Innovations Syst Softw Eng (2008) 4:185–187
DOI 10.1007/s11334-008-0061-4

EDITORIAL

# Introduction to special issue: papers from UML&FM

**Jean-Michel Bruel · Agusti Canals · Sébastien Gérard · Isabelle Perseil**

Received: 1 July 2008 / Accepted: 26 July 2008 / Published online: 12 August 2008


The workshop *Unified Modeling Language and Formal Methods* (UML&FM'2008) aims at encouraging new initiatives of building bridges between informal, semi-formal and formal notations. Indeed, for more than a decade now, the two communities of UML and formal methods have been working together to produce a simultaneously practical and rigorous approach to software engineering. This workshop was open to various subjects and attracted contributions from researchers and practitioners interested in all aspects of integrating UML and formal methods. Thirty nine papers were submitted and reviewed by the International Program Committee. Among them, 15 have been selected (which means an acceptance rate of 38%) 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 at Kitakyushu City, Japan, the 27th of October, 2008. After an introduction of the topic by the workshop organizers, the accepted papers have been presented through four sessions. Then an in-depth discussion of previously identified subjects emerging from the submissions has followed through a panel. A summary of the discussions is available on the UML&FM'2008 web site[1], but in the following we briefly describe the content of the event.

This special issue contains 15 revised and substantially extended papers based on presentations at the workshop. The presentations have been grouped into four sessions followed by a panel, detailed in the following subsections.

## 1 Session 1: End-to-end methodologies

In *Towards a Traceability Model in a MARTE-based Methodology for Real-Time Embedded Systems*, Hung Le Dang, Hubert Dubois and Sébastien Gérard address the traceability management in the context of ACCORD, a MARTE-based methodology for designing distributed real-time embedded systems. The contribution allows us to include requirements directly in the modeling process. The approach also supports the identification of potential traceability links that are modeled using the SysML requirement profile.

In *Foundations of a New Software Engineering Method for Real-time Systems*, Isabelle Perseil and Laurent Pautet present the foundations of a new software engineering method for real-time systems that enables integration of semi-formal and formal notations. This new software engineering method is mostly based upon the Continuum co-modeling methodology, and a model-driven development process.

In *An MDE-Based Method for Bridging Different Design Notations*, Tian Zhang, Frédéric Jouault, Jean Bézivin and Xuandong Li provide an MDE-based approach to build bridges between informal, semi-formal and formal notations: First, different notations are viewed as different DSLs and introduced into MDE, especially into the AMMA platform, by meta modeling. Then, ATL transformation rules are built

J.-M. Bruel (✉)
IRIT, Université de Toulouse, 31700 Blagnac, France
e-mail: bruel@iut-blagnac.fr

A. Canals
C-S, 31500 Toulouse, France
e-mail: agusti.canals@c-s.fr

S. Gérard
CEA, 91191 Gif-sur-Yvette, France
e-mail: gerard@cea.fr

I. Perseil
TELECOM ParisTech, 75013 Paris, France
e-mail: isabelle.perseil@telecom-paristech.fr

[1] http://www.artist-embedded.org/artist/UML-FM-08.html.

for semantics mapping. At last, TCS based model-to-text syntax rules are developed, so as to map models to programs. Consequently, different notations in both graphical style and grammatical style are bridged.

In *Safe Design of High-Performance Embedded Systems in a MDE framework*, Huafeng Yu, Abdoulaye Gamatié, Eric Rutten and Jean-Luc Dekeyser have used the UML MARTE profile to model high-performance embedded systems in the Gaspard2 framework. The design correctness issue is addressed by using the formal validation tools associated with synchronous languages. In this context, model transformations act as the bridge between UML and the chosen validation technologies.

## 2 Session 2: Formal requirements

In *From Use Cases to Test Cases via Meta Model-based Reasoning*, Stefan Gruner presents a project illustrating the way the faulty notion of dependency of the UML Use Case notation can be fixed. First, a detailed meta model of possible relations between Use Cases has been developed. This meta model has been formalized in a natural deduction calculus which can be implemented in PROLOG. As a result of such formalization, a Use Case specification can be queried for inconsistencies as well as for Test Cases. Software tool support for this method is also under development.

