Transformation from graphical model representations into SMP2 models

Röhnsch, A.; Berres, A.; Maibaum, O.; Schumann, H.

Content

- Introduction
- General Transformation Process
- Exemplary model transformation
- Comparison
- Conclusion

Introduction

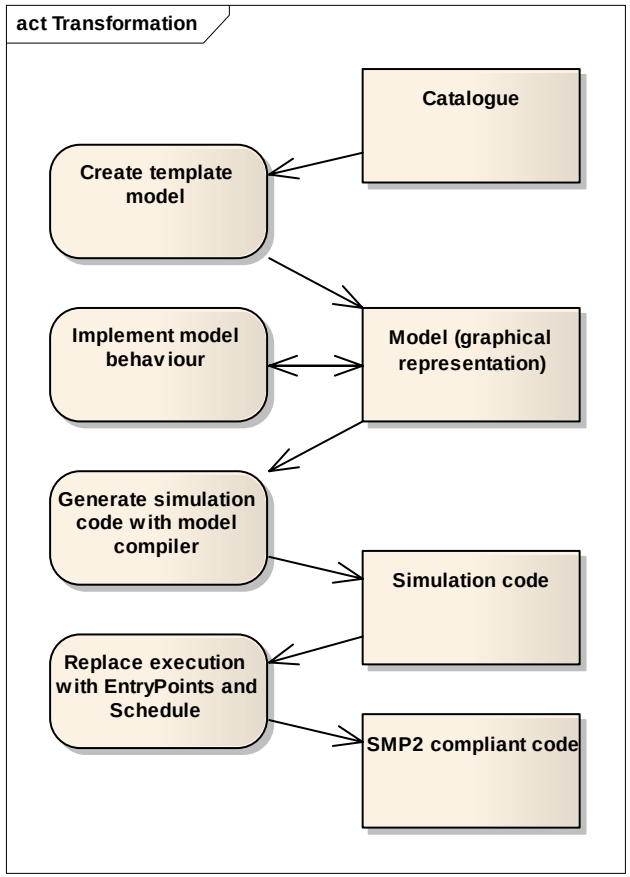
- Create SMP2 models
- Standard way of model implementation
  - Provide SMP2 Catalogue
  - Use SMP2 Language Mapping to create Wrapper code
  - Implement model code using C++
- Disadvantages
  - C++ is too generic for engineering task
  - Engineers not necessarily trained in C++
  - Automatically generated and hand-crafted code need to be put together

Introduction

- Use graphical editor for model implementation
  - Block diagrams commonly used
  - A block diagram is a graphical system representation
  - Domain-specific concepts
  - Engineers familiar with block diagrams and tools
- NLR‘s MOSAIC allows use of Simulink
  - Real-Time Workshop (RTW) creates simulation code
  - MOSAIC uses simulation code to build SMP2 model
  - Version problems with new versions of Simulink and RTW
- How may this work for other environments?

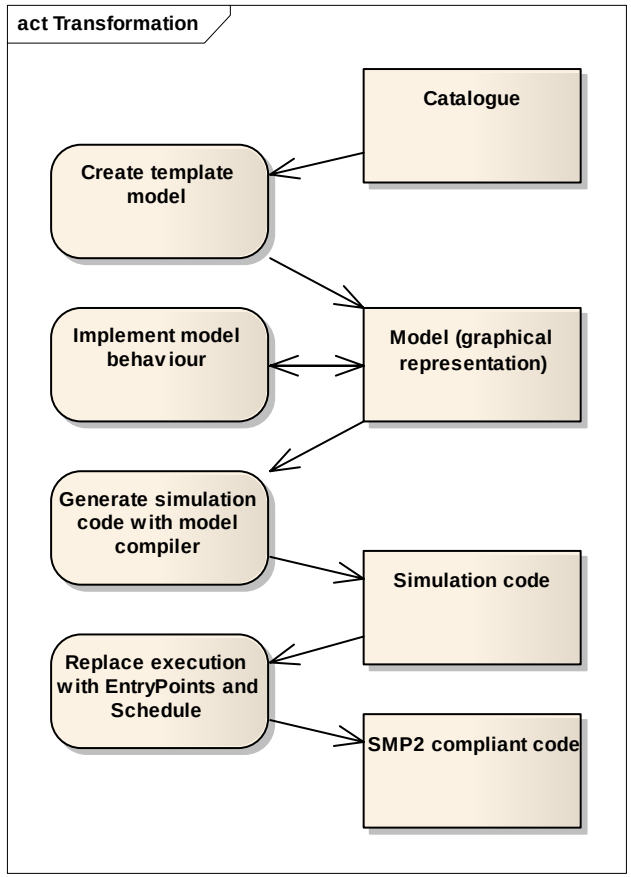
General Transformation Process

- Model structure given as Catalogue
- Provide structure as model representation used by the graphical modeling tool
- Engineers use the graphical modeling tool to define the model‘s behaviour
- Generate simulation code
- Prepare code for SMP2 execution



