OMG SysPhs: Integrating SysML, Simulink, Modelica and FMI

Nerijus Jankevicius
CATIA | No Magic

INCOSE IW, Torrance, Jan 27, 2020
System Model as an Integration Framework

- External Requirements
- System Documentation and Specifications
- Analysis needs
- Closed form
- Discrete event
- Network

System Model (SysML)

Requirements
Parametrics

Requirements
Behavior

Structure
Behavior

traceability rationale
viewpoint

System framework for design

- Mechanical Design Models
- Electrical Design Models
- Software Design Models
- Testing Methods and Models

© 2012-2014 by Sanford Friedenthal
SysML as co-simulation environment
Reduce and standardize mappings

New OMG standard:
SysML Extension for Physical Interaction and Signal Flow Simulation (SysPhS)
## Unified Physics

<table>
<thead>
<tr>
<th>Domain</th>
<th>Flowing Substance</th>
<th>Flow rate</th>
<th>Potential to flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical</td>
<td>Charge</td>
<td>Current</td>
<td>Voltage</td>
</tr>
<tr>
<td>Hydraulic</td>
<td>Volume</td>
<td>Volumetric flow rate</td>
<td>Pressure</td>
</tr>
<tr>
<td>Rotational</td>
<td>Angular momentum</td>
<td>Torque</td>
<td>Angular velocity</td>
</tr>
<tr>
<td>Translational</td>
<td>Linear momentum</td>
<td>Force</td>
<td>Velocity</td>
</tr>
<tr>
<td>Thermal</td>
<td>Entropy</td>
<td>Entropy flow</td>
<td>Temperature</td>
</tr>
</tbody>
</table>

flow rate = amount of substance/time
flow rate * potential = energy / time = power
The Standard: SysPhS

- SysPhS - https://www.omg.org/spec/SysPhS/1.0
  - SysML mapping to Simulink and Modelica
  - SysPhS profile
  - SysPhS library

SysML Extension for Physical Interaction and Signal Flow Simulation

Version 1.0

OMG Document Number: formal/18-05-03
Release Date: June 2018
7.2 Simulation profile

Figure 1: Simulation stereotypes
Modelica vs Simulink

• Modelica
  • Language is better suited for physical modeling (plant)
  • Object oriented approach for modeling physical components (mechanical, electrical, etc.)
  • Causal and A-Causal semantics (equations)
  • Open standard (of the textual language)
  • Multi tool support (although Dymola is dominant)
  • Tool vendor independent

• Simulink
  • Language is well-suited for control algorithms
  • Transformational semantics of signals and signal processing
  • Causal semantics (inputs -> outputs)
  • Well integrated into the “MATLAB universe”
  • Widely used in industry (standard de-facto)
  • Many existing tool integrations
  • Code generation to C/C++/VHDL/Verilog
Platform profile

Figure 33: Simulation platform stereotypes
## Specification examples

<table>
<thead>
<tr>
<th>SysML</th>
<th>Modelica</th>
<th>Simulink</th>
<th>Simscape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port typed by block with an in flow property stereotyped by a non-conserved PhSVariable and typed by Real, Integer, Boolean or one of their specializations (signal flow)</td>
<td>Component typed by an equivalent data type</td>
<td>Inport</td>
<td>Input variable</td>
</tr>
<tr>
<td>Port typed by block with an out flow property stereotyped by a non-conserved PhSVariable and typed by Real, Integer, Boolean or one of their specializations (signal flow)</td>
<td>Component typed by an equivalent data type</td>
<td>Outport</td>
<td>Output variable</td>
</tr>
<tr>
<td>Port typed by block with an inout flow property typed by block (indirectly) specializing ConservedQuantityKind (physical interaction)</td>
<td>Component typed by connector</td>
<td>Connection port</td>
<td>Node typed by domain</td>
</tr>
<tr>
<td>Block (indirectly) specializing ConservedQuantityKind (physical interaction)</td>
<td>Connector</td>
<td>N/A</td>
<td>Domain</td>
</tr>
<tr>
<td>PhSVariables on blocks (indirectly) specializing ConservedQuantityKind (physical interaction)</td>
<td>Components of connector</td>
<td>N/A</td>
<td>Variables of domain</td>
</tr>
</tbody>
</table>

### Figure 24: Connectors in SysML

#### 10.8.3 Modelica modeling

SysML connectors correspond to Modelica connect equations, which link components typed by Modelica connectors. This depends on the correspondence between SysML port types and Modelica connectors (see 10.7.8).

The following Modelica code corresponds to Figure 24. It has a model `Example` with two components `s1` and `s2` of types `SpringA` and `SpringB`, respectively. The models `SpringA` and `SpringB` have two components `p1` and `p2` of type `Flange`, defined similarly to `Spring` in Subclause 10.7.8. `Model` contains a connect equation linking component `p2` of `s1` to component `p1` of `s2`.

```model Example  
  SpringA s1;  
  SpringB s2;  
  equation  
    connect(s1.p2, s2.p1);  
end Example;  ```
The implementation:
Cameo Systems Modeler 19.0 SP3

Simulink export
- BDD and IBD -> Simulink blocks
- Statemachines -> Stateflow
- Parametrics -> S-functions or Simscape (acausal)
- Diagram layout
- Black-box and/or full implementation

Modelica export
- BDD, IBD, Statemachines, Parametrics
- Variables and parameters
- Time derivatives (der(x))
- Dymola diagram layout annotations
- Standard Modelica connectors
- Units and quantity kinds
SysML to Simulink/Modelica

System architecture and implementation