# 5th International Workshop on Modeling in Software Engineering (MiSE 2013)

Joanne M. Atlee
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
jmatlee@uwaterloo.ca

Robert Baillargeon
Sodius
East Amherst, NY, USA
rbaillargeon@sodius.com

Marsha Chechik
Department of Computer Science
University of Toronto
Toronto, ON, Canada
chechik@cs.utoronto.ca

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

Jeff Gray
Department of Computer Science
University of Alabama
Tuscaloosa, Alabama, USA
gray@cs.ua.edu

Richard F. Paige
Department of Computer Science
University of York
York, UK
richard.paige@york.ac.uk

Bernhard Rumpe
Department of Computer Science
RWTH Aachen University
Aachen, Germany
b.rumpe@se-rwth.de

***Abstract*—Models are an important tool in conquering the increasing complexity of modern software systems. Key industries are strategically directing their development environments towards more extensive use of modeling techniques. This workshop sought to understand, through critical analysis, the current and future uses of models in the engineering of software-intensive systems. The MISE-workshop series has proven to be an effective forum for discussing modeling techniques from the MDD and the software engineering perspectives. An important goal of this workshop was to foster exchange between these two communities. The 2013 Modeling in Software Engineering (MiSE) workshop was held at ICSE 2013 in San Francisco, California, during May 18-19, 2013. The focus this year was analysis of successful applications of modeling techniques in specific application domains to determine how experiences can be carried over to other domains. Details are available at: *https://sselab.de/lab2/public/wiki/MiSE/index.php***

***Index Terms*—Modeling languages, model-driven engineering, model transformations, model management**

## I. Workshop Overview

The goals of the software modeling community are to improve the productivity of software developers and to improve the quality of the resulting software products. Whereas in the past, software models might have been viewed primarily as documentation artifacts, the focus of the MiSE workshop considers the use of models to facilitate software development. Models can be used in all phases and activities surrounding software development and deployment: comprehension of domain, requirements, or design alternatives; analysis of designs or quality attributes; detection of interactions among features or components; reuse of requirements, components, frameworks; automatic generation of software implementations or test suites; or run-time abstractions of the program state. Software models can be highly abstract and informal (e.g., depictions of early design proposals) or can be precise enough to enable mathematical inference, reasoning, or even executability. A common theme among all of these uses and representations is the notion of *abstraction* – a software model omits certain details in order to focus attention on other details.

## II. Workshop Description

The purpose of this 2-day workshop was to study and advance the effective use of models in the engineering of software systems. In particular, the workshop provided an opportunity for the exchange of experiences and innovative technical ideas related to modeling. Engineers have used models to manage complexity for centuries, and there is a growing body of work on the use of models to manage inherent problem and solution complexity in software development. The use of software models will become more prevalent as methodologies and tools that manipulate them at various levels of abstraction become available. A secondary goal of the workshop was to further promote cross-fertilization of ideas, techniques and tools between the modeling and software engineering communities.

The workshop considered a broad view of what models are and how they are used, including:

- **Exploration**: Models are used to explore and learn about the problem to be solved, where the "problem" can be, for example, requirements identification, system specification, system or component design, complex protocol or algorithm design.
- **Communication**: Communication models are used to document software decisions (e.g., requirements, designs,

978-1-4673-3076-3/13/$31.00 © 2013 IEEE



[ABC+13] J. M. Atlee, R. Baillargeon, M. Chechik, R. F. France, J. Gray, R. F. Paige, B. Rumpe]
5th International Workshop on Modeling in Software Engineering (MiSE 2013),
Proceedings of the 35th International Conference on Software Engineering, May 18-26, 2013, San Francisco,
pages 1531-1532, IEEE Press, Piscataway, USA, 2013
www.se-rwth.de/publications

and deployment decisions). Support for downstream activities: We use software models to answer questions or check properties (e.g., correctness, fitness for use) of the modeled artifact, to generate other artifacts, or to configure existing systems.

- **Support for downstream activities**: Software models are often used to answer questions or check properties (e.g., correctness, fitness for use) of the modeled artifact, to generate other artifacts, or to configure existing systems.
- **Configurability and adaptation**: Models are used at runtime to configure the system, to adapt it to changed needs of the users. A model of the environment also allows a system to capture its knowledge about the environment it controls or communicates with.

Workshop activities focused on analyzing successful and unsuccessful applications of software modeling techniques to gain insights into challenging modeling problems, including: (1) identifying, describing, and using appropriate abstractions, (2) supporting incremental, iterative development through the use of appropriate model composition, transformation and other model manipulation operators, and (3) automated analysis of possibly large or incomplete models to determine the presence or absence of desired and undesired properties.

