Model-Based Design

DSML Composition
Model Synthesis and
Model Transformation

Janos Sztipanovits and Gabor Karsai
November 14, 2002

UC Berkeley: Chess
Vanderbilt University: ISIS
University of Memphis: MSI

Foundations of Hybrid and Embedded Software Systems
Platforms
(There are many...)

Time-Triggered Architecture (distributed, hard real-time, safe)

Integration framework, composition mechanisms, components

QoS Middleware (such as CORBA)

Application
QoS Middleware
Operating System
Hardware
CPU, MEM, I/O

Strong isolation between SW and HW by Active Control

Application
QoS parameters
Control Algorithm
Measured vars.
Control vars.

Challenges in Model-Based Design

Design
Model-Based Design
Model-Based Design of Embedded Systems

Application Space
Mapping
Implementation Space

DSM
DSM
DSM
DSM

Application Models
Gen./Synth.
Implement. Models

Composition of
- Domain Specific Modeling Languages (DSML)
- Model Synthesis
- Model Transformation
Specification of Domain Specific Modeling Languages (DSML)

\[ L = < C, A, S, M_S, M_C > \]

Concrete Syntax and Abstract Syntax

Signal Flow Language (SF)

Mathematical abstraction for specifying the meaning of models

But What About S?

Notation for representing models:
E.g.: Block diagram
Semantics via Meta-Modeling

- Meta-modeling language with well-defined semantics
- Meta-Model of StateFlow using uml/OCL as meta modeling language.

Semantics via Translation

- Modeling language with well-defined semantics
- Synchronous Dataflow (SDF)
- Hierarchical Signal Flow (HSF)
**DSML Composition**

Composed Behavioral Semantics: HYBRID SYSTEM

**Simple Support for Compositional Meta-Modeling**

**Composition Operators**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Symbol</th>
<th>Informal semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equivalence</td>
<td><img src="image" alt="Equivalence" /></td>
<td>Complete equivalence of two classes</td>
</tr>
<tr>
<td>Implementation Inheritance</td>
<td><img src="image" alt="Inheritance" /></td>
<td>Child inherits all of the parent’s attributes and those containment associations where parent functions as container.</td>
</tr>
<tr>
<td>Interface Inheritance</td>
<td><img src="image" alt="Inheritance" /></td>
<td>Child inherits all associations except containment associations where parent functions as container.</td>
</tr>
</tbody>
</table>
Research Agenda on Domain Specific Modeling Languages

- Precise, compositional meta-modeling
- Multiple aspect modeling in the compositional meta-modeling framework
- Practical issues:
  - Examples, meta-model libraries
  - Meta-programmable tools
  - Link to UML-2

Model Synthesis and Transformations

Model-Based Design of Embedded Systems

- Model Synthesis
- Model Transformation
Roles transformations play in model-based design:
• Refining a design into an implementation
• Code generation
• PIM -> PSM mapping
• Support for model interchange for tool integration

Approach (Karsai): Meta-models for source and target models plus transformations, then generating the transformer.
Iterative ("for") and conditional ("if") constructs in models can greatly enhance expressiveness while reducing complexity.

Input design: with "for" loop

What it means:

Replicate the middle of pipeline "Order" times!

Order = 3
**Other Uses of Transforms on Models: Design Patterns**

Design patterns capture prototypical solutions to a design problem. As such, they are **parametric** and **reusable**.

---

**Research Agenda on Model Transformations**

- Languages and tools for meta generators
- Model synthesis using explicit design patterns
- Model synthesis using constraint-based design-space exploration
- Generative modeling extensions to languages
- Embeddable generators