In *Model-based Requirements Analysis for Reactive Systems with UML Sequence Diagrams and Coloured Petri Nets*, Kristian Bisgaard Lassen and Simon Tjell describe a formal foundation for a specialized approach to automatically checking traces against real-time requirements. The traces are obtained from simulation of Coloured Petri Net (CPN) models of reactive systems. The real-time requirements are expressed in terms of a derivative of UML 2.0 high-level Sequence Diagrams. The automated requirement checking is part of a bigger tool framework in which VDM++ is applied to automatically generate initial CPN models based on problem diagrams. These models are manually enhanced to provide behavioral descriptions of the environment and the system itself.

In *Validation of Requirements Models by Automatic Prototyping*, Li Dan, Xiaoshan Li, Jicong Liu and Zhiming Liu present an approach for transforming UML system requirements models with OCL specifications into executable prototypes. A UML Use Case system operation can be formally defined by a pair of pre and post conditions specified using OCL in the context of the conceptual class model. By analyzing the semantics of the pre and post conditions, the execution of the operation can be prototyped as a sequence of primitive actions. Based on this approach, the authors have developed a tool of automatic prototype generation and analysis: $AutoPA3.0$.

## 3 Session 3: Refinements, transformations

In *Rich Meta Object Facility Formal Integration Platform - Syntax, Semantics and Implementation*, Ralf Buschermöhle and Jörg Oelerink formalizes and extends the Meta Object Facility of the OMG towards a platform to integrate and explore different formal approaches and their relationships. The extension focuses primarily on the definition of algorithms including mechanisms to support the management of arbitrary language layers.

In *Incremental Development of UML Specifications Using Operation Refinements*, Boulbaba Ben Ammar, Mohamed Tahar Bhiri and Jeanine Souquières propose four kinds of operation refinement in order to control modifications when developing and refactoring UML specifications. Each refinement has been described with its properties and illustrated by an example, showing which verifications can be done, using the B formal method.

In *A Formal and Sound Transformation from Focal to UML—An Application to Airport Security Regulations*, David Delahaye, Jean-Frédéric Etienne and Véronique Viguié Donzeau-Gouge propose an automatic transformation of Focal specifications to UML class diagrams. The main motivation for this work lies within the framework of the EDEMOI project, which aims to integrate and apply several requirements engineering and formal methods techniques to analyze airport security regulations.

## 4 Session 4: Formal behavior

In *Ambiguity and Structural Properties of Basic Sequence Diagrams*, C. Sibertin-Blanc, N. Hameurlain and O. Tahir study four semantics for basic sequence diagrams. They show to what extent their differences for a given sequence diagram (that is the amount of ambiguity of this diagram) comes from its structural properties (linearity, local control and local causality). The fulfilment of these properties can serve as a measure of the ambiguity of a sequence diagram, and thus the attention to be paid at its validation.

In *Extending Statecharts with Process Algebra Operators*, Marc Frappier, Frédéric Gervais, Régine Laleau, Benoît Fraikin and Richard-St-Denis describe an adaptation of statecharts to take advantage of process algebra operators like those found in CSP and $EB^3$. The resulting notation is called algebraic state transition diagram (ASTDs).

In *UML Behavioral Consistency Checking using Instantiable Petri Nets*, Yann Thierry-Mieg and Lom-Messan Hillah propose an approach based on translation of a UML model to Instantiable Petri nets (IPN). This formalism is based on the semantics of Petri nets, but introduces the concepts of type and instance. The approach has been implemented and experimental results have been presented.

In *Timing Analysis and Validation with UML: The Case of the Embedded MARS Bus Manager*, Iulian Ober, Susanne Graf, Yuri Yushtein and Ileana Ober present a case study in UML-based modeling and validation of the intricate timing aspects arising in a small but complex component of the airborne Medium Altitude Reconaissance System produced by the Netherlands National Aerospace Laboratory.

In *Clock Constraint Specification Language Specifying Clock Constraints with UML/MARTE*, Frédéric Mallet gives a brief overview of the CCSL syntax as well as its formal semantics, showing how existing UML model elements can be used to apply this syntax in a graphical way and benefit from the semantics. The non-normative concrete syntax called CCSL has been defined in the context of the OMG UML profile for modeling and analysis of real-time and embedded systems (MARTE).

## 5 Panel

A set of questions and answers have served as a conclusive discussion through a panel animated by the organizers. The panelists were Jean Bézivin, Jean-Michel Bruel, Sébastien Gérard and Frédéric Mallet. A summary of the discussions is available on the workshop web site mentioned earlier.

**Acknowledgments** The organizers would like to thank Mike Hinchey for his support and his advices for the successful organization of this workshop.