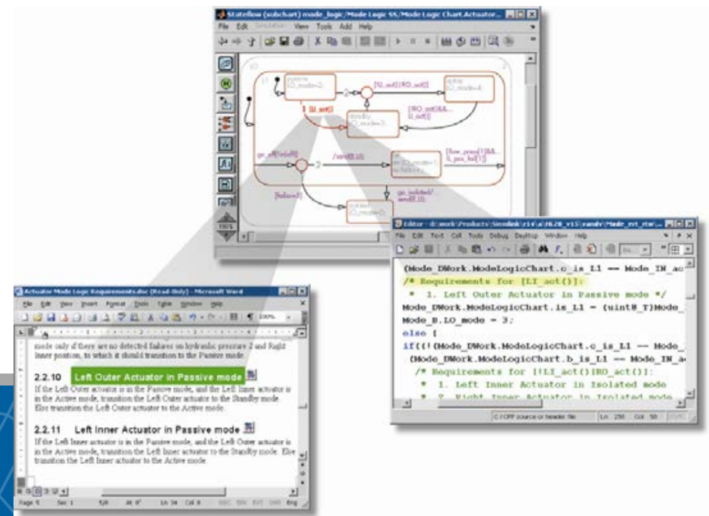
Improve Complexity Management with Model-Based Design in V-Modell



Polarion User-Conference 2013
Stuttgart, 01.10.2013

The MathWorks Team Today

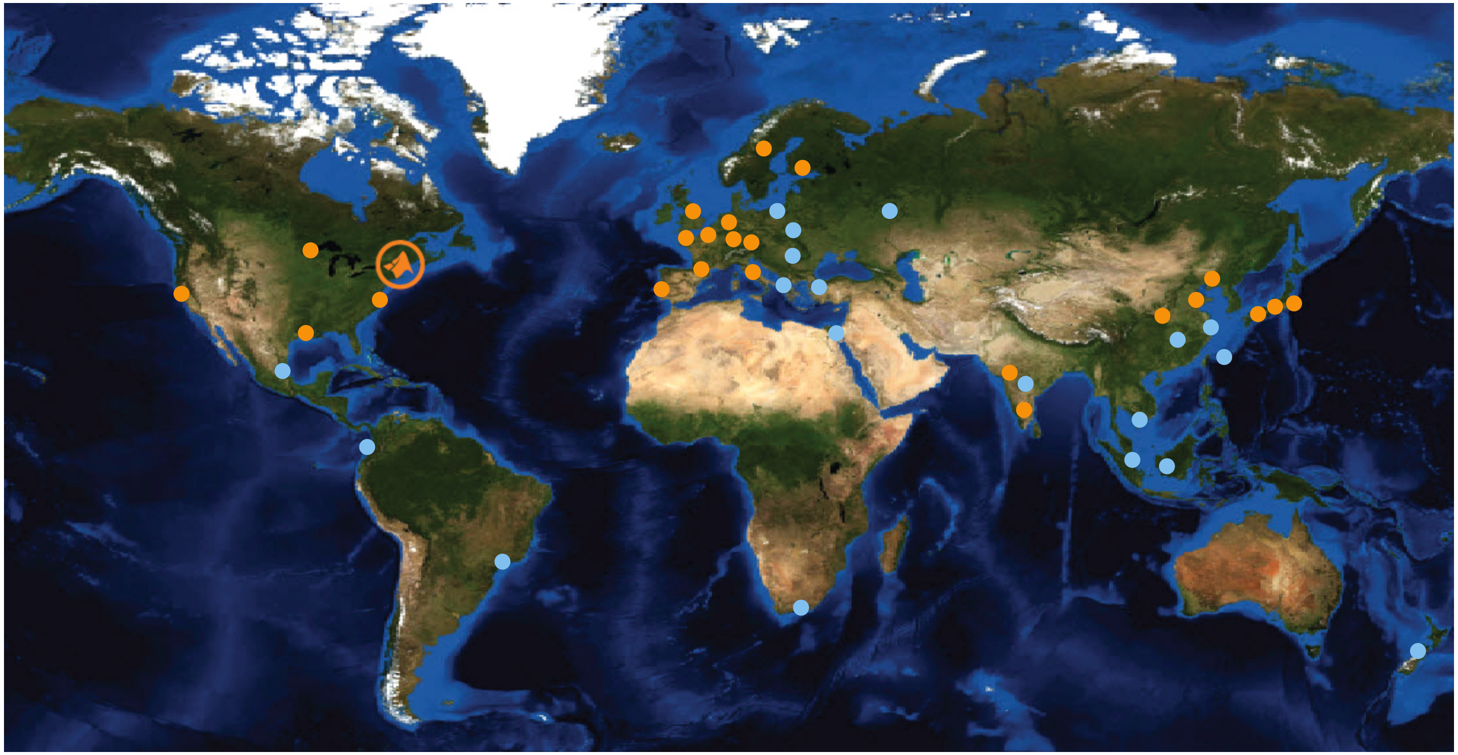
Michael Hopfenzitz
Senior Account Manager

