CIM 2 Modelica Factory

Automated Equation-Based Cyber-Physical Power System Modelica Model Generation and Time-Domain Simulation from CIM

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

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

• Background & Motivation
  – Modeling and Simulation
  – Modelica
  – CIM and Cyber-Physical Systems
• Modelica
  – Language Description
  – MetaModelica
  – CIM/UML to Modelica
• CIM 2 Modelica
  – Initial Conditions
  – Simulations
Motivation

- **Application**
- iTESLA: Innovative Tools for Electrical System Security within Large Areas
- CIM provides standard format for power systems data
- Use of data from TSO
  - Description of data equipment, power systems topology and measurements for model validation

- **Research**
- Development of software architecture supporting transformation from CIM, implementing tools for either translating from CIM to Modelica models
- Development of models of cyber-physical power systems components, communication network components, and other components from other domains
Background

**Modelica**

- Modelica is an OOP for declarative equation based mathematical language
  - Non-proprietary language, suitable for standardization and exchange of models
- Modelica tools, commercial and free of charge

- Electric power steering and controller model

- Thermodinamic Network of the ICE model
  TU Dresden, University of Stuttgart, BMW Group, Germany.
Background

**Common Information Model**

- Conceived for information exchange: power systems topology, equipment, measurements
  - Using UML representation to design a structured data model: Semantic transformation from real world to a model
- Standardization of the model diagrams for cyber-physical components
  - Generators
  - Turbine Governors
  - Capacitors
  - Protections
  - Measurements
- IEC61970 provides standard data model for power systems components
Background SVEIN

**CIM for Dynamics**

* Svein

CIM for dynamics
How the models works with CIM
Synchronous Machine ->
SynchrononousTimeConstantReactance

- **UML model of a cyber-physical system**
  - Description of static behavior

- **Cyber-physical Systems**
  - Describe physical behavior of devices/systems
  - Equation-base systems (DAE and ODE)

- **Cyber-physical systems Equation-based modeling**
  - Class diagram UML
  - Equation code
Modelica

Modeling Language

• Modeling language based on equations, allow specification of mathematical models

• Multi-Domain modeling

• Visual Acausal Hierarchical Component Modeling
  • Physical structure
  • No specification of data flow direction

```model DCMotor
Modelica.Electrical.Analog.Basic.Resistor r1(R = 10);
Modelica.Electrical.Analog.Basic.Inductor i1;
Modelica.Electrical.Analog.Basic.EMF emf1;
Modelica.Electrical.Analog.Basic.Ground g;
equation
  connect(DC.p,R.n);
  connect(R.p,L.n);
  connect(L.p,EM.n);
  connect(EM.p,DC.n);
  connect(DC.n,G.p);
  connect(EM.flange_b,load.flange_a);
end DCMotor;
```
Modelica

Modeling Language

- Typed Declarative Equation-based Textual Language
- Object-Oriented Language with class concept
  - Reuse of classes
  - Reuse of components
  - Scalable and Modular models
- Decoupling the model from the solver

```model GENROU
  parameter Complex It = conj(S/VT) "Some comments here";
  parameter Complex Is = It + VT/Zs;
  parameter Complex fpp = Zs*It;
  parameter Real ang_P = arg(fpp);
  parameter Real ang_I = arg(It);
  parameter Real ang_PI = ang_P - ang_I;
  parameter Real psi = 'abs'(fpp);
  equation
    der(Epq) = (1/Tpd0)*(Efd0 - XsdIfd);
    der(Epd) = (1/Tpq0)*(-1)*(XaqIlq);
    anglev = atan2(p.vi, p.vr);
    Vt = sqrt(p.vr^2 + p.vi^2);
    anglei = atan2(p.ii, p.ir);
    I = sqrt(p.ii^2 + p.ir^2);
  end GENROU;
```

Variable declaration

DAE and ODE Equations
Power Systems Library in Modelica

- The FP7 iTESLA project develops a high level library for modeling power grid components
  - Generators,
  - Governors,
  - Controls,
  - Branches,
  - Loads,
  - Buses,
  - Events
- The library makes available standardized power systems models usually available in power system tools only accessible through proprietary (and expensive) licenses
Modelica

**CIM / UML to Modelica**

Modelica provides data definition and compilers for equation based modeling

ModelicaML is a tool to create UML definition for Modelica models

Design of classes, components and models using a data model representation:

- Definition of start values for components and definition of mathematical equations
- Code generation creates classes and models with relation between classes
**CIM / UML to Modelica**

- Semantic transformation for automatic simulation directly from CIM definition
CIM 2 Modelica

Process flow design

- Automatic generation of Modelica code from CIM/UML definition
- Manual design of CIM/UML definition and Mapping
- Loading CIM/XML and Mapping
- Semantic transformation into Modelica code:
  - Set initial values from load flow solution
  - Set connection between classes
CIM 2 Modelica Mapping

- Relation between CIM classes and Power system library classes
- CIM Attributes and values -> Modelica Variables and starting values
- CIM relations between classes -> Modelica connection between components
  or
- CIM relations between classes -> Use of Modelica classes as objects
**Simulation Engine**

Open-source software for cyber-physical system simulation
Plug-in different compilers and solvers

- JAVA
- HDF5
- PYTHON
- OMC
- JM
- Dymola

Diagram showing the workflow from load model to simulation results with different inputs and outputs.
Thanks for your attention!
and...

That's all Folks!