Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Andreas Seibel
System Analysis and Modeling Group
Hasso Plattner Institute
at the University of Potsdam
andreas.seibel@hpi.uni-potsdam.de

under supervision of
Prof. Dr. Holger Giese
System Analysis and Modeling Group
Hasso Plattner Institute
at the University of Potsdam
holger.giese@hpi.uni-potsdam.de
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Agenda

Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Agenda

- Introduction
- Motivation
- State of the Art
- Our Approach
- Future Work
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Introduction

Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Introduction

What is Model-Driven Development?

„Model-driven development is simply the notion that we can construct a model of a system that we can then transform into the real thing.“

[S. Mellor, 2003]

- Does every developer apply model-driven development?
  - Yes and no! Depends on the kinds of models we consider

- We focus on models at higher level of abstraction (above GPLs)
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Introduction

What is Model-Driven Development?

- Level of abstraction
  - The higher the level of abstraction, the more focussed to the problem domain

- Automation through model operations
  - e.g., employ model transformations to transform models at different levels of abstraction

Abstract

Concrete

Subject matter

Language

Concrete

Abstract

DSMLs

UML

GPLs

• Level of abstraction
• The higher the level of abstraction, the more focussed to the problem domain

• Automation through model operations
• e.g., employ model transformations to transform models at different levels of abstraction

[S.Mellor, 2003]

[Diagram showing levels of abstraction and language relationships]
Motivation

Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

**Complexity in MDD: # models**

- **Three dimensions** of models
  - **X - Concerns**
  - **Y - Level of abstraction**
  - **Z - Versions**

---

Andreas Seibel (andreas.seibel@hpi.uni-potsdam.de) | MDD-RW 2010 | 27th - 28th July | Karlsruhe
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

Complexity in MDD: # dependencies

- Are ...
  - ... models at different levels of abstraction ...
  - ... models representing different concerns ...
  - ... different version of models ...
  - ... orthogonal? not necessarily!
- Models usually have overlaps
Motivation

Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Complexity in MDD: # dependencies

- **Vertical dependencies**
  - *What?* Interrelate different levels of abstraction
  - *Syntax?* Dependencies defined between models
  - *Semantic?* Application of model transformations (MDA)

![Diagram showing the flow from Requirements to Analysis & Design to Implementation with specified by and realized by annotations.](image-url)
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

Complexity in MDD: # dependencies

- **Horizontal dependencies**
  - **What?**
    - Interrelate different concerns
  - **Syntax?**
    - Dependencies defined between models elements and thus between models
  - **Semantic?**
    - Condition between models elements in different models (OCL constraints), etc.
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

Complexity in MDD: # changes

• An MDD development process is not static but rather a highly dynamic process

• Changes spread over different models

• Changes in one model may have impact on dependent models → Inconsistencies

[Jacobson, 1999]
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

Case Study Scenarios: D-MDA

- In cooperation with CA Labs
  ![CA Labs Logo](ca.png)

- Improve service of deploying software to IT infrastructures of huge companies

- We have developed as set of DSMLs for this task

![Diagram of D-MDA case study](diagram.png)
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

Case Study Scenarios: Embedded Systems Modeling

- Partially in cooperation with dSPACE and INCHRON
- Currently under development

![Diagram showing the integration of hardware (HW) and software (SW) architecture, requirements, timing chains, and system configuration with synchronization and deployment processes using SysML, AUTOSAR, and ChronSIM.]
Motivation

Challenges

• MDD is a complex endeavor
  • Working with many models
  • Models are not orthogonal but have overlaps → dependencies
  • Changes occur frequently and nearly everywhere
    • Inconsistencies may occur
    • Thread to the success of MDD-based projects
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

What do we need?

• The capability to *explicitly* define dependencies …
  1. … between model elements of different models (microscopic)
  2. … between models (macroscopic)
  3. … between dependencies (combining micro and macro)

![Diagram showing relationships between modeling languages and components](image-url)

CA case study

dSPACE & INCHRON case study
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Motivation

What do we need?

- Heterogenous dependencies
  - Dependencies should not be restricted in their representation ability

- Support for automation on a macroscopic/microscopic level
  - Identification of dependencies (*Localization*)
  - Impact analysis
  - (Re-)establishing consistency (*Execution*)
State of the Art

Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

State of the Art

From microscopic to macroscopic

Microscopic approaches
- Classical traceability
- Model transformations
- ...

