Computer Aided Multi-paradigm Modelling of Hybrid Systems with AToM³

Juan de Lara and Manuel Alfonseca

E.T.S. de Informática
Universidad Autónoma de Madrid, Madrid, Spain

Hans Vangheluwe

Modelling, Simulation and Design Lab (MSDL)
School of Computer Science, McGill University, Montréal, Canada
Presentation Overview

- Multi-paradigm Modelling and Simulation
- An example Hybrid model
- Meta-modelling
- AToM3: A Tool for Multi-formalism Meta-Modelling
- The future . . .
Multi-paradigm modelling and simulation

1. Different *levels of abstraction*

2. Mixing *different formalisms* (coupling, transformation)

3. Modelling classes of models (formalisms) by *meta-modelling*

4. Modelling *transformations* explicitly
   (in the Graph Grammar formalism)

---

TOMACS 12(4) 2002: Special Issue on CAMPaM.
Mosterman and Vangheluwe.
A Hybrid Example: $T, l$ controlled liquid
Simulation 79(1) 2003: Visual Interactive Simulation for Distance Eductation. de Lara and Alfonseca.
Computer Aided Multi-paradigm Modelling of Hybrid Systems with ATom³
Meta-modelling

A meta-model is a model in formalism MF that describes a model of a class of models (the formalism F) within formalism MF. This meta-model describes structure and constraints.

The meta-model processor takes user input and processes it to create, delete, or verify models (local, global) in formalism F.

A model in formalism F is created and validated by the meta-model processor.
Statecharts Meta-model

Hyperedge

Blob
Name type=String init.val
composed_of
has_inside

Orthogonal
Name type=String init.val

Initial
Actions type=String init.
has_Initial

iconnection
Statecharts Meta-model in AToM³
Edit Entity
Edit Entity Appearance

Attributes

Appearance

Select
Delete
Line
Polygon
Oval
Rectangle
Text
Attribute
Connector
Properties
Set Constraint
Changes at run-time

OK
Cancel
The Continuous Plant Model (ALG + ODE)

Inputs (discontinuous → hybrid model):
- Emptying, filling flow rate $\phi$
- Rate of adding/removing heat $W$

Parameters:
- Cross-section surface of vessel $A$
- Specific heat of liquid $c$
- Density of liquid $\rho$

State variables:
- Temperature $T$
- Level of liquid $l$

Outputs (sensors):
- $is\_low, is\_high, is\_cold, is\_hot$

\[
\begin{align*}
\frac{dT}{dt} &= \frac{1}{l}[\frac{W}{c\rho A} - \phi T] \\
\frac{dl}{dt} &= \phi \\
\text{is}_\text{low} &= (l < l_{low}) \\
\text{is}_\text{high} &= (l > l_{high}) \\
\text{is}_\text{cold} &= (T < T_{cold}) \\
\text{is}_\text{hot} &= (T > T_{hot})
\end{align*}
\]
CBD Meta-model (ER)
\[ x'' = -Kx, \quad x(0) = 0, \quad x'(0) = 1 \]
\[
\begin{cases}
\dot{x} = y, & x(t_0) = x_0 \\
\dot{y} = -(x) \cdot K, & y(t_0) = y_0 \\
x_0 = 0.0 \\
K = 1.0 \\
y_0 = 1.0
\end{cases}
\]
\[ \begin{aligned}
\dot{x} &= y, \quad x(t_0) = x_0 \\
\dot{y} &= -x \cdot K, \quad y(t_0) = y_0
\end{aligned} \]

\[ \begin{aligned}
x_0 &= 0.0 \\
K &= 1.0 \\
y_0 &= 1.0
\end{aligned} \]
function xdot = harmonic(t, x)
xdot = zeros(2, 1);
xdot(1) = x(2);
xdot(2) = -(x(1)) * 1.0;

x0 = [0.0; 1.0];
OOCSMP meta-model (ER)
The Complete Model
Meta-meta-...
ER Meta-model (ER)

ERrelationship

ERentity

name type=String init.val
attributes type=List init
Model Transformation Specification

- A model in formalism ER
  - Meta-model: a model of a class of models (the formalism NFA) semantics within formalism ER
  - User input: create, delete, verify (local, global)
  - Model transformer = meta-model processor

- A model in formalism MF
  - Meta-model: a model of a class of models (the formalism F) semantics within formalism MF
  - User input: create, delete, verify (local, global)
  - Model transformer = meta-model processor

- A model in formalism NFA

- A model in formalism FSA

- Multi-formalism model transformer = meta-model processor
Related Work

- The OMG’s Model Driven Architecture (MDA)
- Graph Grammars: PROGRES, AGG, ...
- Ptolemy (Berkely, Ed Lee): co-simulation
- Vanderbilt (ISIS): Multigraph Architecture, GME
- Honeywell: DOME
The Future …

- Domain-specific modelling and simulation environments
- Experiment with variations (flavours) of formalisms (syntax and semantics)
- Formalism Transformation (FTG)
- Graph Grammars *models* for all Transformations
- Simulator Meta-specification (reference implementation)
- Model exchange (DTD from meta-model, XML from model)
Acknowledgements

- The Modelling, Simulation and Design Lab (MSDL)
- Jean-Sébastien Bolduc (CBD transformations)
- NSERC http://www.nserc.ca