

Welcome to the Sixth OOPSLA Workshop on Domain-Specific Modeling – DSM'06

Preface

Domain-Specific Modeling (DSM) raises the level of abstraction beyond programming by specifying the solution directly using domain concepts. In many cases, final products can be generated automatically from these high-level specifications. This automation is possible because both the language and generators need fit the requirements of only one company and domain.

Industrial experiences from applying DSM consistently show it to be 5-10 times faster than current practices, including current UML-based implementations of MDA. As Booch* et al. have stated, "the full value of MDA is only achieved when the modeling concepts map directly to domain concepts rather than computer technology concepts." For example, DSM for cell phone software would have concepts like "Soft key button", "SMS" and "Ring tone", and generators to create calls to corresponding code components.

More investigation is still needed in order to advance the acceptance and viability of domain-specific modeling. This workshop, which is in its six incarnation at OOPSLA 2006, features research and position papers describing new ideas at either a practical or theoretical level. On the practical side, several papers in these proceedings describe application of modeling techniques within a specific domain. In addition to industrial projects, several authors from academia present research ideas that initiate and forward the technical underpinnings of domain-specific modeling. In particular, the 22 papers included in this proceedings highlight the importance of metamodeling, which significantly eases the implementation of domain-specific languages and provides support for experimenting with the modeling language as it is built (thus, metamodel-based language definition also assists in the task of constructing generators that reduce the burden of tool creation and maintenance). We hope that you will enjoy the workshop and find the information within these proceedings valuable toward your understanding of the current state-of-the-art in domain-specific modeling.

Jeff Gray, Juha-Pekka Tolvanen, Jonathan Sprinkle

October 2006

* Grady Booch, Alan Brown, Sridhar Iyengar, Jim Rumbaugh, and Bran Selic, *MDA Journal*, May 2004

6th WORKSHOP ON DOMAIN-SPECIFIC MODELING

22nd October, 2006, Portland, Oregon USA

Program committee

Scott Ambler, IBM
Pierre America, Philips
Philip T. Cox, Dalhousie University
Krzysztof Czarnecki, University of Waterloo
Andy Evans, Xactium
Jeff Gray, University of Alabama at Birmingham
Jack Greenfield, Microsoft
Jürgen Jung, University of Duisburg-Essen
Steven Kelly, MetaCase
Jürgen Kerstna, St. Jude Medical
Kalle Lyytinen, Case Western Reserve University
Pentti Marttiin, Nokia
Birger Møller-Pedersen, University of Oslo
Matti Rossi, Helsinki School of Economics
Arturo Sanchez, University of North Florida
Jonathan Sprinkle, University of California, Berkeley
Juha-Pekka Tolvanen, MetaCase
Markus Völter, independent consultant

Organizing committee

Jeff Gray, University of Alabama at Birmingham
Jonathan Sprinkle, University of California, Berkeley
Juha-Pekka Tolvanen, MetaCase

Table of Contents

Welcome Message from the Organizers	i
List of Program and Organizing Committees	ii
Table of Contents	iii-iv
Using Domain-Specific Modeling towards Computer Games Development Industrialization <i>André W. B. Furtado and André L. M. Santos</i>	1-14
Building a Flexible Software Factory Using Partial Domain Specific Models <i>Jos Warmer and Anneke Kleppe</i>	15-22
Conceptual design of web application families: the BWW approach <i>Roberto Paiano, Andrea Pandurino, and Anna Lisa Guido</i>	23-32
Building End-User Programming Systems Based on a Domain-Specific Language <i>Herbert Prähofer, Dominik Hurnaus, and Hanspeter Mössenböck</i>	33-42
Dart: A Meta-Level Object-Oriented Framework for Task-Specific Behavior Modeling by Domain Experts <i>Reza Razavi, Jean-François Perrot, and Ralph Johnson</i>	43-55
Genie: a Domain-Specific Modeling Tool for the Generation of Adaptive and Reflective Middleware Families <i>Nelly Bencomo and Gordon Blair</i>	56-66
Incremental Development of a Domain-Specific Language That Supports Multiple Application Styles <i>Kevin Bierhoff, Edy Liongosari, and Kishore Swaminathan</i>	67-78
Programmatic Building of Models Just for Pretty Printing <i>Tero Hasu</i>	79-86
Toward Families of QVT DSL and Tool <i>Benoît Langlois, Daniel Exertier, and Ghanshyamsinh Devda</i>	87-97
Preserving Architectural Knowledge Through Domain-Specific Modeling <i>Femi G. Olumofin and Vojislav B. Mićić</i>	98-104
Domain Model Driven Development of Web Applications <i>Dzenan Ridjanovic</i>	105-112
Generative Programming for a Component-based Framework of Distributed Embedded Systems <i>Xu Ke and Krzysztof Sierszecki</i>	113-122
Techniques for Metamodel Composition <i>Matthew Emerson and Janos Sztipanovits</i>	123-139
On Relationships among Models, Meta Models and Ontologies <i>Motoshi Saeki and Haruhiko Kaiya</i>	140-149
Roles in Software Development using Domain Specific Modelling Languages <i>Holger Krahn, Bernhard Rumpe, and Steven Volkel</i>	150-158
Lightweight Domain-Specific Modeling and Model-Driven Development <i>Risto Pitkänen and Tommi Mikkonen</i>	159-168
How to represent Models, Languages and Transformations? <i>Martin Feilkas</i>	169-176
Model Integration in Domains Requiring User-Model Interaction and Continuous Adaptation of Metamodel <i>Peter Krall</i>	177-184

Table of Contents (continued)

The Practice of Deploying DSM Report from a Japanese Appliance Maker Trenches <i>Laurent Safa</i>	185-196
A Model-Based Workflow Approach for Scientific Applications <i>Leonardo Salayandía, Paulo Pinheiro da Silva, Ann Q. Gates, and Alvaro Rebellon</i>	197-203
Bootstrapping Domain-Specific Model-Driven Software Development within Philips <i>Hans Jonkers, Marc Stroucken, and Richard Vdovjak</i>	204-213
Domain-Specific Model Composition Using a Lattice of Coalgebras <i>Jennifer Streb, Garrin Kimmell, Nicolas Frisby, and Perry Alexander</i>	214-221