Team-Based Collaboration in Simulink

Chris Fillyaw – Application Engineer
Detroit, MI
Development of a complex system
Agenda

- Team-based workflow considerations
- Reproducing the design environment
- Enabling and managing design iteration
Model-Based Design workflow

System Model

Controller Model

Motor Model

C & HDL

Mathematical Model

Controller Implementation

Motor Hardware

Embedded System
Complex designs require multiple disciplines
Model-Based Design workflow

System Model

Controller Model

Motor Model

Test bench
Algorithm Spec
Data File

Motor Model
Data File

Controller Implementation

Motor Hardware

Embedded System
Paradigm shift from individual engineer to a team of engineers

• Share information
• Standardize interfaces
• Iterate on design
Selecting a model architecture for collaboration

- Test Cases
- Controller
- Plant

- Implementation Code
  - Referenced Model
  - Data File
- Library
- Data File

Share · Standardize · Iterate
Model architecture info on the web
http://www.mathworks.com/model-based-design/best-practices/

Developing a Model Architecture

Model architecture structures components, interfaces, and data in a way that supports system design goals. These goals may include design clarity, determinism, component reuse, code efficiency, and compatibility with a particular embedded implementation. An effective model architecture lets you trade off competing goals. It also facilitates project management, collaborative and concurrent development, simulation performance, and testing.

With Model-Based Design, the hierarchical organization of the model mimics the architecture of the target software environment. As a result, many architectural decisions, such as software hierarchy, order of computation, and interface dependencies between components, can be made during development.

Learn More

- Paper: Model-Based Design for Large Systems
- Paper: Data Modeling and Management
- Paper: Large-Scale Modeling for Embedded Applications
- Paper: Applying Model-Based Design to Commercial Vehicle Electronics Systems
- Paper: Development of AUTOSAR Software Components Within Model-Based Design
Selecting a model architecture for collaboration

Field-Oriented Control of Permanent Magnet Synchronous Machine
System Test Bench

Model Description: Field-Oriented Control of Permanent Magnet Synchronous Machine

Demonstrates a Field-Oriented Control algorithm with Space Vector Modulation for a Permanent Magnet Synchronous Machine (PMSM). The test bench can be used to evaluate the system performance. Examples include turning the motor on, searching for a valid rotor position, transitioning to closed loop operation, and changing speed and torque during closed loop control. The Embedded Processor subsystem contains the controller algorithm (which supports C code generation) as well as simulation models of peripherals.

Share · Standardize · Iterate
Explore the model architecture
Considerations for reproducing the project across the design team

- Project file management
- Version control
- Project initialization and termination
Simulink Projects

Manage design-related files efficiently within Simulink

- Search, manage, and share related files in a Simulink project
- View revision information
- Peer review of changes using XML comparison tools
- Access version control functionality

Demo: >>Simulink.ModelManagement.Project.projectDemo('airframe');
Simulink Project Shortcuts

Access and execute utility script and key files in a project

- Access and execute utility scripts and key files
- Option to execute scripts at project start-up/shutdown
  - Setup environment
  - Clean up operations
Source Control Integration

Integrate with 3rd party revision control tools

- Basic revision control
  - Lightweight local version control tool for single user
  - Basic integration with command line Subversion (SVN)
- Adaptor SDK free download available for integration between Simulink Projects and 3rd revision control tools

Developed with SDK

Local Version Control  Subversion  Other Version Control Tools
Design iteration for controls design and code generation

- Test Cases
- Controller
- Plant
- Implementation Code
  - Referenced Model
  - Data File

Share · Standardize · Iterate
Ensuring design consistency through modeling standards

- Consistent modeling style
- Model reuse across design tasks
- Meeting standards (MISRA, ISO, MAAB)
Modeling Standards Checking in Simulink
Simulink Verification and Validation

- **Analysis Engine**
  - Model Advisor (Simulink)

- **Checks**
  - Readability and Semantics
  - Performance and Efficiency
  - Change and Configuration Management
  - Potential runtime errors

Model Advisor Interface
Automated checking against modeling standards

**jc_0201: Usable characters for Subsystem names**

**ID:** Title  
jc_0201: Usable characters for Subsystem

**Priority**  
Strongly recommended

**Scope**  
MAAB

**MATLAB Versions**  
All

**Prerequisites**  
None

**Description**  
The names of all Subsystem blocks should conform to the following constraints:

**Form**
- Should not start with a number
- Should not include blank spaces
- Should not include carriage returns

Share · Standardize · Iterate
Managing design iterations using Model Comparison

Subversion Repository

Share · Standardize · Iterate
Merge Simulink Models Based on XML Comparison Differences

Merge Simulink models from within XML comparison report

- Merge models within the tool by merging changes from left to right:
  - Left model is the base
  - Right model is the one edited
- Merge individual parameters, blocks, or entire subsystems
- Compare models with version in the repository

New merge option

Requires Simulink Report Generator

» slxml_sfcar
Summary

- Share information across the team with Simulink Projects
- Ensure design consistency and adhere to standards with Model Advisor
- Manage design iteration with Model Comparison