An Overview of the SysML-Modelica Transformation Specification

Chris Paredis (Georgia Tech)

Y. Bernard (Airbus), R. Burkhart (Deere & Co), H. de Koning (ESA/ESTEC), S. Friedenthal (Lockheed Martin Corp.), P. Fritzson (Linköping University), N. Rouquette (JPL), W. Schamai (EADS)
Two complementary languages for Systems Engineering:
- Descriptive modeling in SysML
- Formal equation-based modeling for analyses and trade studies in Modelica

Objective:
- Leverage the strengths of both SysML and Modelica by integrating them to create a more expressive and formal MBSE language.
- Define a formal Transformation Specification:
  - a SysML4Modelica profile
  - a Modelica abstract syntax metamodel
  - a mapping between Modelica and the profile
Presentation Overview

- What is SysML?
- What is Modelica?

- Motivating Example: Design & Analysis of Robot

- SysML-Modelica Transformation Specification
- Transformations in Systems Modeling
- Timeline towards Specification Adoption
- Summary
What is SysML? (www.omgsysml.org)

1. Structure

```
ibd [block] Anti-LockController
[Internal Block Diagram]
```

```
c1:modulator Interface
allocatedFrom
```

```
d1:TractionDetector
allocatedFrom
```

```
m1:BrakeModulator
allocatedFrom
```

```
c1:modulator Interface
```

```
d1:TractionDetector
```

```
m1:BrakeModulator
```

```
c1:modulator Interface
```

```
d1:TractionDetector
```

```
m1:BrakeModulator
```

```
c1:modulator Interface
```

```
d1:TractionDetector
```

```
m1:BrakeModulator
```

```
c1:modulator Interface
```

2. Behavior

```
act PreventLockup [Swimlane Diagram]
```

```
allocate
```

```
allocate
```

```
allocate
```

```
satisfies
```

```
requirement Anti-Lock Performance
```

```
values DutyCycle: Percentage
```

```
allocatedTo
```

```
allocatedTo
```

```
allocatedTo
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

```
allocate
```

3. Parametrics

```
par [constraintBlock] StraightLineVehicleDynamics [Parametric Diagram]
```

```
F = ma
```

```
a = dv/dt
```

```
v = dx/dt
```

```
f = (tf*bf)*(1-tl)
```

```
vt: F: t:
```

```
v: F: a: m:
```

```
v: F: a: m:
```

```
v: F: a: m:
```

```
v: F: a: m:
```

4. Requirements

```
req [package] VehicleSpecifications [Requirements Diagram - Braking Requirements]
```

```
Vehicle System Specification
```

```
Braking Subsystem Specification
```

```
VeriﬁedBy
```

```
SatisﬁedBy
```

```
<deriveReqt>
```

```
<requirement> StoppingDistance
```

```
<requirement> Anti-LockPerformance
```

```
id="102"
text="The vehicle shall stop from 60 mph within 150 ft on a clean dry surface."
```

```
VerifiedBy
```

```
<requirement> Anti-LockPerformance
```

```
id="337"
text="Braking subsystem shall prevent wheel lockup under all braking conditions."
```

```
SatisﬁedBy
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```

```
<block> Anti-LockController
```

```
<interactio
```
What is Modelica? (www.modelica.org)

- State-of-the-art Modeling Language for System Dynamics
  - Differential Algebraic Equations (DAE)
  - Discrete Events

- Formal, object-oriented language
- Standardized by the Modelica Association
  - Open language specification – tool independent
- Multi-domain modeling
- Ports represent energy flow (undirected) or signal flow (directed)
- Acausal, equation-based, declarative (f-m*a=0)
Modelica: Active and Mature Community

- Modelica association — 20+ free libs (www.modelica.org)
- 6 commercial solvers, 3 open-source solvers (Dymola, MapleSim, SimulationX, OpenModelica, …)
- EUROSYSLIB project — 20+ libs under development (http://www.itea2.org/public/project_leaflets/EUROSYSLIB_profile_oct-07.pdf)
A Robot Example in Modelica

motor torque
Connections represent Kirchhoff semantics

- Across variables (voltage, pressure,...) are equal
- Through variables (current, flow rate,...) add to zero

Graphical symbols defined as annotations in textual models

```modelica
model Spring "Linear 1D translational spring"
   extends Translational.Interfaces.PartialCompliant;
   parameter SI.TranslationalSpringConstant c(final min=0, start = 1) "spring constant ";
   parameter SI.Distance s_rel0=0 "unstretched spring length";

   equation
     f = c*(s_rel - s_rel0);

end Spring;
```
Presentation Overview

- What is SysML?
- What is Modelica?

