Towards Configuration Support for Collaborative Simulator Development

– A Product Line Approach in Model Based Systems Engineering

CoMetS 2011, Paris

Henric Andersson, Magnus Carlsson, Johan Ölvander

Dept. of Management and Engineering
Linköping University, Sweden

&

Saab Aeronautics, Linköping, Sweden
Presentation of the presenter

- MsC in Control Engineering (1994)
- Modeling and simulation of power plants at ABB
- Saab Aeronautics
  - Flight Control
  - Project Management / Systems Engineering
  - Product Line Engineering
  - Research / PhD (2005 -> 2012)
Content

- Context - Simulator Usage
- Model Based Systems Engineering
- Product Line Engineering
- Example application
- Configurator implementation
- Conclusions & Further work
Simulator example 1(3)

Engineer in the loop; Development, Training
Simulator example 2(3)

Pilot in the loop; Training, verification
Simulator example 3(3)

Desktop; batch simulation
Simulator creation from models
The Product Line Approach – a concept for reuse
MODEL BASED DEVELOPMENT

- Global Scheduler
- Local solvers
- Model update rate 1-100 Hz
THE VALUES OF MBSE

Model Based Development

Traditional Development

Maturity
100%

Verbal definition  Document  Analysis  Simulation  Test Flight/veriﬁcation  Training/Usage

Model  Review  Prototype
Model overview and classification
FLOW OF SIMULATION MODELS

Origin of models:

- Deliverables from sub suppliers (e.g. Engine)
- In-house development of aircraft specific models (e.g. Mass & Inertia)
- In-house development of simulator specific models (e.g. Cockpit)
Challenges in Aircraft Simulation

Some challenges in set-up and support of large-scale simulations:

• Different operating systems & simulation platforms

• Many models ~100 including “legacy codes”

• Variants of the systems that the models represent

• Variants of “the same” model, e.g. different levels of fidelity

• Versions of models, e.g. due to error correction

• Parametric models with different sets of System Parameters

• Lack of standards & tools for collaboration
Basic components of M&S Software Product Line (SPL)

- Basic Mean: Assets with variations.
- Variation methods:
  - Model variants. A “variant master” describe their properties
  - Configurable models. Switches to instantiate desired behavior
Collaboration aspects of the SPL infrastructure

- Collaboration at model supplier interaction
  - Architectural requirements and standards collaboration/agreement
  - Transparency of development status

- Collaboration at customer interaction
  - Early validation of product functions and properties
Binding time alternatives

Model store

Check-out → Compile → Execute

Variation binding and product instantiation

Software production/build process

Software assets with variations

Decisions about variations for a product instance

Variation binding and product instantiation

Products

Binding time alternatives

%Source code
pragma
if alpha then
good
else
bad
endif
## Binding Time overview

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Check-out-time</th>
<th>Compile-time</th>
<th>Run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates</td>
<td>Different source code variants</td>
<td>Different object code variants</td>
<td>Different instances</td>
</tr>
<tr>
<td>Used for</td>
<td>Reliable configuration</td>
<td>Implementation oriented configuration</td>
<td>Fast reconfiguration</td>
</tr>
<tr>
<td>Example</td>
<td>When security / IRP aspects is important</td>
<td>Target / platform variation</td>
<td>Reconfiguration at end-user site</td>
</tr>
</tbody>
</table>
Configuration & Customization
System Architecture

Valid product data & structures (PDM)

Simulation Models

Feature Model
Constraints

Inference Engine

Model Meta information

Simulator Configuration
Product Variant Master – an analyze
- the first step towards a configurator prototype

Object oriented model of model variants
- Objects to represent model variants
- Attributes to repr. variation points
- Rules for valid / not valid combinations
Prototype implementation
– Tacton Configurator Studio (COTS tool)
THE UTILIZATION POINTS OF MODELS

Insight
Create Understanding and deeper Insight

Automation
Data storage Analyze Simulation Transformation

Collaboration
Communication Visualization Decision support Documentation

??..!!
Conclusions and further work

Conclusions
• Constraints input from PDM for integration of configuration data between PDM and the simulation environment
• Model Interface Compatibility is crucial. The emerging FMI (Functional Mock-up Interface) standard is promising for improved collaboration

Further Work
• Develop a robust meta-model for model variability, configuration/customization
• Connection to emerging standards; PLMXML, FMI & SysML
  – Use XML, XSD & XSLT for data storage, exchange, presentation and mapping
• Validate prototype configurator implementation in the application project

Thank You!