Model-based Design

Embedded Systems Engineering
Benedikt Huber
WS 2010/11
Overview

- **Introduction**
  - System Theory
  - Model Driven Architecture (MDA)
  - MATLAB/SimuLink Demo
  - Summary
History

- General Systems Theory (GST) was the fundamental underpinning of most commercial software design techniques by the 1970s
- Computer-Aided Software Engineering (CASE) tools developed in the 1980s
- Creating the Unified Modeling Language (UML) in the 1990s
- Model Driven Architecture (MDA) launched in 2001
Challenges

- Exploding code sizes and complexity of software
- Increase the productivity of the individual engineer
- Agility of applications, technologies, platforms
- Coordinate the resources of people with expertise in a wide range of disciplines
When software engineers build systems...

Software-centric design methodologies:
- Treat software as main entity
- Overlook domain knowledge
- Dominate the functional design

UML is not the answer to system design
Overview

- Introduction
- System Theory
- Model Driven Architecture (MDA)
- MATLAB/SimuLink Demo
- Summary
Introduction to System Theory

System theory is an approach to a complex structure that **abstracts away from the particular** physical, chemical, or biological nature of its components and simply considers **the structure they together implement**, in terms of the functional role of individual parts and their contribution to the functioning of the whole.

- Systems theory first originated in biology in the 1920s out of the need to explain the interrelatedness of organisms in ecosystems.
- Systems as interacting components: Purpose, Environment, Interrelations, Boundary/Interfaces, Input/Output.
- Emergence is the way complex systems and patterns arise out of a multiplicity of relatively simple interactions.
Which models are there?

- Scale models
- Partial models
- Process models
- State models
- Etc.

Substitute for direct measurement and experimentation
Model Definition

Representation
- A model represents some *thing*.
- Model and *thing* are connected by a *morphism*.

Abstraction
- The model suppresses irrelevant detail and focuses on important aspects.

Pragmatics
- The model is created for a purpose.

[Herbert Stachowiak, *Allgemeine Modelltheorie*]
"What's a model?"

"The code is the model"

"Manage code and model"

"The model is the code"

"Let's talk models!"
Graphical modeling tools reduce the complexity of model designs by breaking them into hierarchies of individual design blocks.
Overview

- Introduction
- System Theory
- Model Driven Architecture (MDA)
- MATLAB/Simulink Demo
- Summary
Traditional approach for a controller

Development strategy for a digital control system:
- Design controller on paper with block diagrams
- Implement the design in a programming language
- Integrate controller with the plant model
- Adjust parameters of the controller through simulation
- Trash or take: reinterate the design until satisfactory

Rapid prototyping is the **application of productivity tools** to develop working prototypes of control systems in the minimum amount of time.
Model-Based Design

• From Paper-based approach to Executable Model
• Model captures all information about concept, design, implementation
• Model is used in all development stages (Research, Design, Implementation and Verification&Validation)
• Model is contiinusly updated and elaborated
Uses of System Models

- Simulation, Visualization
- Static System Analysis, Formal Verification
- Virtual Fault Injection
- Synthesis of Implementations & Test Suites
- Documentation, Presentation
Key steps in the MBD approach

1. **System identification** (modeling the plant) is an iterative process. By acquiring and processing raw data from a real-world system and choosing a mathematical model to represent the plant behavior.

2. **Controller analysis and synthesis.** Identify dynamic characteristics of the plant model and synthesize an appropriate controller.

3. **Offline simulation.** Simulation allows specification, requirements, and modeling errors to be found early.

4. **Deployment**
MBD: System Simulation

• **Software in the Loop**
  • Non real-time simulation, e.g. using SimuLink
  • Using plant model and system controller model

• **Hardware in the Loop**
  • Real-time simulation of plant model
  • Controller implemented on target (code generation)

• **Simulation Goals**
  • Verification and Validation
  • Collect Data to Improve Models
Advantages of the MBD approach

- MBD provides a **common design environment**, which facilitates general communication, data analysis, and system verification between development groups.

- Engineers can **locate and correct errors early** in system design, when the time and financial impact of system modification are minimized.

- **Design reuse**, for upgrades and for derivative systems with expanded capabilities, is facilitated.
Model Driven Architecture (MDA)

- OMG Standard: Model Driven Architecture (MDA).
- **Application of models and generators** to improve the software development process
- The objective is **NO total automation**, but a reasonable proportion

**Goals**
- Speed up the development process: „automation through formalization“
- Tackle the complexity challenge through abstraction
MDA Process

(1) Computation Independent Model (CIM): informal specification

(2) Platform Independent Model (PIM): model which is independent from a specific computational platform (for business processes)

(3) Platform Specific Model (PSM): model which is dependent on a particular architecture, service definitions

(4) Code model and target platform

- MDA defines conceptual separation between models and the transformation between models:
  - Model-to-model transformation (MOF QVT)
  - Model-to-code transformation
Overview

- Introduction
- System Theory
- Model Driven Architecture (MDA)
  - MATLAB/Simulink Demo
- Summary
Summary

- What is a model?
- Key steps in the MBD
- Model Driven Architecture
- Model-based System Design with Simulink/Stateflow
THE END

Thanks for your attention!