Christian Guß
Application Engineer

Agenda

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

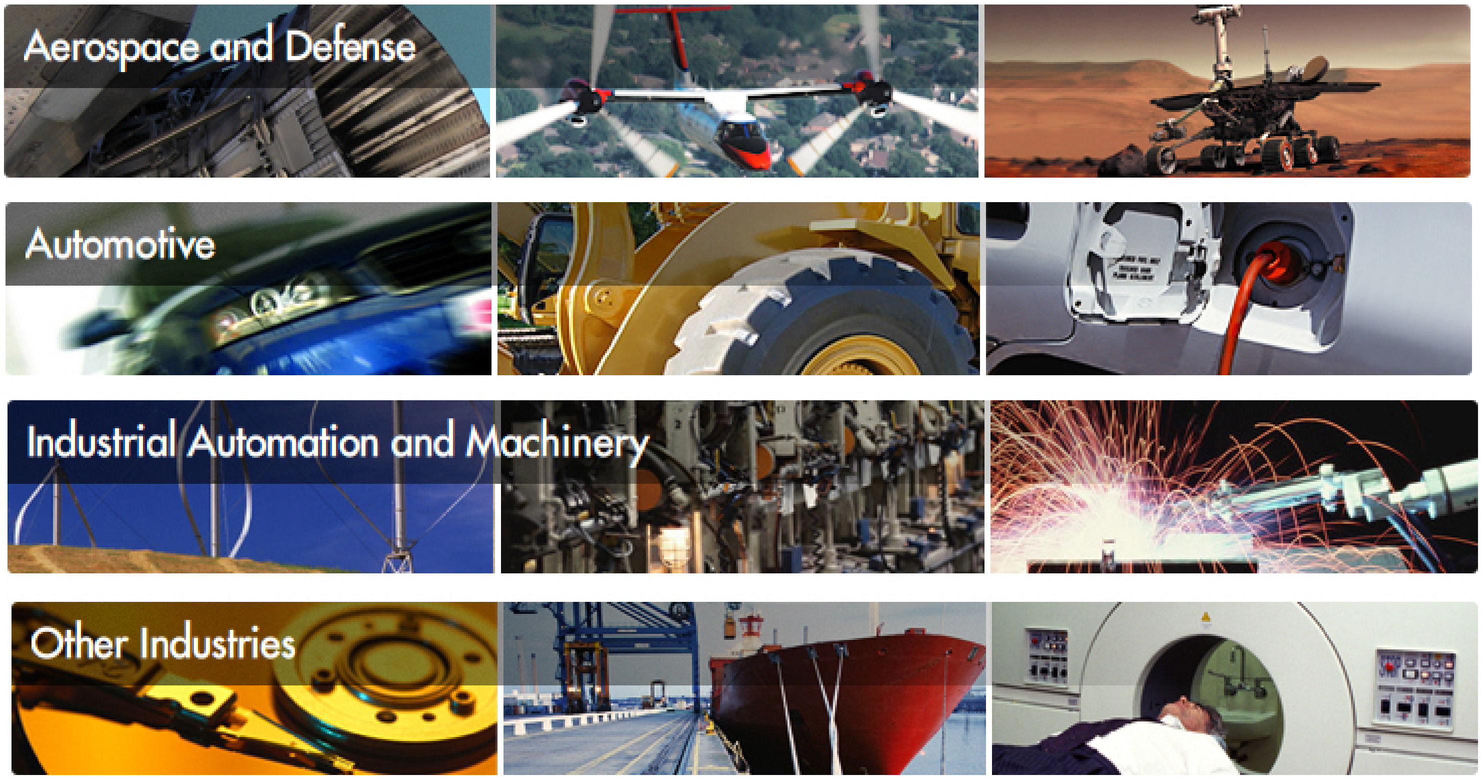
MathWorks at a Glance



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

- **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

High-Integrity Applications



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

