

Modeling in Software Engineering

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

Ana Moreira
Computer Science Department
Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
2829-516 Caparica, Portugal
amm@di.fct.unl.pt

Robert France, Geri Georg
Department of Computer Science
Colorado State University
Fort Collins, Colorado
{france, georg}@cs.colostate.edu

Bernhard Rumpe
Software Systems Engineering
Faculty for Computer Science
Mhlenpfordtstr. 23
38106 Braunschweig, Germany
b.rumpe@tu-bs.de

Steffen Zschaler
Institute for Software and Multimedia Technology
Faculty for Computer Science
Technische Universität Dresden
D-01062 Dresden, Germany
steffen.zschaler@inf.tu-dresden.de

1. Workshop Overview

The software modeling community is primarily concerned with reducing the gap between problem and software implementation through the use of models that describe complex systems at multiple levels of abstraction and from a variety of perspectives. A *model* is an abstraction of some aspect of an existing or planned system. Models are created to serve particular purposes, for example, to present a human-understandable description of some aspect of a system or to present information in a form that can be mechanically analyzed.

There is a perception that software models are primarily documentation artifacts and thus their creation and use are peripheral to software development. This narrow perspective fails to recognize that software development involves both explicit and mental models of both problems and solutions – that even software implementations are models, expressed in terms of abstractions provided by programming languages. From this perspective, software development is a modeling activity and thus there is a need for research that focuses on how to better leverage modeling techniques during software development. The presentations and discussions in this workshop will focus on the challenge of sup-

porting the use of models in all phases of the software life-cycle, including the use of models at runtime. Discussions will focus on innovative technical ideas and experience related to identifying and evolving appropriate abstractions, manipulating and managing software models, and teaching and learning modeling techniques.

Software-modeling research evolves and integrates research results from a number of areas in computer science, including languages and semantics, visualization, translation and transformation, and analysis and reasoning. Applied to software engineering, it addresses problems in requirements, architecture, and detailed design modeling, including work on viewpoint conflict analysis and on feature interaction analysis. It encompasses the realization of *model-driven engineering (MDE)*, whose vision of software development centers on systematic transformations of problem-level models to software implementations.

One of the objectives of this workshop is to build bridges to other software engineering research communities that facilitate collaborative research on software modeling problems. For example, work on formal specification techniques is particularly relevant. Modeling languages must have formally defined semantics if they are to be used to create analyzable models. Work on developing formal analysis tech-

niques for models utilizes and builds on work in the formal specification research area. While it is currently the case that popular modeling languages have poorly defined semantics, there is a growing realization that MDE requires semantic-based manipulation of models and thus appropriate aspects of modeling languages must be formalized. From the other perspective, MDE can provide a valuable context for formal-methods research, and can challenge the formal-methods community to accommodate rich modeling features and languages.

The need to deal with the complexity of developing and operating adaptive software provides another opportunity for the use of software-modeling techniques. This workshop will explore how the scope of software modeling can be broadened to encompass research on the use of models during the execution of software.

2. Workshop Topics and Objectives

This workshop will provide a forum for discussing the challenges associated with modeling software, with incorporating modeling practices into the software development process, and with teaching and learning modeling techniques. The following are the key categories of technical challenges that will be discussed:

- **Abstraction challenges:** These challenges arise from concerns associated with providing support for creating and using problem-level abstractions in modeling languages, and for rigorously analyzing models. Problems associated with modeling systems using multiple, overlapping viewpoints that utilize possibly heterogeneous languages also fall into this category. Topics include domain-specific modeling, supporting separation and integration of concerns, modeling of non-functional properties, modeling patterns and strategies, and analyzing models.
- **Model manipulation and management challenges:** These challenges arise from problems associated with (1) defining, analyzing, and using model transformations, (2) maintaining traceability links among model elements to support model evolution and roundtrip engineering, (3) maintaining consistency among viewpoints, (4) tracking versions, and (5) using models during runtime.

During the workshop key challenges will be identified, modeling experience analyzed, and promising ideas explored. The aim is to develop a roadmap that can be used to guide software-modeling research. The workshop will also provide a forum in which researchers from the ICSE and MODELS communities share experience and learn more about each others' perspectives on modeling research.

3. Workshop Structure

The workshop will be held over one and a half days. The first day will be a half day event in which papers are presented. The second full day will consist of a half day of presentations and a half day of informal discussion panels in which lead discussants will stimulate discussions in three topic areas:

- Abstraction challenges
- Model manipulation and management challenges
- Education issues

In order to foster lively discussions, each paper presentation will be structured as follows:

1. A 15-minute presentation by an author of the paper, summarising the key points in the paper.
2. A 5-minute counter-presentation by a participant of the workshop. This presentation summarises the paper from a different view point and exposes interesting issues that can form the basis of a discussion.
3. A 10-minute slot for open discussions related or stimulated by the paper presented.

Paper presentations will be structured into sessions each grouping three thematically related papers.

The lead discussants in the panels will stimulate discussion by presenting their views on research in the respective areas and raising key research questions and challenges. In the closing session of the workshop, the lead discussants will summarize the discussions in terms of the research questions and challenges that are considered to be critical to further development of software-modeling knowledge and technologies.