## III. Workshop Structure

The MiSE workshop was held over two days. The workshop was highly interactive and focused on the sharing of ideas towards a shared vision of research goals. Included in the program were paper presentations, open discussions, a panel discussion, posters, and a keynote talk. Paper presentations were grouped into five thematically related sessions of three 20-minute presentations, with 30 minutes of discussion at the end of each session. Half of the second day was spent in working groups based on topics collected during paper discussions.

## IV. Selected Papers for Presentation

The workshop received 23 submissions, with each submission reviewed by three members of the Program Committee. After the review and discussion of all submissions, 11 were accepted as Full papers, 3 as Short papers, and 3 as posters. The following are the accepted papers across five categories.

*Model Representation*

- Lionel Montrieux, Yijun Yu, Michel Wermelinger and Zhenjiang Hu, "Issues in Representing Domain-Specific Concerns in Model-Driven Engineering"
- Michalis Famelis and Stephanie Santosa, "MAV-Vis: a Notation for Model Uncertainty"
- Anitha Murugesan, Sanjai Rayadurgam and Mats Heimdahl, "Modes,Features, and State-Based Modeling for Clarity and Flexibility"

*Model Analysis and Development Support*

- Alfredo Motta, Nicolas Mangano and André van der Hoek, "Light-weight Analysis of Software Design Models at the Whiteboard"
- Sebastian Herold and Andreas Rausch, "Complementing Model-Driven Development for the Detection of Software Architecture Erosion"
- Matthieu Foucault, Sébastien Barbier and David Lugato, "Enhancing Version Control with Domain-Specific Semantics"

*Applications of Modeling*

- Marco Miglierina, Giovanni Paolo Gibilisco, Danilo Ardagna and Elisabetta Di Nitto, "Model Based Control for Multi-Cloud Applications"
- Naoyasu Ubayashi and Yasutaka Kamei, "Design Module: A Modularity Vision Beyond Code -Not Only Program Code But Also a Design Model Is a Module"
- Robert Pettit and Navneet Mezcciani, "Highlighting the Challenges of Model-Based Engineering for Spaceflight Software Systems"

*Runtime Models*

- Eladio Domínguez Murillo, Beatriz Pérez and María A. Zapata, "A UML Profile for Dynamic Execution Persistence with Monitoring Purposes"
- Alexander Borgida, Fabiano Dalpiaz, Jennifer Horkoff and John Mylopoulos, "Requirements Models for Design- and Runtime: A Position Paper"

*Short Papers*

- Omar Badreddin and Timothy C. Lethbridge, "Model Oriented Programming: Bridging the Code-Model Divide"
- Federico Tomassetti, Antonio Vetro', Marco Torchiano, Markus Voelter and Bernd Kolb, "A model-based approach to language integration"
- Everton Guimaraes, Alessandro Garcia, Eduardo Figueiredo and Yuanfang Cai, "Prioritizing Software Anomalies with Software Metrics and Architecture Blueprints: A Controlled Experiment"

## V. Program Committee and External Reviewers

The Program Committee was composed of a diverse group of members from nine different countries. There were 70 reviews provided for the 23 submissions that were received. We thank the following members of the MiSE 2013 Program Committee for their assistance in reviewing and discussing the papers that were selected for inclusion in the final program:

- Lionel Briand, University of Luxembourg, Luxembourg
- Manfred Broy, Technical University of Munich, Germany
- Krzysztof Czarnecki, University of Waterloo, Canada
- Juergen Dingel, Queen's University, Canada
- Geri Georg, Colorado State University, USA
- Mats Heimdahl, University of Minnesota, USA
- Michael Jackson, The Open University, UK
- Jean-Marc Jezequel, INRIA & Univ. Rennes 1, France
- Dimitris Kolovos, University of York, UK
- Jeff Kramer, Imperial College London, UK
- Ana Moreira, Universidade Nova Lisboa, Portugal
- Alfonso Pierantonio, University of L'Aquila, Italy
- Awais Rashid, Lancaster University, UK
- Andrey Sadovykh, Softeam, France
- Sebastian Uchitel, Universidad de Buenos Aires, Argentina
- Steffen Zschaler, Kings College, UK

We also thank the following external reviewers for their supplemental reviews:

Kacper Bak, Jonathan Corley, Jianmei Guo, Ludovico Iovino, Antoaneta Kondeva, Christian Leuxner, Klaus Mueller, Shiva Nejati, Mehrdad Sabetzadeh, Martin Schindler