cf. Buncefield Investigation Glossary www.buncefieldinvestigation.gov.uk/glossary.htm

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

The MEDUMAT Transport ventilator.
Image © Weinmann Medical Technology.

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

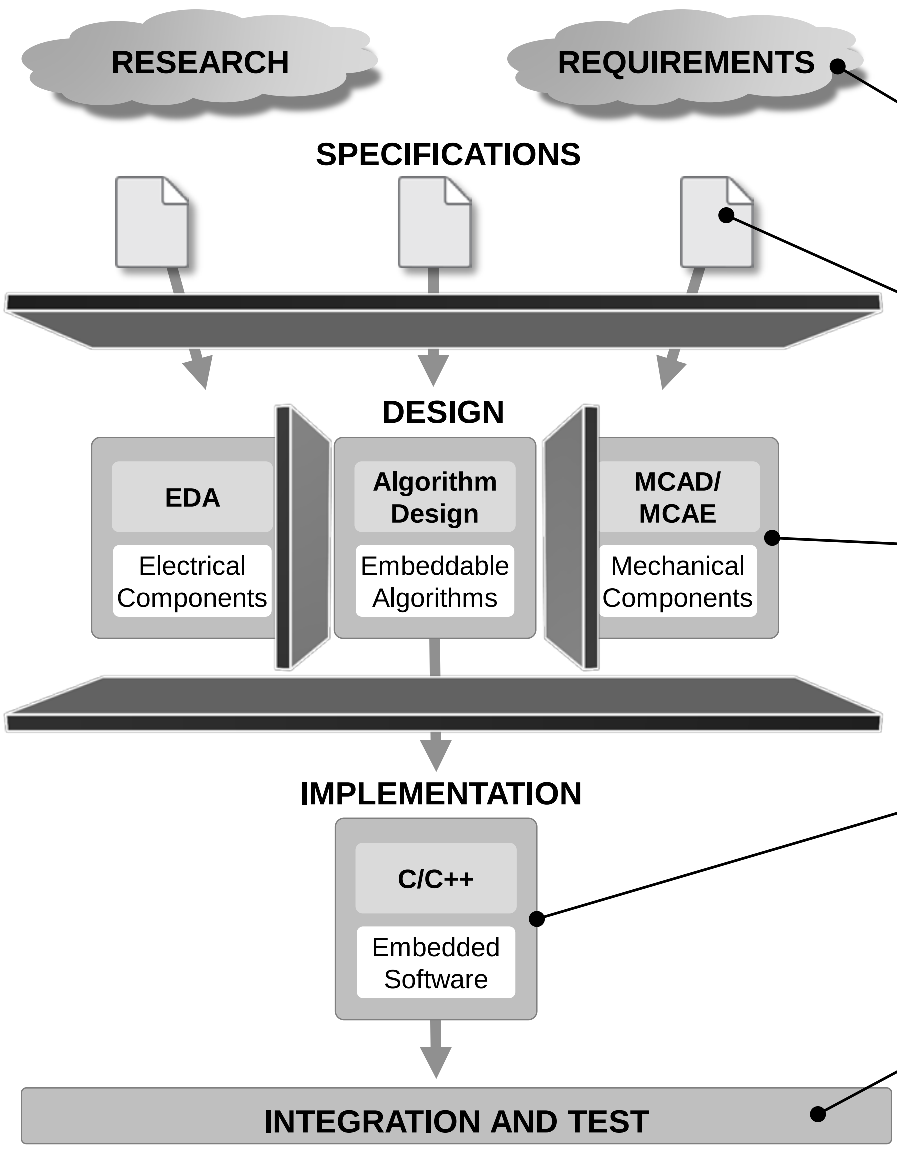
- Code development and reviews accelerated by 50%
- Dozens of design alternatives explored
- 60% of core design reused

