Editorial

Model aesthetics

Robert France, Bernhard Rumpe

We thank the readers that commented on our editorial on model quality in issue 2004-3, and we strongly encourage readers to send comments to us on the editorials and the papers published in SoSyM (our email addresses are included at the end of this editorial). Some comments pointed out that the study of model quality addresses a wide-ranging set of questions and concerns. Model quality is not only concerned with how faithfully a model describes desired properties of the real world or system; it should also be concerned with innate attributes that affect qualities such as analyzability, understandability, and evolvability: Is the model readable? Is the model ambiguous? Is the model concise and complete? Is the model unnecessarily redundant? It may be possible to identify "model smells" (similar to "code smells") that provide indicators of a model's quality. We encourage researchers in this area to submit high quality papers on this topic to SoSyM. Papers that describe the results of empirical studies on model quality and aesthetics are especially encouraged.

In a previous editorial, we discussed a new and emerging discipline called "Model Engineering" (see Editorial of Sosym Issue 2003-2). The discussion was based on a vision of model engineering held by one of the leading experts in the area, *Jean Bézivin*. In this issue, we are very pleased to present an expert paper by Jean on model engineering. This issue also contains a special section on model-based tool integration, edited by Andy Schürr and Heiko Dörr. The two editors have done an outstanding job of selecting high quality papers that reflect the state of the research and practice in this area.

Contents in this issue

This issue contains a special section on "**Model-based tool integration**" by *Andy Schürr*, Darmstadt University of Technology and *Heiko Dörr*, DaimlerChrysler Research Berlin. In the context of effective use of models, integrated tool support is critical. Andy and Heiko have put together a very good set of papers discussing new approaches to the integration of model-based tools. Following this editorial, there is an introduction to the special section that describes the four papers in the section.

The second part of this issue contains an expert voice and three regular papers. In the expert voice "**On the unification power of models**", *Jean Bézivin* describes the beneficial impact a model driven engineering (MDE) perspective can have on software development. In the paper, Jean examines how software was developed in the past and how it is developed currently. He then extrapolates into the future and proposes that the concept of "model" will become a paradigm as prevalent as the current object paradigm.

The regular paper "**Problem frame semantics for software development**" by *Jon G. Hall, Lucia Rapanotti* and *Michael Jackson* is based on Michael's seminal book on problem frames. It describes their relationship to an appropriate requirements engineering model and provides a language for describing problem frames, thus opening the way for tool assistance when using problem frames.

In the regular paper "**Modeling Multi-agent systems with ANote**" *Ricardo Choren* and *Carlos Lucena* describe a model-based approach targeting the development of agent systems. The authors use a domain-specific modeling language (DSL) called *Anote* in a manner that is consistent with the usage we described in the Editorial of issue 2005-01. *ANote* covers multiple views: a structure, a state and an interaction oriented view, but uses specific elements from their domain of describing problem frames. *ANote* is an UML-based language for describing agent systems that is expressive yet cleaner and more

lightweight than the full UML. This approach demonstrates how new UML-based DSL's can be defined and applied.

The third regular paper "**PSL: A semantic domain for flow models**" by *Conrad Bock* and *Michael Gruninger* describe a process specification language (PSL) and associated techniques for constraining and analyzing possible chains and parallel structures of activities. The graphical design of workflows is thus supported by techniques that allow the definition of constraints on activity order and activity selection.

We hope you enjoy reading the articles in this issue.

Robert France <france@CS.ColoState.EDU>
Bernhard Rumpe <Bernhard.Rumpe@sosym.org>
Editors in Chief