Improve Complexity Management with Model-Based Design in V-Modell

Polarion User-Conference 2013
Stuttgart, 01.10.2013
The MathWorks Team Today

Michael Hopfenzitz
Senior Account Manager

Christian Guß
Application Engineer
Agenda

- Introduction
- Complexity in Embedded Software Projects
- MathWorks User-Stories
- Model-Based Design Approach
- Traceability between Polarion Requirements and Simulink
MathWorks at a Glance

- **Headquarters:** Natick, Massachusetts US
- **Other U.S. Locations:** California, Michigan, Texas, Washington, DC
- **Europe:** France, Germany, Italy, Netherlands, Spain, Sweden, Switzerland, United Kingdom
- **Asia-Pacific:** Australia, China, India, Japan, Korea
- **Worldwide training and consulting**
- **Distributors serving more than 20 countries**

Earth's topography on a Miller cylindrical projection, created with MATLAB and Mapping Toolbox
High-Integrity Applications

Software-based systems that are designed and maintained such that they have a high probability of carrying out their intended function

Weinmann Develops Life-Saving Transport Ventilator Using Model-Based Design

**Challenge**
Develop embedded software for an advanced emergency and hospital transport ventilator

**Solution**
Use MATLAB and Simulink for Model-Based Design to model and simulate the controller, generate production code, and streamline compliance certification

**Results**
- Code development and reviews accelerated by 50%
- Dozens of design alternatives explored
- 60% of core design reused

“Modeling, simulating, and implementing the ventilator’s embedded software with Simulink greatly simplified compliance certification. The model helped ensure a structured development process and provided thorough documentation and a visual representation of the system for the certification review.”

Dr. Florian Dietz
Weinmann

Link to user story

The MEDUMAT Transport ventilator. Image © Weinmann Medical Technology.
Complexity Challenges in Software Development

**RESEARCH**

**SPECIFICATIONS**

**REQUIREMENTS**

**DESIGN**

**IMPLEMENTATION**

**INTEGRATION AND TEST**

- **Requirement Documents**
  - Difficult to analyze
  - Difficult to manage as they change

- **Paper Specifications**
  - Easy to misinterpret
  - Difficult to integrate with design

- **Physical Prototypes**
  - Incomplete and expensive
  - Prevents rapid iteration
  - No system-level testing

- **Manual Coding**
  - Time consuming
  - Introduces defects and variance
  - Difficult to reuse

- **Testing**
  - Design and integration issues found late
  - Difficult to feed insights back into design process
  - Traceability
What is Model-Based Design?

Model, Simulate, Verify the control algorithm, Auto-generate C code, Deploy & Test on the embedded hardware
Example: Glycemic Control System

Input

Controller

Plant

Output

Sensor
Executable Specification
Component Design – Subsystems
Model-Based Design
Development Process

Requirements

System Design
- Environment
- Physical Components
- Algorithms

Component Design

Subsystem Design

Implementation
- Embedded Software: C, C++, VHDL, Verilog
- Digital Electronics: MCU, DSP, FPGA, ASIC

Research
- Data Analysis
- Algorithm Development
- Data Modeling

Integration & Test
- System-Level Integration & Test
- Complete Integration & Test
- Integration testing
- Code Verification and Validation
- User Acceptance Testing

System-Level Specification

Subsystem Implementation

Generate
Model-Based Design
Development Process

Requirements
Requirements capturing in Polarion

System Design
- Environment
- Physical Components
- Algorithms

Component Design

Research
- Data Analysis
- Algorithm Development
- Data Modeling

Subsystem Design

Subsystem Implementation
- Embedded Software
- Digital Electronics
- C, C++, VHDL, Verilog
- MCU, DSP, FPGA, ASIC

Integration & Test
- Complete Integration & Test
- System-Level Integration & Test
- User Acceptance Testing
- Code Verification and Validation

System-Level Specification

Integration testing

System Design

Component Design
Polarion Connector for MATLAB Simulink

See: www.polarion.com/connectors/matlab

Extension Requirements
- Polarion 2013
- MATLAB/Simulink 2013a
- Simulink Validation and Verification Toolbox
Highlight Requirements Inside
Traceability from Simulink to Polarion
Traceability from Simulink to Polarion
Traceability from Polarion to Simulink

IPCS-61 - The pump shall begin in a default inactive state, and only leave this state upon receipt of the appropriate Power On command. The infusion rate is 0 at this stage.

Link to Simulink Model

Role | URL
-----|---------------------------------------------
Model-Based Design
Multi-Domain Modeling and Algorithm Development

Methods for modeling systems in different domains

Data Flow (Block diagram)

Physical Modeling (Schematic)

Modeling of Event-Driven Systems (State - Machines)
Model-Based Design
Early Concept Verification

- Executable specifications
- Predict dynamic system behaviour by simulation
  - System & environment models
  - Less physical prototypes
- Use of simulation results for system design
  - Fast What-If studies
  - Short iteration cycles

Idea → Simple Model → Detailed Model
Model-Based Design

Automatic Code Generation

- C/C++, VHDL and PLC-Code Generation from one model
- Support for Fixed Point Data Format
  - Automatic scaling
  - Supported in Simulation and Code-Generation
- Easy integration of legacy C/C++-Code
- System development independent of the target
Traceability: Requirement ↔ Model ↔ Code HTML Code Generation Report
Model-Based Design
Continuous Verification and Validation

- Requirements
- System Design
  - Environment
  - Physical Components
  - Algorithms
- Component Design
- Subsystem Design
- Research
  - Data Analysis
  - Algorithm Development
  - Data Modeling
- Verification and Validation
- System-Level Specification
- Subsystem Implementation
- Subsystem Integration & Test
- Code Verification and Validation
- Integration testing
- Complete Integration & Test
- User Acceptance Testing
Benefits of Model-Based Design

- Models: Core of the Development Process
- Unambiguous Description of Requirements (Executable Specification)
- Fast Evaluation of Design Variants
- Frontloading - Early Test and Verification
- Automatic Code Generation

⇒ Better Cooperation, Communication and Collaboration
⇒ Higher Product Quality
Support and Community

MathWorks® | Application Engineering
MathWorks® | Training Services

MATLAB® CENTRAL

MathWorks® | Consulting Services
MathWorks® | Connections Program

MathWorks® | Technical Support