

## *Editorial*

# First Issue of the International Journal on Software and Systems Modeling

Published online: 12 September 2002 – © Springer-Verlag 2002

Welcome to the first issue of the Software and Systems Modeling (SoSyM) journal. The idea of launching a journal that publishes quality scientific works on modeling of software and software-based systems was planted in 2000 during a steering committee meeting of the International Conference on the Unified Modeling Language. It has been our pleasure to see the idea blossom into a first issue that contains scientific work that should be relevant to practicing system developers and to researchers. In this editorial we outline the intended purpose, scope and objectives of the journal.

### 1 Aims and Scope of SoSyM

SoSyM is a quarterly journal that focuses on theoretical and practical aspects of software and system modeling languages, methods and techniques. The aim of the journal is to publish high-quality work in these areas. Of particular interest are papers that investigate theoretical underpinnings of modeling languages and model-based analysis and testing techniques, rigorously analyze modeling experiences, present the results of experiments concerned with the validation of modeling techniques and notations, and present scalable modeling techniques and methods that facilitate rigorous and economical development of software. The journal targets researchers, system and software developers, and students that have a vested interest in results generated by high-quality research into model-based development techniques.

Recent interest in modeling notations and techniques has resulted in a rapidly growing body of research work that clearly can benefit from a journal that focuses on system and software modeling. The journal will be unique in its emphasis on research results that can have a significant and immediate impact on the current state of the practice, and research that lays

firm foundations for the development of more sophisticated model-based development techniques. The aim is to provide researchers as well as tool vendors and standardization committees with insights that can lead to better modeling languages and techniques, and provide software and system developers with a deeper understanding of modeling languages and techniques that can lead to more effective application. The composition of the editorial board reflects the intent that papers published in the journal appeal to developers in industry and government agencies, and to researchers and educators.

The journal's title reflects the intent to include not only papers that focus on software modeling, but also papers that take a system view of software development. Software is often developed in the context of larger encompassing systems. An encompassing system could be a business system in which the software automates some aspects of a work-flow, or an embedded system in which software interfaces with hardware. Modeling relevant aspects of an encompassing system and its relationships with the software allows one to analyze and understand behaviors that emerge as a result of the interactions between the software and its environment. Papers that discuss how concepts from models of non-software based systems (e.g. economic and living systems) can be used to enhance modeling of software systems are also within the scope of SoSyM.

The types of modeling notations and methods that are within the scope of the journal are not restricted. Authors are strongly encouraged to submit high quality work pertaining to UML, non-UML modeling and specification notations (e.g., B, Larch, LOTOS, Maude, MSCs, Petri-Nets, SDL, Z), and functional and other non-OO software and system modeling techniques and methods. In particular, the UML standardization community stands to gain through exposure to high-quality concepts found in

other modeling languages. Such exposure can facilitate the building of bridges from UML to other modeling languages; an important aspect of the current Object Management Group's (OMG) initiative called Model-Driven Architecture (MDA).

## 2 The Future of Modeling

There is a growing realization that models can play a critical role in the development of large complex software-based systems. While other approaches have advocated this for quite some time, a recent key indication of this is the MDA (Model-Driven Architecture) initiative of the OMG (Object Management Group). The primary aim of the MDA is to make models the primary artifacts of system development, thus raising the level of abstraction at which developers construct complex integrated systems. The MDA is concerned with modeling all aspects of software development. In an MDA approach, models can range from models of business functions and entities, to detailed models of operational systems. Included are models of (1) enterprise system architectures, (2) the infrastructure supporting integrated systems (e.g., middleware, database management systems, security and communication protocols), (3) applications and data, and (3) management concerns (e.g., financial and risk models). While the MDA emphasizes use of the UML (as a family of languages), there is acknowledgement that MDA technologies need to support the use of other modeling notations as well. Key aspects of the MDA are (1) separation of platform independent concerns from platform specific concerns, and (2) precisely defined relationships across models (including source code) that help developers understand the interplay among different aspects of software and system development. MDA can be viewed as an attempt to standardize the way software-based systems are currently developed. In particular, MDA provides a framework for the development of sophisticated, modeling tools (including domain-specific tools) that can be used to map abstract models into code, derive test suites, and to rigorously analyze modeled properties (e.g., type checkers, dataflow analysis tools, and model checking tools).

