An framework for model-driven product design and development using Modelica

Adrian Pop, Olof Johansson, Peter Fritzson
Programming Environments Laboratory (PELAB)
Department of Computer and Information Science (IDA)
in collaboration with Department of Mechanical Engineering (IKP)
Linköping University (LiU)
Outline

- Introduction
- Architecture Overview
- Framework Details
  - Modelica & ModelicaXML
  - Modelica Model Database ModelicaDB
  - FMDesign
  - Selection & Configuration tool
  - Automatic Model Generation tool
- Conclusions & Future Work
Introduction

- Product design
  - product concept modeling and evaluation
  - physical modeling and simulation

- Integration of
  - conceptual modeling tools and
  - modeling and simulation tools
Product Design phases

- Generating requirement and desirables specification
- Concept generation
- Concept selection
- Parameter calculations
- Analysis and evaluation
- Sensitivity and trade-off analysis
- Identification of key Life cycle value factors

FM Design Tool
Modelica Design Tool
FM Design Database
Modelica Design Database

Modelica Integration
- **Declarative language**
  - Equations and mathematical functions allow acausal modeling, high level specification, increased correctness

- **Multi-domain modeling**
  - Combine electrical, mechanical, thermodynamic, hydraulic, biological, control, event, real-time, etc...

- **Everything is a class**
  - Strongly typed object-oriented language with a general class concept, Java & Matlab like syntax

- **Visual component programming**
  - Hierarchical system architecture capabilities
class Test "comment"
    Real x;
    Real xdot;
    equation
        xdot = der(x);
end Test;

<modelicaxml>
    <definition ident= "Test"
        comment="comment">
        <component ident="x" type="Real"
            visibility="public" />
        <component ident="xdot" type="Real"
            visibility="public" />
        <equation>...</equation>
    </definition>
</modelicaxml>
ModelicaDB - Modelica Model Database

- is populated with simulation models by importing their ModelicaXML representation

- provides
  - simulation models repository
  - search and organizational features
  - flexibility and scalability
  - collaborative development
Example: design phases of an Aircraft Product

- aircraft conceptual model in FMDesign
  - decomposition of the aircraft into functions and means
  - mapping between means and Modelica simulation components (Implementation Tree)
- simulation of various design choices
- choosing the best design choice using the simulation results
The Selection and Configuration Tool

- part of FMDesign and coupled with
  - ModelicaDB for searching capabilities
  - modeling tools (MathModelica, Dymola, OpenModelica) for creating/editing simulation models (components)

- provides
  - search for simulation models
  - creating/editing simulation models in external Modelica tools
  - configuration dialogs for selected simulation models for specific means implementation
Simulation Components for an Aircraft Product
The Automatic Model Generator Tool

- part of FMDesign
- traverses an Implementation Tree (of type simulation)
- generates ModelicaXML models that are translated back into Modelica
- calls external simulation tools for simulation
- feeds the simulation results back to the designer to help him/her choose the best design choice
Aircraft simulation & visualization
Conclusions

- Conceptual Modeling (FMDesign) combined with Modeling and Simulation tools (Modelica)
- A prototype is currently in development at Linköping University

Future Work

- Using our ModelicaXML composition framework in the automatic model generator tool
- Using Ontologies based on Semantic Web languages (Description Logic) for product concept design and classification of component libraries
- Automatic selection of best design choice based on simulation results and requirements
Thank you!

Questions?
Analysis model of a function means design system prototype. It is a summary of the design FMDesign V1.2a.

Architecture of the FM design system, showing inheritance and attributes of classes.