Model-Integrated Computing (MIC)  
(or MDA, MDE, MDD, MBD,...)

Janos Sztipanovits & Gabor Karsai  
ISIS, Vanderbilt University
Model-based design focuses on the *formal representation, composition, and manipulation of models* during the design process.
Research Agenda in Model-Based Design

1. Composition of Domain Specific Modeling Languages
2. Model Transformation
3. Model Synthesis
doTransition (fsm as FSM, s as State, t as Transition) =
require s.active
step exitState (s)
step if t.outputEvent <> null then emitEvent (fsm, t.outputEvent)
step activateState (fsm, t.dst)

Semantic Domain
Meta-models

Domain Specific Modeling Languages (DSML)

L = < C, A, S, MS, MC>

Abstract Syntax A

Concrete Syntax C

Parsing

Semantic Mapping

Semantic Domain S

Concepts
Relations
Well formed-ness rules

Notation for representing models

Mathematical abstraction for specifying the meaning of models

Domain models Interchange Formats

Abstract Syntax Meta-models

Semantic Domain Meta-models

DSML-s are the foundations for model-based design
Mission Control Platform (MCP) Tool Chain from “Above”

Mission Control Platform Tool Chain

Common Semantic Domain
Abstract Syntax: Meta-Models
Domain Models and Tool Interchange Formats

Mission Control Platform Tool Chain

Component Model
Component Structure
Thread Models
Component Interaction
Schedulability Analysis
System Integration

Rational Rose
PRISM → ESML
ESML → AIF
ESML → CFG

BoldStroke/PRISM CFG

GME

AIRES
PRISM
Æ
ESML
Æ
CFG
Æ
AIF
Constructing Design Flow: Modeling and Transformations

- Large influence of concrete syntax
- No clear role of semantics
- It is not clear what are we doing?
Abstract Syntax Metamodelling

- Gives structural semantics for the models
- Set-valued Semantic Domain for the metamodels and transformations
Mission Control Platform at the “Middle” Layer

Common Semantic Domain

Abstract Syntax and Transformations: Meta-Models

Domain Models and Tool Interchange Formats

Mission Control Platform Tool Chain
Metamodeling of DSML Using **MOF**

Metamodeling of MOF Using **MetaGME**

**Metamodeling Languages**

- **DSML** → **MOF**
- **MOF** → **MetaGME**

### Changing from GME/Meta to MOF

- **T₂** → **MOFA_{DSML}** → **MOF** → **MetaGME** → **MetaGME_{A_{MOF}}** → **MetaGME**
- **T₁** → **MetaGME** → **MetaGME_{A_{MOF}}** → **MetaGME**
Metaprogrammable Tool Suite

Generic Model Editor Tool Suite
GME

Unified Data Model
UDM

Open Tool Integration Framework
OTIF

Model Transformation Tool Suite
GReAT

Design Space Exploration Tool Suite
DESERT

• Simulators
• Verifiers
• Model Checkers
• Generators

Open Source; ESCHER maturation program (www.escherinstitute.org)
Ongoing Research on DSML-s and Model Transformations

- Compositional construction of Metamodels (inheritance, packages, libraries, operators)
- Compositional construction of Model Transformations
- Multiple Aspect Modeling and modeling of aspect inter-dependences:
  - constraint-based,
  - transformation-based
- Formal semantics of transformations
- Reasoning about properties of transformations
- Platform modeling and use of embedded platform models in transformations
- Pushback reasoning in transformations
- Generation of efficient code from graph transformations
- Transformations for embedded system platforms
- Using graph transformations for embedded component adaptation
- Embedding graph transformations in the run-time platform
Metamodelling and Model Transformation Use Cases

1. Transformational Specification of Behavioral Semantics
Metamodeling and Model Transformations Use Cases

2. Semantic Anchoring of DSML-s

- The “Semantic Units” are selected abstract semantics such as MoC-s
- DSML-s or their aspects are anchored to the MoC-s using transformations
- The “Semantic Units” are specified in a formal framework
More On Semantic Anchoring

- **Step 1**
  - Define the DSML metamodel \( \langle A, C, M_c \rangle \)

- **Step 2**
  - Select a proper MoC as a “semantic unit” (MoC library): \( L_i = \langle A_i, C_i, M_{C_i}, S_i, M_{S_i} \rangle \)

- **Step 3**
  - Anchor the semantics: \( M_A = A \rightarrow A_i \)
  - DSML semantics: \( L = \langle A, C, M_c, S_i, M_A \circ M_{S_i} \rangle \)
Experimental Framework

MOF
Set-Valued SD

HSML
MetaModel

HSML
Model

Syntax conversion
Expressed in
Instance of

FSM
MetaModel

FSM
Model

FSM
Data Model

FSM
Data Inst.

Asml
Set-Valued + Abstract State Machine SD

FSM
Interpreter
Long-Term Vision: Semantic Anchoring
Architecture of DSML-s

Syntactic Level

M3

Meta-metamodel
Set-Valued Semantics

DSML Metamodel
Set-Valued Semantics

Model Transf. Spec.

Semantic Level

M2

Metamodel
Set-Valued Semantics

Model Transf. Spec.

M3 M2 M1

Translation

Model Transf.

M1

Model Semantics

Model Simulation
Model Verification
Model Execution

MoC “Semantic Units”

Meta-metamodel
Set-Valued Semantics

Semantic Domain
Behavioral Semantics

ASML

JAVA

TLA+

Host Language
Behavioral Semantics

ASML

JAVA

TLA+

Host Language
Behavioral Semantics

Model Semantics

ASML

JAVA

TLA+

Host Language
Behavioral Semantics

Model Simulation
Model Verification
Model Execution

MoC “Semantic Units”
A “Semantic Unit” for Time Automata Based Modeling Languages

- Common semantic domain for varied timed automata based modeling languages: (In cooperation with VERIMAG)
  - Guard
  - Priority
  - Synchronization

AsmL Tools
- Model Checker
- Test Case Generator
- Model Simulator

MoC Semantic Unit
- Time Automata (AsmL Supported Semantics)

Time Automata Variants
- UPPAAL Lang.
- IF Lang.
- GME Toolset

GReAT Tool
- Semantic Anchoring
- Transformation

Model Checker
- $M_S$
- $M_C$
“Evidence”: IF Metamodels
Metamodelling and Model Transformation Use Cases

3. Semantic Integration of Tools

Modeling Tool

DSML-1

DSML-2

Analysis Tool

Semantic Unit
(Common Semantic Domain)

Metamodeling and Model Transformation Use Cases
Ongoing Cooperations

• Artist 2 (Prof. Sifakis, Verimag)
  - MIC-based modeling interface for IF;
  - Semantic Unit specification for Time Automata
  - Modeling Language Design Environment

• DECOS (Prof. Kopetz, TUV)
  - MIC-based modeling environment for DECOS
  - Automotive Design Tool Chain
  - DESERT Applications