CIM-Compliant Model-to-Model Transformation

For Modelica Models Generation and Power Systems Dynamic Simulations

Francisco J. Gómez¹, Prof. Luigi Vanfretti¹
Svein H. Olsen²

fragom@kth.se, luigiv@kth.se
Electric Power Systems Dept.
KTH
Stockholm, Sweden

svein.harald.olsen@statnett.no
Research and Development Division
Statnett SF
Oslo, Norway

MODPROD, 8TH February 2017, Linköping
Outline

• Introduction & Motivation

• Background on MDSE & Model Transformations
  • M2T, M2M
  • UML & SysML

• Workflow M2M – Research Focus on MDSE
  • Reverse Engineering
  • Binding Semantics: Mapping Meta-Model
  • Power Network SysML Meta-Model

• Conclusions
Exchange of information

- Network planning, power systems operations, demands high degree of coordination and consistency in data exchanges,
- Significantly streamlined through a common data exchange standard
- The exchange of dynamic models provides power system data related to the parameters of an associated block diagram

Harmonization of the different information modeling and physical modeling computer languages attractive to support power system model exchange and dynamic applications
Common Information Model

- IEC CIM Standard based on UML to represent semantic information of a real power system.

- OOP principles, defines all the basic components and topology of the power network.

- ENTSO-E adopts different IEC CIM standards to conform the Common Grid Model Exchange Standard (CGMES).
The *OpenIPSL* Power System Library

- OpenIPSL Modelica library for modeling power grid components.
  - Model components
  - Tested and validated against reference software tools
  - Sample test networks (IEEE models, and others)

- The library makes available standardized and *de facto* standard power systems models usually available in power system tools only accessible through proprietary (and expensive) licenses.
Outline

• Background on MDSE & Model Transformations
  • M2T, M2M
  • UML & SysML

Model transformation – M2T

*Model-2-Text* - “parsers” or “data file format converters/filters”.

Model is implemented and processed using the same programming language used to produce the computation core of the analysis tool.
Adopting the CIM standards, implementation software interfaces to perform model-to-text transformations (M2T) so to adopt the modeling semantics from CIM.
Model transformation – M2M

Model-2-Model transformation processes source models to generate target models,
Endogenous (in-place) – transformation defined within the same modeling language
Exogenous (out-place) – transformation between different modeling languages

model gensal
...
parameter Real wbase = 2 * pi * 50 "system base speed";
parameter Complex Epqp = fpp + a * It;
parameter Real delta0 = arg(Epqp);
parameter Real Pm0 = p0 + (id0 * id0 + iq0 * iq0) * Rs;
real delta "rotor angle";
real w "machine speed deviation, p.u.";
...
initial equation
delta = delta0;
w = 0;
equation
...
der(w) = ((Pm0 - D * w) / (w + 1) - Te) / (2 * H);
der(delta) = wbase * w;
end gensal;
UML & CIM

- **UML**: set of model elements representing an analysis of the properties and behavior of a system.

- **IEC 61970-301 CIM Base UML package** containing static information

- **IEC 61970-457 CIM for Dynamics Profile**, UML package containing dynamic information

- **Limited description of the dynamics information**
SysML & Modelica

- SysML: Add design principles such as requirements modeling and reuse of UML class diagrams as block diagrams supporting parametric modeling interoperability
- Document and design concept model that we want to implement
- Modelica language offers mechanism to implement SysML models
Outline

- Workflow M2M – Research Focus on MDSE
  - Reverse Engineering
  - Binding Semantics: Mapping Meta-Model
  - Power Network SysML Meta-Model

Reverse engineering

- CIM 2 Modelica Factory
  - XML for component mapping,
  - JAVA code implementation

Apply Model-Driven Software Engineering principles:
Let’s formalized this solution!
Model-2-Model Transformation

*Design of a M2M workflow* to generate power system Modelica models generation from CIM

- Automatically generate target Modelica models from source CIM models
- General workflow, represent key actions to implement
Binding Semantics: Mapping meta-model

Identification of relevant attributes from CIM and OpenIPSL

Each OpenIPSL component has mapping rules

Class Structure / Meta-Model to populate CIM values using JAXB (XML/JAVA parser)
Binding Semantics: Mapping Meta-Model

Class to map the connections between CIM classes

Component’s mapping: Specific class for each component and general classes for common attributes
Modelica language stereotypes for models and components.

- **Model** – for a high-level power system model
- **Class** – for component-level power system model
- **Connector**

Identification of relevant keywords to use for parameter, variable and object declaration

Meta-model, to instantiate component objects
Power Network SysML Meta-Model

One class for each stereotype with the relevant keywords for parameter declaration

Define the declaration of the component’s variables
Power Network SysML Meta-Model

Class implementation for the connections

Class implementation for the high-level models. Store the instances of library components that compose the power system models.
Model-2-Model Transformation

SMIB 9-Bus System
• Conclusions
Conclusions

- Workflow that defines a modeling process that takes advantage of CIM semantics, UML/SysML and Modelica languages.

- Defines a method to complement CIM Dynamics profile with equation-based component model definitions for physical modeling behavior,
  - Using the Modelica language for supporting the CIM, for dynamic simulation analysis.

- Defines of a scalable, modular and reusable mapping between different modeling semantics for M2M transformation.
Thank you!