Co-simulation using OMSimulator

Lennart Ochel
10th OpenModelica Annual Workshop, February 5, 2018
OMSimulator

- Co-simulation environment primarily based on FMUs
- Combining TLM and FMI approaches for co-simulation

<table>
<thead>
<tr>
<th>TLM</th>
<th>FMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical connections</td>
<td>Signal connections</td>
</tr>
<tr>
<td>Delayed connections</td>
<td>Non-delayed connections</td>
</tr>
<tr>
<td>Distributed processes</td>
<td>Single process</td>
</tr>
</tbody>
</table>
OMSimulator

<table>
<thead>
<tr>
<th>OMTLMSimulator</th>
<th>OMFMiSISimulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Physical connections</td>
<td>• Signal connections</td>
</tr>
<tr>
<td>• Delayed connections</td>
<td>• Non-delayed connections</td>
</tr>
<tr>
<td>• Distributed processes</td>
<td>• Single process</td>
</tr>
</tbody>
</table>
OMSimulator

OMSimulatorLib.lib

OMSimulator.exe

OMTLMSimulator
- Physical connections
- Delayed connections
- Distributed processes

OMFMISimulator
- Signal connections
- Non-delayed connections
- Single process
OMTLMSimulator

- Dedicated talk at MODPROD
- Bottom-up approach

MODPROD Parallel Session 2a: TLM-based co-simulation using FMI and direct tool connections

MODPROD Tutorial 2: FMI for Composite Modelling, Co-Simulation and Model Exchange

Diagram:
- Master
- Manager
- Monitor
- TCP/IP
- Wrapper
  - FMU / External Tool
OMTLMSimulator

• Dedicated talk at MODPROD
• Bottom-up approach
• Only TLM connections
• Distributed simulation
• External tool integration (e.g. Simulink, Adams, BEAST)

MODPROD Parallel Session 2a:
TLM-based co-simulation using FMI and direct tool connections

MODPROD Tutorial 2:
FMI for Composite Modelling, Co-Simulation and Model Exchange
OMFMISimulator

- Simulator for connected FMUs
- Only signal connections
- Top-down approach
- Single process
- Scripting interface

MODPROD Tutorial 2:
FMI for Composite Modelling, Co-Simulation and Model Exchange
OMFMISimulator

Python interface:

```python
m = oms.newModel()
# instantiate FMUs
oms.instantiateFMU(m, "A.fmu", "A")
oms.instantiateFMU(m, "B.fmu", "B")
oms.instantiateFMU(m, "C.fmu", "C")
oms.instantiateFMU(m, "D.fmu", "D")
# add connections
oms.addConnection(m, "A.u", "C.y1")
oms.addConnection(m, "B.y", "D.u1")
oms.addConnection(m, "C.y2", "D.u2")
oms.addConnection(m, "C.u", "D.y")
```

MODPROD Tutorial 2:
FMI for Composite Modelling, Co-Simulation and Model Exchange
OMSimulator
OMSimulator - Roadmap

TLM Master (SKF) → OMTLMSimulator (SKF/OSMC) → OMSimulator (OSMC)

OMFMISimulator (OSMC) → OMSimulator (OSMC)

Version 1.0, shipped with OpenModelica 1.12
Version 2.0, shipped with OpenModelica 1.13
OpenModelica FMI Export
Model Structure

Required
• Outputs
• Derivatives
• InitialUnknowns

Optional
• Dependency information
Model Structure

Example: Modelica.Blocks.Continuous.Integrator

Initial Unknowns are missing completely:

1: y
2: der(y)
3: _D_outputStateAlias_y
4: u

OpenModelica Compiler OMCompiler v1.13.0-dev.397+g17ce08d
FMU State

- All values that are needed to continue a simulation.
- Optional: fmi2GetFMUstate / fmi2SetFMUstate
- Needed to implement rollback mechanism
**FMU State**

Possible Scenario (simulating from $t$ to $t + h$)

- Input(B) := Output(A)
- Call doStep for FMU A
  - $t \rightarrow t + h$
- Call doStep for FMU B
  - $t \rightarrow t + h$
FMU State

Possible Scenario (simulating from $t$ to $t + h$)

- Input(B) := Output(A)
- Call doStep for FMU A
  - $t \rightarrow t + h$
- Call doStep for FMU B
  - FMU B failed to compute whole step
    - $t \rightarrow t^*, \ t \leq t^* < t + h$
  - Both FMUs are out of sync
Summary & Outlook
Available features

- Simulation, basic MA, no rollback yet
- FMI 2.0 import for ME & CS
- Lookup tables (mat & csv)
- Detection of fake loops & handling of algebraic loops
- Scripting support (Python & Lua)
Work in Progress

- Integration of TLM & FMI
- Common API
- Graphical user interface (OMEdit & Papyrus)
- SSP support
Outlook

• Parallel execution

• Step size control

• Advanced MA (utilizing roll-back mechanism)
Download

• New member of the OpenModelica tool set
  – First stable version is included in OpenModelica 1.12

• Latest (possibly unstable) versions:
  – OMTLMSimulator: https://github.com/OpenModelica/OMTLMSimulator/
  – OMSimulator: https://github.com/OpenModelica/OMSimulator/