The diversity of the software to be developed today and in the future makes it unlikely that a single approach to software development will satisfy all development needs. We expect that they will continue to be a plethora of sometimes competing software development techniques and methods, based on a variety of paradigms. The challenge to software engineers is to develop an understanding of the many development techniques and methods so that they can determine the situations in which use of particular approaches are appropriate. From this perspective, software methods and techniques are part of a software engineer's toolbox, or development

portfolio. This journal aims to facilitate continual improvement of modeling techniques and methods by providing a forum in which high-quality work on modeling can be disseminated.

In looking ahead at the future of software modeling it is worthwhile to look at the past for inspiration. Here, we take our inspiration from the work of Copernicus. Copernicus developed a new heliocentric model that, when conceived, was less accurate than the old geocentric model that was based on a very refined system of irregular circles. Despite its shortcoming, the new model was easier to understand, better motivated, and thus more attractive to use. Insights gained through the usage of the Copernicus model paved the way for developments that helped improve the model's accuracy over time. In order to continually improve modeling notations and methods, it is necessary to voice new ideas and challenge the so-called "common understanding". Unlike Copernicus, software methodologists have to cope with multiple, sometimes competing, prevailing paradigms. This can be a source of frustration for some, and inspiration for others. Continual improvement of software development techniques and methods is possible when communities of methodologists encourage and exploit open discussions and constructive criticisms of prevailing software development paradigms.

## 3 Topic Areas

We invite authors to submit papers that discuss and analyze concerns and experiences pertaining to software and system modeling languages, techniques, tools, practices, principles and other facets. This includes domain independent as well as domain specific techniques. The following are some of the topic areas that are of special interest:

- Methodological issues
- Development of modeling standards
- Formal syntax and semantics of modeling languages
- Rigorous model-based analysis
- Model composition and transformation
- Relationships between models
- Relationships between models, code and environment
- Refinement and abstraction in modeling
- Metamodeling techniques
- Measuring quality of models
- Modeling support for aspect-oriented development
- Ontological approaches to model engineering
- Domain specific modeling
- Generating test and code artifacts from models
- Modeling tests
- Model refactoring
- Model development tool environments
- Case studies and experience reports with significant lessons learned
- Comparative analyses of modeling languages and techniques
- Scientific assessment of modeling practices

## 4 Summary of articles in this issue

A journal issue will have two main sections:

- **Experts Voice:** This section will consist of relatively short papers from authors who have developed modeling expertise in broad areas of interest. The intent is to present readers with perspectives of modeling research and practices that are based on high-quality research and analytical experiences. In this issue we start with an article from a noted methodologist, Michael Jackson. In his paper, Michael Jackson outlines his view on the basic tenets of description notations.
- **Regular Section:** This section will include scientific papers on system and software modeling. Included in this issue are three papers.
  - On the interpretation of binary associations in the Unified Modelling Language by Perdita Stevens. This paper critiques the UML notion of a fundamental object-oriented modeling concept: association. The paper provides suggestions for improving and clarifying the notion of association in the UML.
  - Ontological evaluation of the UML using the Bunge-Wand-Weber model by Andreas L. Opdahl and Brian Henderson-Sellers. In this paper an ontological model of information systems, the Bunge-Wand-Weber (BWW) model, is used to analyze the Unified Modeling Language(UML) as a language for representing problem domains. The analysis reveals overlapping, vaguely defined and spurious UML-constructs, as well as some areas where UML does not presently offer adequate support.
  - A UML-based approach to system testing by Lionel Briand and Yvan Labiche. In this paper a UML-based system testing technique is presented. The goal of the work is to support the derivation of functional system test requirements from UML behavioral models, that are used to produce test cases, test oracles, and test drivers once detailed design information becomes available.

Future issues will also include a section that includes information on modeling conferences and workshops, and reviews of books and professional meetings. If you can provide information for this planned section please have a look on our web site [www.SoSVM.org](http://www.SoSVM.org) for contact information.

## 5 SoSyM mailing group:

<http://groups.yahoo.com/group/sosym-announcement/>

A low-traffic mailing group for SoSyM announcements has been created on the internet. Anyone can subscribe,

but only the SoSyM Editors are allowed to post to the group. If you are interested in receiving SoSyM-related news you may subscribe at <http://groups.yahoo.com/group/sosym-announcement/>

## 6 Submission and review process

### 6.1 Submission process

Submitted papers should be prepared with a font size no smaller than 11pt, and with 1.5 line spacing. The number of pages is not restricted. An abstract, keywords and appropriate literature list, as well as an extra summary, are required. Other than that, no particular submission format is required; documents can be prepared in Word or Latex. Papers submitted to SoSyM must not be simultaneously submitted as is, in an extended or in a shortened form to other journals or conferences. Authors, however, can submit extended forms of papers that have previously appeared in conference proceedings. Such submissions must clearly state that the paper has been published elsewhere, must reference the paper in the submission, and must clearly state how the paper significantly extends the published version.

