Model-Driven Engineering: Raising the Abstraction Level through Domain-Specific Modeling

Jeff Gray
University of Alabama
Department of Computer Science
Tuscaloosa, AL 35487
gray@cs.ua.edu

Jules White and Aniruddha Gokhale
Vanderbilt University
Institute for Software Integrated Systems
Nashville, TN 37235
{jules, gokhale}@dre.vanderbilt.edu

ABSTRACT
Model-Driven Engineering (MDE) has emerged as a promising paradigm in software engineering by emphasizing the use of models not just for documentation and communication purposes, but as first-class artifacts to be transformed into other work products (e.g., other models, source code, and test scripts). MDE supports full-scale round-trip engineering, from idea inception to operationalization. Historically, models have been developed using general-purpose modeling languages, such as the Unified Modeling Language (UML). A more recent trend is to use domain-specific modeling languages (DSMLs), which assist domain experts in working within their own problem space without being concerned about technical details of the solution space (e.g., programming languages and middleware). DSMLs also provide an accessible way to communicate with stakeholders who are not familiar with the fast changing technologies. This introductory tutorial will present a summary of the areas represented by MDE and offer some insight into the benefits of using DSMLs in both research and teaching.

1. STATEMENT OF PURPOSE
The purpose of this tutorial is to present an introductory survey of topics related to model-driven engineering. In particular, the tutorial will offer a context for understanding the benefits that can be achieved by using domain-specific modeling languages. The tutorial will describe the core principles of MDE, give attendees a hands-on experience using a tool that represents the area, and present case studies showing the advantages of adopting MDE.

The tutorial will be contained in three sessions, represented by the following activities:

- Introduction and Motivational Background: The first section of the tutorial will introduce attendees to the core concepts and foundational principles of MDE and DSMLs (e.g., metamodeling and model transformation). This session will define the terms used in this area, motivate the benefits that can be realized in adopting MDE, and summarize existing application and domain areas where the concepts have been applied successfully.
- Hands-on Exercise with a Metamodeling Tool: This tutorial will be participatory in nature. Attendees will be provided with, and encouraged to install, a metamodeling tool that represents the area. This part of the session will walk through the creation of a simple domain-specific modeling language, which will contain enough detail to provide the attendees with experience that extends their understanding of the ideas introduced in the first session.
- Case Studies: The final session of this tutorial will demonstrate the design, implementation, and usage of domain-specific modeling languages that have been created for both research and education purposes.

2. OBJECTIVES
The objectives of this tutorial are to raise awareness of MDE techniques and to provide attendees with knowledge of the following concepts: an understanding of the benefits and advantages that motivate the use of MDE, concepts of metamodeling and model transformation (e.g., model refinement and also code generation from models), survey of metamodeling tools available, hands-on investigation of tutorial topics (e.g., defining a simple modeling language with the tutorial speakers).

3. INFORMATION FOR ATTENDEES
This tutorial is designed to be introductory in nature. Attendees should have a core understanding of object-oriented concepts; very basic exposure to UML class diagrams is desirable, but not required. The tutorial is not intended as an advanced discourse on the topic, but rather a summary of the area of MDE to those who are not familiar with the general benefits and techniques used in model engineering. Attendees are welcome from a broad range of backgrounds (e.g., undergraduate and graduate students, as well as faculty and professional developers).

Attendees are encouraged to bring a laptop. The second session of the tutorial will be strongly hands-on and attendees will have the opportunity to explore the concepts of the tutorial while developing a simple modeling language. However, a laptop is not required and the topic of the tutorial can still be understood by engaging the tutorial speakers during the hands-on session. Please contact the authors in advance for any information about pre-installing the software. We will also try to have the modeling tools installed on the computer lab hosting the presentation.
4. TUTORIAL SPEAKER BIOGRAPHIES

Dr. Jeff Gray is an Associate Professor in the Department of Computer Science at the University of Alabama. His research interests include model-driven engineering, aspect orientation, code clones, and generative programming. Jeff received a Ph.D. in Computer Science from Vanderbilt University and both the BS and MS in Computer Science from West Virginia University. He is a member of the ACM and a Senior Member of the IEEE. Jeff is a co-founder of the OOPSLA workshop on domain-specific modeling and a frequent organizer of events in the modeling community. Details about Jeff’s interests can be found at: http://www.cs.ua.edu/~gray

Dr. Jules White is a Research Assistant Professor at Vanderbilt University. He received his BA in Computer Science from Brown University, his MS in Computer Science from Vanderbilt University, and his Ph.D. in Computer Science from Vanderbilt University. Dr. White’s research focuses on applying a combination of model-driven engineering and constraint-based optimization techniques to the deployment and configuration of complex software systems. Dr. White is the project leader for the Generic Eclipse Modeling System (GEMS), an Eclipse Foundation project. GEMS will be the tool used to introduce modeling concepts in this tutorial. More information about Jules can be obtained at: http://www.dre.vanderbilt.edu/~jules/

Dr. Aniruddha S. Gokhale is an Assistant Professor of Computer Science and Engineering in the Department of Electrical Engineering and Computer Science at Vanderbilt University. He received his BE (Computer Eng) from Pune University; MS (Computer Science) from Arizona State University; and D.Sc (Computer Science) from Washington University. Prior to joining Vanderbilt, he was a Member of Technical Staff at Bell Labs, Lucent Technologies. Dr. Gokhale is a member of IEEE and ACM. Dr. Gokhale’s research combines model-driven engineering and middleware for distributed systems, notably real-time and embedded systems. He is the project leader for the CoSMIC model-driven engineering tool suite at Vanderbilt. More information about Aniruddha’s research and interests can be found at: http://www.dre.vanderbilt.edu/~gokhale/