General Transformation Process
Generation of simulation code

- Compiler tool that works on model representation of graphical model tool
- Code composed of runtime library and model code
- replace simulation execution by SMP2 EntryPoints and an Assembly/Schedule
- Publish model variables



Exemplary model transformation
Modelica

- Object-oriented modeling language
- For modeling of complex physical systems

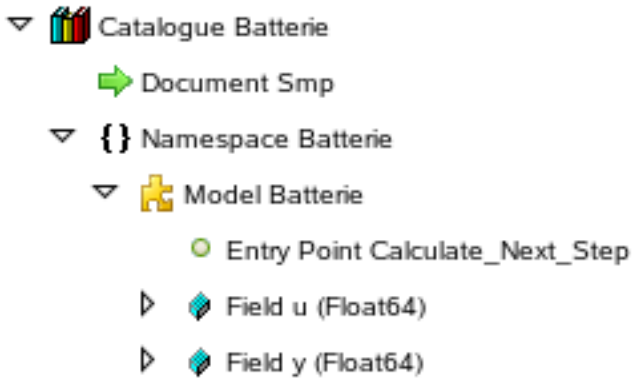
- Started in 1996 with experience from similar languages
- Language specification on version 3.0 since 2007

- Model library with standard components

- New components by reuse or creation

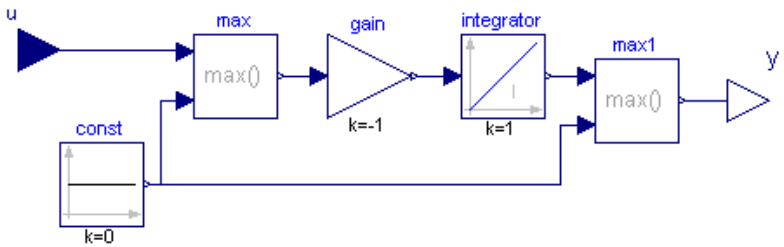
- Define models by using block diagrams

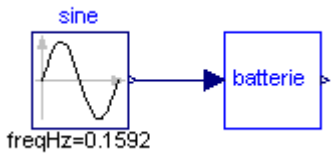
Exemplary model transformation
Model structure




```
model Batterie
  Modelica.Blocks.Interfaces.RealInput u;
  Modelica.Blocks.Interfaces.RealOutput y;
end Batterie;
```

Exemplary model transformation
Implementation of behaviour






```
model Batterie
  Modelica.Blocks.Interfaces.RealInput u;
  Modelica.Blocks.Interfaces.RealOutput
   y;
  Modelica.Blocks.Sources.Constant
   const(k=0);
  Modelica.Blocks.Math.Max max;
  Modelica.Blocks.Math.Gain gain(k=-1);
  Modelica.Blocks.Continuous.Integrator
   integrator;
  Modelica.Blocks.Math.Max max1;
equation
  connect(u, max.u1);
  connect(const.y, max.u2);
  connect(max.y, gain.u);
  connect(gain.y, integrator.u);
  connect(integrator.y, max1.u1);
  connect(const.y, max1.u2);
  connect(max1.y, y);
end Batterie;
```

Exemplary model transformation
OpenModelica

- OpenModelica is an Open Source modeling and simulation environment for Modelica
- In development, does not support full Modelica language specification
- Easy to modify
- OpenModelica Compiler (OMC) creates model code for a simulation run of a Modelica model
- The model code is executed by a runtime library provided with OMC