Final typesetting is done by Springer. Publishing policy is first come, first served. Reasons for fast-track papers may exist. There will also be special sections that include the enhanced versions of the best papers from previously published conferences. These papers will be reviewed thoroughly before publication decisions are made.

For further details and updated information on the submission process see our website <http://www.sosym.org/>

### 6.2 Reviewer Selection

Upon the receipt of manuscript for consideration, the editor-in-chief either directly select minimum of 4 reviewers to review the submitted manuscript or assign a supervisor from the international Editor Board, who then selects the reviewers. Assignment of reviewers is based on expertise in the field as well as reviewer assignment load.

### 6.3 Review Process

All submissions for SoSyM will be received electronically. Upon the acknowledgement of the receipt of the manuscript by the authors, a copy of the manuscript along with guidelines for the review and a review form are sent to each selected reviewer for the manuscript. The reviewers names will not be available to the authors.

Please note that the editors-in-chief will make every effort to ensure a timely (8 week) review process. However, due to traveling, holidays, and most likely the usual workload of the experts that our reviewers usually are, it is not always feasible.

#### 6.4 Reviewers Responsibilities

Reviewers are expected to return their detailed and accurate reviews to the editor-in-chief electronically within about 6 weeks. Collected reviews are given to the supervisor who then decides whether to (1) reject, (2) accept with major revision and a re-review, (3) accept with minor revision, (4) accept without change or (5) force additional reviews. Reviews should be extensive and constructive, but it is not the responsibility of the reviewer to enhance language issues, provide missing literature, etc.

Reviewers may not have stronger contacts to any of the authors, e.g. having worked in the same project, published together or in the same university recently.

### 7 Acknowledgements

We thank the editorial board for their indispensable advice and support during the planning phases of the journal. We also thank the reviewers who contributed their valuable time to complete reviews in a timely manner,

and those who have expressed their willingness to review future submissions. The success of a journal hinges on the quality of its editorial board and the reviewers. The quality of effort that we have seen from our editorial board and reviewers bodes well for the future of the journal. Once a year a list of reviewers will be published.

A number of individuals have provided significant help and support in producing this first issue. We thank Pierre-Alain Muller and Jean Bezivin, the members of the UML conference steering who proposed the idea of launching a journal and supported our efforts in getting it started. Special thanks to Geri Georg for her indispensable help in emaning the review process. Thanks also to Roger Burkhart, Anita Bürk, Ria Frauenfeld, Dorothea Glaunsinger, Elke Janosch, Kathleen Hiller. We especially thank Hermann Engesser from the Springer Verlag and Wayne Yuhasz from Springer New York; without their support this journal would not have been possible.

Sincerely,  
Robert France, Bernhard Rumpe  
SoSyM Editors-In-Chief

Robert France  
Computer Science Department  
Colorado State University  
Fort Collins, Colorado, USA  
france@cs.colostate.edu

Bernhard Rumpe  
Software & Systems Engineering  
Munich University of Technology  
Munich, Germany  
rumpe@in.tum.de

### Editorial board

Scott Ambler (Canada)  
Egidio Astesiano (Italy)  
Jean Bezivin (France)  
Jim Bieman (USA)  
Grady Booch (USA)  
Lionel Briand (Canada)  
Ed Brinksma (Netherlands)  
Manfred Broy (Germany)  
Jean-Michel Bruel (France)  
Jaelson Castro (Brazil)  
Betty Cheng (USA)  
Tony Clark (UK)  
Steve Cook (UK)  
Desmond D'Souza (USA)

Andy Evans (UK)  
Gregor Engels (Germany)  
David Garlan (US)  
Martin Gogolla (Germany)  
David Harel (Israel)  
Constance Heitmeyer (USA)  
Brian Henderson-Sellers  
(Australia)  
Heinrich Hussmann (Germany)  
Michael Jackson (UK)  
Ivar Jacobson (USA)  
Jean-Marc Jezequel (France)  
Stuart Kent (UK)  
Cris Kobryn (USA)

Ingolf Krger (USA)  
Kevin Lano (UK)  
Gary Leavens (USA)  
Pierre-Alain Muller (France)  
Jeff Offutt (USA)  
Wolfgang Reisig (Germany)  
August Scheer (Germany)  
Bran Selic (Canada)  
Perdita Stevens (UK)  
Roel Wieringa (Netherlands)  
Alan Wills (UK)  
Jos Warmer (Netherlands)  
Jim Woodcock (UK)