specified in the manner understandable to them.

3 In this theme issue

With the aim to discuss research challenges related to non-functional system properties in domain specific modeling languages (NFPinDSML), a series of workshops has been established affiliated with the MODELS conference. This theme issue follows up the 1st NFPinDSML workshop. The papers were solicited through an open call. Overall 12 submissions are received. Through a rigorous reviewing process, 6 submissions have met the quality requirements of the Journal on Software and Systems Modeling.

With an intention for provide an approach to develop service-oriented systems, Nora Koch, Stephen Gilmore, László Gönczy, Philip Mayer, Mirco Tribastone, and Dániel Varró in "*Non-functional properties in the model-driven*

development of service-oriented systems" introduce a DSML implemented as a UML profile entitled UML4SOA. To evaluate various different non-functional system properties, they also introduce transformations from UML4SOA models annotated with MARTE profile, to PEPA process algebra.

In the paper "*A dependability profile within MARTE*", Simona Bernardi, José Merseguer, and Dorina C. Petriu extend the existing UML profile for modeling and analysis real-time and embedded systems with concepts for modeling and quantitative analysis of dependability, and herewith introduce a DSML for such purposes. Dependability is a characteristic of a system that encompasses a vast number of non-functional properties. In this work, the authors focus on extending MARTE with modeling and analysis constructs for reliability, availability, maintainability, and safety.

To bridge the gap in communication between safety engineers, software engineers and certifying authorities, Yvan Labiche, Gregory Zoughbi, and Lionel Briand, in the paper "*Modeling safety and airworthiness* (*RTCA DO-178B*) *information: conceptual model and UML profile*", introduce a metamodel for specification of safety and airworthiness information according to RTCA DO-178B standard. Furthermore, they introduce a UML profile for integration of specifications in UML models. Finally, they introduce an approach to generating certification related information from design models, to ensure that safety requirements are met in the design process.

In the paper "*A framework to support alignment of secure software engineering with legal regulations*", Shareeful Islam, Haralambos Mouratidis, and Jan Jürjens address the problem of alignment of security of software systems with legal regulations by proposing a framework that enables elicitation of security requirements from the appropriate laws and regulations, and traceability of those requirements throughout the development stages.

Security issues are also a topic of Christiano Braga's paper "*A transformation contract to generate aspects from access control policies*". In this paper, Braga addresses the challenge of generating code with correct implementation of security aspects specified in the model. In this paper, Braga focuses on transformations from role-based access control language SecureUML to the aspects for access control language.

The paper entitled "*Beyond loop bounds: comparing annotation languages for worst-case execution time analysis*" by Raimund Kirner, Jens Knoop, Adrian Prantl, Markus Schordan, and Albrecht Kadlec introduces a comparative study of annotation languages for the worst-case execution times. The paper presents the state-of-the-art in the area of worst-case execution time analysis, identifies open issues, and can serve as a guideline for the future research.

Author Biographies

Marko Bošković is a graduate from the Military Academy, Belgrade, and holds a Ph.D., from the University of Oldenburg, Germany. Currently he is a PostDoc Fellow in the School of Computing and Information Systems at Athabasca University, and School of Interactive Arts and Technology at Simon Fraser University, Canada. He is researching in the area of integrating non-functional properties assessment techniques in the development of Service Oriented Software Product Lines. His research interests are integration of non-functional system requirements estimation and evaluation in Domain Specific Modeling Languages, and model-driven engineering in general. He served as reviewer for several journals and international conferences and workshops. He is on the editorial board of one electronic journal.

Dragan Gašević is a Canada Research Chair in Semantic Technologies and an associate professor in the School of Computing and Information Systems at Athabasca University. He is also an adjunct professor in the School of Interactive Arts and Technology at Simon Fraser University and an associated research member of the GOOD OLD AI Research Network at the University of Belgrade. He is a recipient of Alberta Ingenuity's 2008 New Faculty Award. His research interests include semantic technologies, software language engineering, technology-enhanced learning, and service-oriented architectures. In these areas, he has (co-)authored around 250 research papers. He has been serving on the editorial boards of three international journals and has edited special issues in journals such as IET Software and IEEE TSE. He has been the organizer, chair, and member of program committees of many international conferences.

Dr. Claus Pahl is a graduate from the University of Technology in Braunschweig and holds a Ph.D., from the University of Dortmund. He is a senior lecturer and is the leader of the Software and Systems Engineering Research Group at Dublin City University, which focuses on software architecture, model-driven development, and service engineering in particular. Claus has published more than 190 papers. He is a reviewer for journals and a member of more than 80 program committees for various conferences and workshops. He has organized major international conferences and is on the editorial board of five journals.

Bernhard Schätz heads the group for "Model-Based Development" at the Research and Transfer Institute Fortiss gGmbH. Furthermore, he holds a position as a permanent senior researcher at Technische Universität, München, where he is currently on leave. His research interests include models and tools for the model-based development of embedded real-time systems. Specific fields of interest are models for the support of early phases (e.g., requirements, functional and logical architectures), integration of domain-specific formalisms (e.g., computational-oriented and reactive formalisms), and mechanized techniques for the verification (e.g., timing properties) and transformation (e.g., function combination) in the development of embedded software systems. He received his Ph.D., in Computer Science in 1998 at the Technische Universität, München. Besides his membership in several national and international committees, he is/was member of the organizing committee of the MODELS'09-Workshop on "Models and Evolution", Organizing/Programm Committee Dagstuhl-Workshop "Modellbasierte Entwicklung eingebetteter Systeme" MBEES 2005-2009, the Dagstuhl Seminar on "Model-Based Engineering of Embedded Real-Time Systems", Chair Industrial Track of the IEEE Conference "ECBS'06 Engineering of Computer-Based Systems" 2006, member of the Committee of the Industrial Track at the GI Conference "Modellierung 2006. On the practical side, besides several industrial consultancy projects (including Deutsche Bundesbank and BMW AG), he is a co-founder and member of the advisory board of Validas Model Validation AG (see www.validas.de).