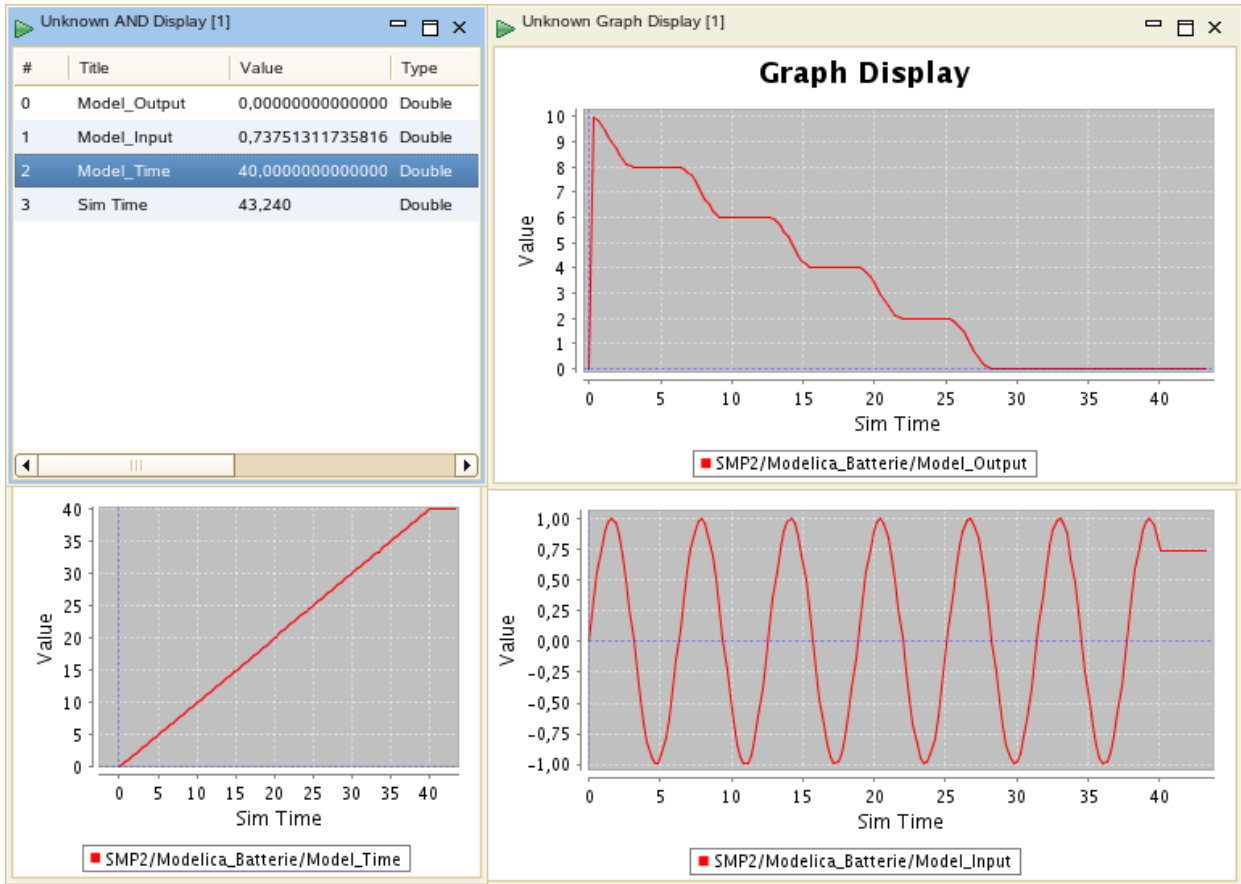
Exemplary model transformation
Execution and Publication

- Regroup execution code in runtime library
- Initialise, Calculate_Next_Step loop, Finalise

- Create SMP2 Wrapper code using the Language Mapping
- Call runtime functions from SMP2 EntryPoints
- Initialise and Finalise called once
- Schedule Calculate_Next_Step similarly to execution in runtime library.

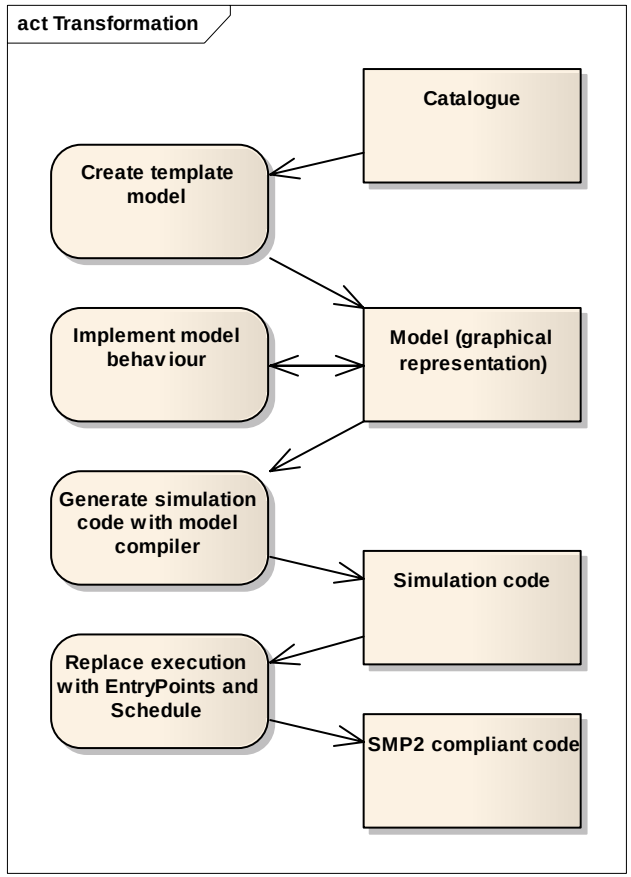
- Publish model variables

Exemplary model transformation
Simulation with SIMSAT



Comparison
Automation of the transformation process

- C++ approach already automated.
  - Trigger Language Mapping code creation
- Actions that need to be automated for the MOSAIC approach:
  - Create Simulink template model
  - Trigger RTW code generation
  - Trigger MOSAIC transformation
- Actions to be automated/developed for the Modelica approach:
  - Create Modelica template model
  - Trigger OMC code generation
  - Develop reliable generic SMP2 transformation



Conclusion

- C++ technically simplest approach for implementation of model behaviour, but not quite suitable
- We want to let engineers use a block diagram approach
- Simulink works with MOSAIC, but has version issues
- Other environments can be used and are worth a try
- For example Modelica

Thank you!