Introduction

Computational modeling has become the norm in industry to remain competitive and be successful. As such, Model-Based Design of embedded software has enterprise-wise implications and modeling is not limited to isolated uses by a single engineer or team. Instead, it has reached a proliferation much akin to large software design, with requirements for infrastructure support such as version control, configuration management, and automated processing.

The comprehensive use of models in design has created a set of challenges beyond that of supporting one isolated design task. In particular, the need to combine, couple, and integrate models at different levels of abstraction and in different formalisms is posing a set of specific problems that the field of Computer Automated Multiparadigm Modeling (CAMPaM) is aiming to address. The essential element of multi-paradigm modeling is the use of explicit models throughout. This leads to a framework with models to represent the syntax of formalisms used for modeling, models of the transformations that represent the operational semantics, as well as model-to-model transformations for inter-formalism transformation. These models are then used to facilitate generative tasks in language engineering, such as evolving a domain specific modeling formalism as its requirements change, but also in a tool engineering space, such as automatic generation of integrated development environments. Moreover, an explicit model of a model transformation allows analyses such as termination characteristics, consistency, and determinism.

Topics of interest include but are not limited to:

- **Language Engineering:**
  - Concepts for multiple paradigm composition.
  - Existing multi-paradigm approaches
    - Analysis.
- Overview.
  - Composability.
    - Generic reasoning techniques.
    - Scalability of compositions.
- Tool Engineering:
  - Concepts for tool support of multi-paradigm modeling.
  - Tool support for compositional language constructs.
  - Multi-paradigm modeling tools.
  - Tool Interoperability.
- Applications:
  - Modeling concrete and abstract syntax and their relation.
  - Integration of visual and textual modeling techniques.
  - Executability and support for code generation.
  - Model transformation modeling:
    - Modularity.
    - Textual or visual specification.
    - Declarative vs. imperative approaches.
    - Execution control.
    - Verification of properties.
  - Traceability.
  - Model composition in different formalisms at the syntax and semantics level

Contributions should clearly address the foundations of multiparadigm modeling by demonstrating the use of models to achieve the stated objectives and discuss the benefits of explicit modeling.

Instructions for manuscript preparation

For manuscript formatting and other guidelines, please visit the Author Guidelines for SIMULATION page. Manuscripts should be prepared and submitted online at http://mc.manuscriptcentral.com/simulation. Please note in your online cover letter the special issue for which your submission is intended. For this special issue, we’ll invite for consideration extended and revised versions of the best papers of the 2nd International workshop "Multi-Paradigm Modeling: Concepts and Tools". However, the call for papers is open to additional papers, thus we specially encourage authors that were not able to attend the workshop to submit their work in related subjects.

Note: Manuscripts must not have been previously published or be submitted for publication elsewhere. Each submitted manuscript must include title, names, authors’ affiliations, postal and e-mail addresses, an extended paper, and a list of keywords. For multiple author submission, please identify the corresponding author.

Due Dates
Submission of Papers    April 30, 2008
Notification of Acceptance    June 15, 2008
Final camera-ready manuscript due    July 15, 2008
Expected date of publication    Winter 2008

For questions contact the guest editors at:

Dr. Pieter J. Mosterman
The MathWorks, Inc.,
Natick, MA 01760 (USA)

Dr. Tihamer Levendovszky
Dep. of Automation and Applied Informatics
Budapest (Hungary)

Dr. Juan de Lara
School of Computer Science
Universidad Autónoma de Madrid (Spain)