Macroscopic approaches
- Megamodeling
- Macromodeling
- Multimodeling
- ...

![Diagram showing relationships between models and elements](image-url)
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

State of the Art

Direct comparison of approaches

<table>
<thead>
<tr>
<th>Microscopic Approaches</th>
<th>microscopic</th>
<th>macroscopic</th>
<th>heterogenity</th>
<th>localization</th>
<th>impact analysis</th>
<th>execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>[N. Drivalos, 2008]</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[M. Aleksy, 2008]</td>
<td>x</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[H. Asuncion, 2008]</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>(x)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[J. R. Falleri, 2006]</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[H. LeDang, 2008]</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[B. Vanhooff, 2007]</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[N. Aizenbud-Reshef, 2005]</td>
<td>x</td>
<td>-</td>
<td>(x)</td>
<td>(x)</td>
<td>(x)</td>
<td>(x)</td>
</tr>
<tr>
<td>[P. Mader, 2008]</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>(x)</td>
<td>(x)</td>
<td>-</td>
</tr>
<tr>
<td>[S. Walderhaug, 2006]</td>
<td>x</td>
<td>-</td>
<td>(x)</td>
<td>(x)</td>
<td>(x)</td>
<td>(x)</td>
</tr>
<tr>
<td>[M. Barbero, 2007]</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[R. Salay, 2009]</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>[M. Del Fabro, 2009]</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>[M. Fritzse, 2009]</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>(x)</td>
<td>(x)</td>
<td>(x)</td>
</tr>
<tr>
<td>[F. Allilaire, 2006]</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>(x)</td>
</tr>
<tr>
<td>[J. M. Favre, 2005]</td>
<td>-</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[F. Jouault, 2010]</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[P. A. Muller, 2009]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[D. Kolovos, 2008]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>x</td>
</tr>
</tbody>
</table>

- not supported  x supported  (x) mentioned
Our Approach

Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios
Our Approach

Overview

- Combines microscopic and macroscopic dependencies by extending the notion of megamodels

- **Modeling:**
  not restricted to any specific DSML or even GPML

- **Megamodelling:**
  acts as integration language for DSMLs or even GPML

- **Tooling:**
  supports users using our approach (localization, IA, execution, ...)

Andreas Seibel (andreas.seibel@hpi.uni-potsdam.de) | MDD-RW 2010 | 27th - 28th July | Karlsruhe
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Our Approach

Modeling

- Applying a supported modeling environment (e.g., EMF)
- Make models available in some repository (e.g., workspace in Eclipse) for integration
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Our Approach

Megamodeling

- Metamodel of a megamodel (model of modeling)

Models
Model Elements
Dependencies
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Contribution

Megamodelling

- Explain by example

![Diagram showing relationships between different modeling languages and megamodelling concepts](image-url)
Megamodeling

- A megamodel can be considered as ...
  - … a static model
  - … a dynamic model
    - dependencies are interpreted as executable units
    - (Re-)establishing consistency
  - Heterogenous dependencies
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Contribution

Tooling: Localization

- Localization supports identification of dependencies
- Applied on the whole megamodel
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Contribution

Tooling: Impact Analysis

- Whenever models change, existing dependencies may get invalidated!
- Output of invalidated dependencies are at least potentially inconsistent!
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Contribution

**Tooling: Execution**

- Potential inconstencies should be resolved
- In this example, dependency executed by a synchronization
- Execution is obtained on the whole megamodel
Future Work

Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

Future Work

Research perspective?

Not mentioned in this talk?

• Can be used as a language for horizontal and vertical composition of heterogenous model operations

• Localization and execution are capable of interpreting the megamodel for this task

What have to be done?

• Working on a complete formal definition of the approach

• Research:

  • Megamodels and versioning
  • Megamodels and runtime systems
  • Megamodels and patterns
  • ...

Future Work

Take away

• Facilitate *megamodels* to support MDD by ...  
  • ... integrating *heterogenous models*  
  • ... explicitly capturing *heterogenous dependencies*  
    • between model elements (microscopic)  
    • between models (macroscopic)  
    • between dependencies  
  • ... providing *automation* on a macroscopic/microscopic level
Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

References


Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

References


Dealing with Multiple Overlapping Modeling Languages in Complex Model-Driven Development Scenarios

References

