

Introduction to special issue: papers from UML&FM'2011

I. Perseil · J. P. Gibson

Received: 30 April 2011 / Accepted: 1 September 2011 / Published online: 21 October 2011
© Springer-Verlag London Limited 2011

The workshop *Unified Modeling Language and Formal Methods* (UML&FM'2011) aims at encouraging new initiatives of building bridges between informal, semi-formal and formal notations.

For more than 15 years till now, the two communities of UML and formal methods have been working together to produce a simultaneously practical (via UML) and rigorous (via formal methods) approach to software engineering. This fourth edition of the workshop was open to various subjects and attracted contributions from researchers and practitioners interested in all aspects of integrating UML and formal methods.

Thirty-two papers were submitted and reviewed by the International Program Committee. Among them, eleven have been selected for their quality as well as for their interest in terms of discussions for the workshop attendees.

The workshop was held for a full day in Limerick, Ireland, on 20 June 2011. After an introduction of the topic by the workshop organizers, the accepted papers have been presented through four sessions. All presentation slides are available on the UML&FM'2011 web site,¹ but in the following we briefly describe the content of the event.

The special issue contains 10 revised and substantially extended papers based on presentations at the workshop and an overview of a the new Action Language standard (Alf). The presentations have been grouped into four sessions.

I. Perseil (✉)
INSERM, 101, rue de Tolbiac, 75013 Paris, France
e-mail: isabelle.perseil@inserm.fr

J. P. Gibson
Telecom Management SudParis, 9 rue Charles Fourier,
91011 Evry Cedex, France
e-mail: paul.gibson@it-sudparis.eu

1 Session 1: model-driven development

In *A Generic Framework: from Modeling to Code*, Neeraj Singh and Dominique Mery present an approach to building generic framework for rigorous model-driven development (MDD) that is based on combining semi-formal notations with formal modelling languages, correctness of the system using model checker and automatic code generation from the verified formal specification. The main objective of this work is to apply model-driven techniques and tools with formal verification and its code generation for designing critical systems. An assessment of the proposed framework is given through a case study, relative to the development of a cardiac pacemaker system.

In *Formal Object Oriented Development of a Voting System Test Oracle*, J Paul Gibson, Jean-Luc Raffy and Eric Lallet report on the formal development of a test oracle for an e-voting system, detailing their approach which was based on a lightweight integration of Event-B, UML and Java. This lightweight integration focuses on a coherent development process rather than on formal integration at the model/language level.

2 Session 2: SysML

In *Development of an automated MBT toolchain from UML/SysML models*, Jonathan Lasalle, Fabien Peureux and Frederic Fondement propose an approach based on existing test generation and test execution tools that makes possible to automatically derive and execute functional test cases from UML or SysML models. Its process is composed of the following steps: modeling (UML or SysML functional view),

¹ <http://www.artist-embedded.org/artist/Programme-Slides,2272.html>.

abstract test case generation (symbolic execution of the model), concretization (generation of executable test scripts from abstract test cases) and analysis (assignation of the test verdict). This process is automated by a toolchain based on Topcased modeler, Smartesting test generator and Clemessy TestInView.

In *UML / SysML semantic tunings*, Ileana Ober, Iulian Ober, Iulia Dragomir and El Arbi Aboussoror describe how applying their framework on large industrial models revealed the fact that some features of the UML/SysML semantics which lead to bottlenecks in verification are not actually necessary in the models that they have considered, thus leaving place for optimisations. Their paper discusses the gap existing between the choices made in the general UML/SysML semantic framework and the actual needs of the users. They illustrate it based on the semantics of ports, for which they give a simplified version of the semantics.

In *Formal verification of components assembly based on SysML and interface automata*, Samir Chouali and Ahmed Hammad propose an approach which combines component SysML models and interface automata in order to assemble components and to verify formally their interoperability. They specify component-based system architecture with component SysML Block definition diagrams, and the composition links between components with Internal Block Diagrams. Components protocols are specified with sequence diagrams, they are necessary to exploit interface automata formalism. Interface automata is a common Input Output (I/O) automata-based formalism intended to specify the signature and the protocol level of component interfaces.

3 Session 3: Z, B and OCL

In *Test Data Generation for Web Application using a UML Class Diagram with OCL Constraints*, Shoichiro Fujiwara, Kazuki Munakata, Tadahiro Uehara, Yoshiharu Maeda and Asako Katayama present their current work toward efficient and effective verification of web applications. Their approach begins with formally representing specifications in a basic design phase. This paper argues that application behaviors in the basic design can be expressed by states of screen and database around events like screen transitions. They use a UML class diagram with OCL to describe the application behaviors and data constraints.

In *Coupling-based UML Transformations of Formal Z Specifications*, Andreas Bollin demonstrates by the example of formal Z, that the utilization of measures can play an important role during forward as well as reverse engineering activities. It suggests to make use of coupling-based measures and shows how to overcome parts of existing impediments. The basic idea is to optimize the values of coupling for all related operations. With that two issues are addressed: firstly,

the mapping of operations to classes and methods is based on reproducible measures that are intuitively comprehensible. Secondly, due to maximizing the values for coupling between methods within a class, a possible deferred implementation also very likely has comparable properties.

In *A Proposal for Extending UML-B to Support a Conceptual Model*, Thiago C. De Sousa, Colin F. Snook and Paulo Sergio Muniz Silva present a proposal for extending UML-B to support a conceptual model in order to provide an easier starting point for the actual development process. More precisely, they propose two diagrams to facilitate the passing from requirements to the initial formal model: a first one to represent system behavior based on UML 2 Interaction Overview Diagram (IOD) and a second one for system structure based on Boundary–Control–Entity Stereotyped Class Diagram (BCE). They show how to translate the former into an Event-B specification and explain how to link the latter to the original UML-B using a simple ATM example as case study. In *Combining UML and B for the Formal Specification of an Access Control Filter*, Jérémy Milhau, Akram Idani, Regine Laleau, Mohamed Amine Labiadh, Yves Ledru and Marc Frappier present a methodology to specify access control policies starting by a set of graphical diagrams: UML for the functional model, SecureUML for static access control and ASTD for dynamic access control. These diagrams are translated into a set of B machines. Finally, the authors present the formal specification of an access control filter that coordinates the different kinds of access control rules and the specification of functional operations.

4 Session 4: UML2 State Machines

In *Institutionalising UML 2.0 State Machines*, Daniel Calegari and Nora Szasz define an institution for UML 2.0 State Machines. The building blocks of their institution are based on a previous semantics dealing with processing simple input events within a transition step. They also extend these semantic definitions for handling sequences of events, and then for considering runs through the state machine.

5 Overview of a new action language standard

In *Alf formal*, Isabelle Perseil describes the state-of-the-art of the new action language for Foundational UML, with its advantages and drawbacks. The author concludes by stating that it is very important to use a language which allows a good integration with formal methods; therefore, the Alf language should be extended for the need of critical systems design.

Acknowledgments The organizers would like to thank Mike Hinchey, Emil Vassev and David Sinclair for their support and their advices for the successful organization of this workshop.