Model-Based Design of Industrial Applications
Invited Speaker Session

Swiss Federal Institute of Technology
Zurich, Switzerland.

Dr. Mohamed Anas
Manager, Engineering Group
MathWorks GmbH

September 11, 2015

Agenda

- Overview of Industrial Model-Based Design Process
  - Benefits of Modelling and Simulation, Case Study
- Automatic Code Generation
  - Model elaboration for code generation
- Concluding Remarks
**Additional Expectations, Let’s Interact**

- What is code generation?
- How to generate efficient code?
- Making the generated code target aware
- Elaborate on MEX files
- Selecting the modelling domain, textual versus graphical
- Performance profiling the generated code
- How best to debug the code; readability
- Testing the generated code for functional correctness against the model

---

**Three Things to Takeaway!**

1. Verify early through Simulation
2. Elaborate Models Incrementally
3. Code and Data Interfaces!
Things you have done so far

1. System Definition
2. Identify System Components
3. Formulate Equations
4. Construct Simulink Diagram (Controller and Plant)
5. Run the Simulation
6. Tune Controller
7. Validate the Results

Model-Based Design of Industrial Applications

- REQUIREMENTS
- RESEARCH
- DESIGN
  - Multi-Domain Physical Systems
    - Signal Processing
    - Communications
    - Control Algorithms
    - Supervisory Logic
- CODE
  - C, C++
  - VHDL
  - Verilog
  - Structured Text
  - Industrial Control IDE
- INTEGRATION
- CONTINUOUS TESTING 
  - Testing, Verification, Validation and Documentation Technologies

Multi-Domain Modelling, Simulation and Analysis Technologies

Code Generation Technologies
Industrial Example 1

Summary
A decade of efficiency gains

**ABB gains**
- 10+ years of Model-Based Design experience
  - MATLAB/Simulink and Automatic Code Generation
- A lot of things learnt along the way
  - Efficiency gains by getting new products to market faster
  - Expanded product range by 4 times, supported by the same number of platform engineers
  - Automatic code generation eliminates 30-50% of bugs introduced in manual coding

**Things to remember**
- Treat it the same as traditional software development
  - Need the correct processes in place even though it is graphically based
  - Modelling is complex and must be done in the correct way
  - Tools alone will not give you efficiency gains
  - Model-Based Design coupled with the modern controllers allows faster development
  - Even for the low-cost $2 processors!

Complexity is here. Control it. Reduce to simpler steps, adapt ways of working, and leverage the technological changes in computational performance and tools.

Industrial Example 2

Summary

**Model-Based Design with Simulation and Code Generation**
- generates 156k+ lines of source code in 15 minutes
- helps achieve IEC 62304 standard highest software classification level C
- motivates both signal processing and control engineers
- reduces validation time by 80%, 2 weeks down to 2 days
- proves the design with fewer prototypes
- improves time to market
- works well for small and medium enterprises (SME) but only IF
  - it is well integrated in your development process
  - you take care of modelling activities correctly
  - you know what you are doing

**Project: Bellavista**
- Intensive and intermediate care ventilator for neonates, children and adults
- Non-invasive and invasive ventilation
- Smart ventilation modes that adapt to patient characteristics and condition
- Produced in Switzerland by intmedical, sold worldwide since 2009
Model-Based Design Adoption Grid

- **Requirements-based V&V**
  - Virtual Verification & Validation
  - System Validation
  - Fully-leveraged Model-Based Design

- **System Simulation**
  - Closed-loop Simulation
  - Hardware-in-the-Loop
  - Simulation-based Development and Co-Simulation

- **Algorithm Modeling**
  - Textual and Graphical Specification
  - Design Prototyping
  - Textual and Graphical Programming

- **Simulation**
  - Real-Time Testing
  - Production

**Code Generation Adoption**

**Hands-on Walkthrough of the Design Flow**
Example Activities around Code Generation

- Incremental model elaboration
- Defining data representation
  - simulation vs. code generation behaviour
- Generating code
- Automating build processes
- Integrating external code with generated code
- Setting up generated code to interface with components in the run-time environment
- Verifying the generated code

Model-Based Design, Summary

- **Provides a CAE solution** for multidomain system-level development
- **Uses simulation models** to mathematically describe the system
- **Automatically generates code** for real-time testing and implementation
- **Lets you test continuously** throughout the development process