Journal of Logical and Algebraic Methods in Programming 85 (2016) 123–124

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Editorial preface for the JLAMP Special Issue on Formal Methods for Software Product Line Engineering

This special issue is devoted to the themes of the FMSPLE workshop series on formal methods and analysis in Software Product Line Engineering (SPLE). SPLE aims at developing a family of (software) systems by reuse in order to reduce time-to-market and to increase product quality. The correctness of the artefacts intended for reuse, as well as the correctness of the developed products, is of crucial interest for many safety-critical or business-critical applications. Formal methods and analysis techniques have been successfully applied in single system engineering in order to rigorously establish critical system requirements. While SPLE has matured considerably over the last decade, many challenges still remain, among which efficient variability management, the consistency between domain and application engineering, the reduction of quality assurance efforts, and the consistent and sustainable evolution of product families. However, formal methods and analysis techniques are still not applied broadly enough in SPLE, despite their potential to improve product quality. One of the reasons for this is that existing formal approaches from single system engineering do not consider variability, the quintessential feature of product lines.

Initiated in 2010, FMSPLE is a yearly workshop with the following long-term objectives:

1. Raise awareness and find a common understanding of practical challenges and existing solution approaches in the different communities working on formal methods and analysis techniques for SPLE.
2. Create a broader community interested in formal methods and analysis techniques for software product lines in order to keep SPLE research and tools up-to-date with the latest technologies and with practical challenges.

An open call for papers led to the submission of ten papers. After extensive reviewing, during which each paper was reviewed by at least three experts, the following six papers have been accepted.

The article by Alexander von Rhein, Thomas Thüm, Ina Schaefer, Jörg Liebig and Sven Apel, *Variability Encoding: From Compile-Time to Load-Time Variability*, provides a formal definition of the process of variability encoding, starting from the code base of the configurable program in a Java-like core language and ending with the generation of a variant simulator, as well as a proof of its behavioural correctness.

The article by Qinglei Zhang and Ridha Khedri, *On the Weaving Process of Aspect-Oriented Product Family Algebra*, presents an aspect-oriented product family algebra for the formal specification of product families and their aspects and a formal semantics of the algebraic weaving operation (of aspects into base specifications) as a term rewriting system, together with proofs that this weaving process is convergent and that its rewriting system is terminating and confluent.

The article by Damiano Zanardini, Elvira Albert and Karina Villela, *Resource-Usage-Aware Configuration in Software Product Lines*, compares four implemented strategies for the resource-usage-aware configuration of software product lines based on static analysis, which aim to find a selection of features with good behaviour from the point of view of resource usage whilst complying with user-defined quality constraints.

The article by Fatemeh Ghassemi and Mohammad Reza Mousavi, *Product Line Process Theory*, studies properties of PL-CCS, an extension of CCS with a binary variant operator used for the behavioural modelling of software product lines. These include the definition of different notions of behavioural equivalence based on bisimilarity, a sound and ground-complete axiomatisation of PL-CCS terms modulo product line bisimilarity, and a proof of the fact that product line bisimilarity has the same distinguishing power of the multi-valued modal μ-calculus.

The article by Carlos Camacho, Luis Llana and Alberto Núñez, *Cost-Related Interface for Software Product Lines*, extends the SPLA algebra for formal representations of variability models with a facility for attaching costs to actions in order to specify and evaluate the cost of adding new features to a product. The associated cost depends both on the action (i.e. features to be added) and on the history of earlier actions (i.e. features added before).

The article by Malte Lochau, Stephan Mennicke, Hauke Baller and Lars Ribbeck, *Incremental Model Checking of Delta-Oriented Software Product Lines*, introduces DeltaCSS, a delta-oriented extension of CSS to formalise behavioural variability in

http://dx.doi.org/10.1016/j.jlamp.2015.09.006

software product line specifications in a modular way, and presents and evaluates an incremental model-checking strategy and prototypical implementation in Maude to efficiently verify DeltaCCS models.

We thank all authors for their contributions and the referees for their careful and exhaustive reviewing. We would particularly like to thank Don Batory for accepting our invitation to write a Foreword to this special issue. We are grateful to Rocco De Nicola, Editor-in-Chief of the Journal of Logical and Algebraic Methods in Programming (JLAMP), Luca Aceto, the editor responsible for JLAMP special issues, and Alberto Lluch Lafuente, the editorial manager, for their support in the creation of this special issue. Finally, we are grateful to the people at Elsevier for their assistance and patience during the editorial process.

Maurice H. ter Beek
ISTI–CNR, Pisa, Italy
E-mail address: maurice.terbeek@isti.cnr.it

Dave Clarke
Uppsala University, Sweden
E-mail address: dave.clarke@it.uu.se

Ina Schaefer
TU Braunschweig, Germany
E-mail address: i.schaefer@tu-braunschweig.de

Available online 25 September 2015