

Implementing Domain-Specific Modeling Languages and Generators

Tutorial at the
2005 IEEE Symposium on Visual Languages and Human-Centric
Computing
(VL/HCC'05)
Dallas, Texas, USA
September 24, 2005

Steven Kelly

| | |
|----------|---|
| Date | September 24, 2005, half day, morning |
| Venue | VL/HCC Symposium 2005 |
| Address | Southfork Hotel, 1600 North Central Expressway, Plano, TX 75074, USA |
| Audience | Attendees should have significant development experience, not necessarily OO, and should have used at least one modeling language and design/generation tool. |

Domain-Specific Modeling (DSM) languages provide a viable solution for improving development productivity by raising the level of abstraction beyond coding. With DSM, the models are made up of elements representing concepts that are part of the problem domain world, not the code world (unlike for example the core UML concepts). DSM languages follow domain abstractions and semantics, allowing developers to perceive themselves as working directly with domain concepts. In many cases, full final product code can be automatically generated from these high-level specifications with domain-specific code generators.

This tutorial introduces DSM and looks at how it differs from modeling languages like UML that focus more on the level of the code world. This is followed by real-life examples of DSM from various fields of software product development. The main part of the tutorial addresses the guidelines for implementing DSM: how to identify the necessary language constructs, and different ways of building code generation. Participants will be able to try their hand and learn these skills in practice in group exercises.

Introduction (30 min)

- What is domain-specific modeling (DSM)?

- Why DSM?
- DSM examples from different application domains
- Where to apply

Implementing modeling languages (60 min)

- How to identify modeling concepts and rules
- Metamodels and metamodeling
- Sample languages
- Exercise: extending a modeling language (i.e. metamodel)

Implementing generators (60 min)

- Different generator approaches
- Interfacing to generated code
- Sample generators (C, Java, XML, Assembler)
- Exercise: extending generator

Wrap-up and conclusions (30 min)

- Industrial experience reports
- Summary
- What next?

Tutorial Format

Half-day, morning of September 24th, at UTD Campus. The tutorial uses slides, demonstrations of tools, languages and generators, and interactive group tasks to create parts of sample domain-specific languages and code generators.

Speaker

Dr. Steven Kelly is the CTO of MetaCase and co-founder of the DSM Forum. He has over ten years of experience of building metaCASE environments and acting as a consultant on their use in domain-specific modeling. As architect and lead developer of MetaEdit+, MetaCase's domain-specific modeling tool, he has seen it win or be a finalist in awards from Byte, the Innosuomi prize for innovation awarded by the Finnish President, Net.Object Days, and the Software Development Jolt awards. Ever-present on the program committee of the OOPSLA workshops on Domain-Specific Modeling, he co-organized the first workshop in 2001. Author of over 20 articles, most recently in journals such as Dr. Dobb's and ObjektSpektrum, he is also a member of the editorial board for the Journal of Database Management. Outside of work, he has co-authored the first grammar of the Kenyan Orma language, and is a soccer player in the Finnish Third Division.

Steven Kelly
MetaCase
Ylistönmäentie 31
FI-40500 Jyväskylä, Finland
stevek@metacase.com
Tel. +358 14 4451 401
Fax +358 14 4451 405