Motivating Example: Design & Analysis of Robot

- SysML-Modelica Transformation Specification
- Transformations in Systems Modeling
- Timeline towards Specification Adoption
- Summary
A Robot Example in Modelica
SysML-Modelica Robot Example: UseCases & Requirements

- Operator
- Maintenance
- Programmer

Robot System
- Cost
- Performance
- Reliability
- Energy Consumption
- Weight
- Tracking Accuracy

Emergency Shutdown
- (Workspace Safety Violation)

Execute Robot Task
- extension points
  - Workspace Safety Violation

Program Trajectory
SysML-Modelica Robot Example:
Robot Domain BDD & IBD
SysML-Modelica Robot Example: Robot Arm BDD
Analysis models depend on descriptive models
SysML4Modelica Analytical Model: Compose Model from SysML Standard Library

Drag and drop into IBD [ModelicaModel]
SysML4Modelica Analytical Model: Detailed IBD
SysML4Modelica Analytical Model: Relation to Modelica Native Model
SysML4Modelica Analytical Model: Allocation
SysML-Modelica Robot Example: Modelica model with simulation results
Analysis results are incorporated in Trade Study
Presentation Overview

- What is SysML?
- What is Modelica?

- Motivating Example: Design & Analysis of Robot

SysML-Modelica Transformation Specification
- Transformations in Systems Modeling
- Timeline towards Specification Adoption
- Summary
SysML-Modelica Transformation follows the principles of Model-Driven Architecture (MDA)
model Spring "Linear 1D translational spring"
  extends Translational.Interfaces.PartialCompliant;
  parameter SI.TranslationalSpringConstant c(final min=0, start = 1)
    "spring constant ";
  parameter SI.Distance s_rel0=0 "unstretched spring length";
  equation
    f = c*(s_rel - s_rel0);
end Spring;
Reference implementation: Based on OMG QVT

QVT = Query / View / Transformation
Transformations in Systems Modeling

- Model Object
- Model Dependency

System Model
Transformations in Systems Modeling

- Model Object
- Model Dependency

SysML

System Model
Descriptive to Analytical Transformation

Correspondence Models

Descriptive Models

Analytical Models
Timeline of Specification Adoption

- **SysML**
  - SysML RFP: March 2003
  - 1.0 Specification: September 2007
  - Currently: Revision Task Force 1.3

- **Modelica**
  - 1.0 Specification: September 1997
  - 3.1 Specification: May 2009

- **SysML-Modelica**
  - Initial idea: July 2005
  - INCOSE MBSE Challenge Project: August 2007 – now
  - OMG Working Group established: December 2008
  - Approved for public comment (RFC): June 2010
  - Future: Adoption as OMG Specification in September 2010 (?)
Summary

Objective:
- Leverage the strengths of both SysML and Modelica by integrating them to create a more expressive and formal MBSE language.

Descriptive Modeling in SysML

+ Formal Equation-Based Modeling for Analyses and Trade Studies in Modelica

http://doc.omg.org/syseng/2010-6-8
Acknowledgements: Working Group Members

- Yves Bernard (EADS)
- Roger Burkhart (Deere & Co)
- Wuzhu Chen (Univ. Braunschweig)
- Hans-Peter De Koning (ESA)
- Sandy Friedenthal (Lockheed Martin)
- Peter Fritzson (Linköping University)
- Nerijus Jankevicius (No Magic)
- Alek Kerzhner (Georgia Tech)
- Andreas Korff (Atego)
- Chris Paredis (Georgia Tech)
- Axel Reichwein (Georgia Tech)
- Nicolas Rouquette (JPL)
- Wladimir Schamai (EADS)

http://doc.omg.org/syseng/2010-6-8