"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

Complexity Challenges in Software Development



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

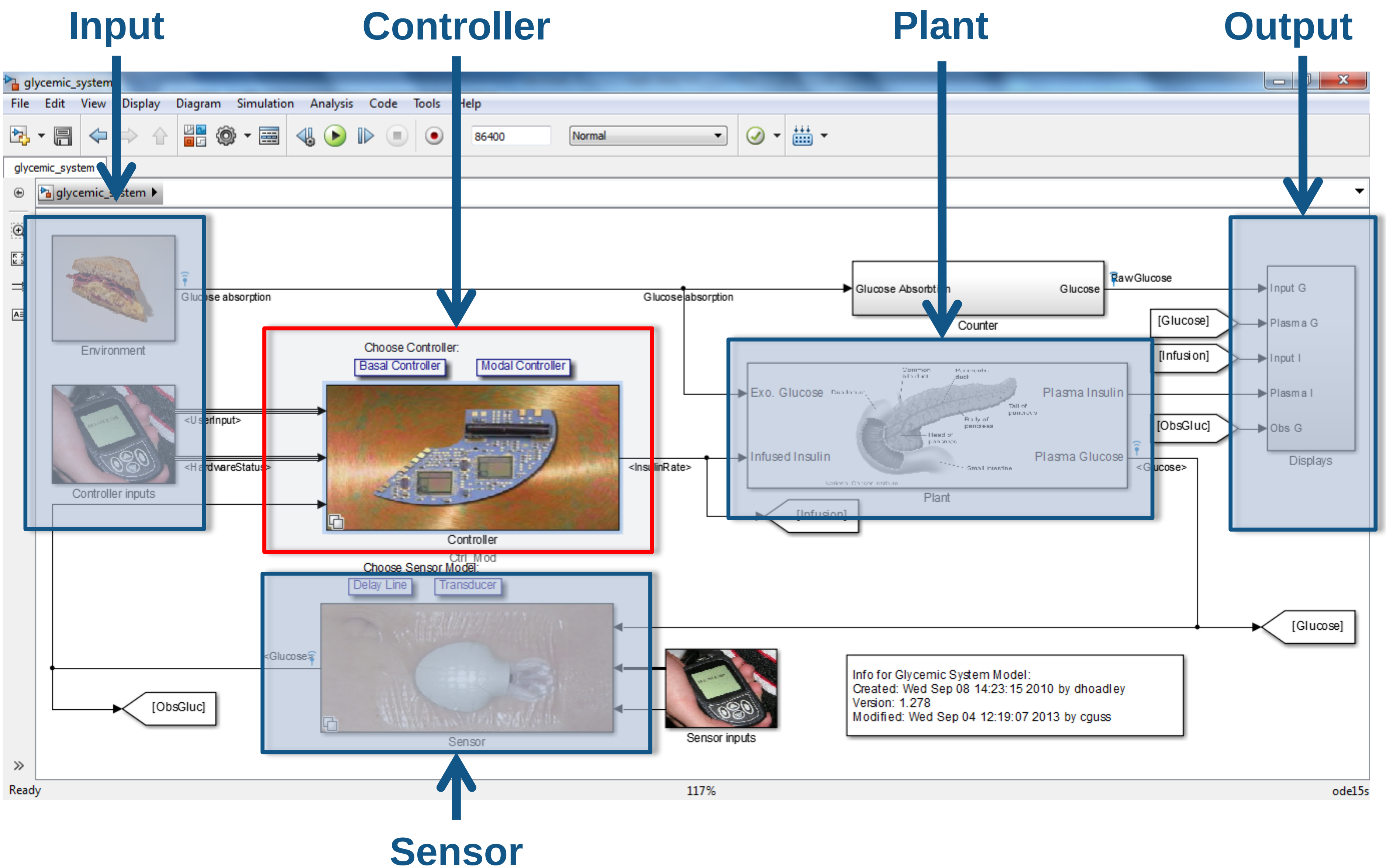
What is Model-Based Design?

Model, Simulate, Verify the control algorithm,

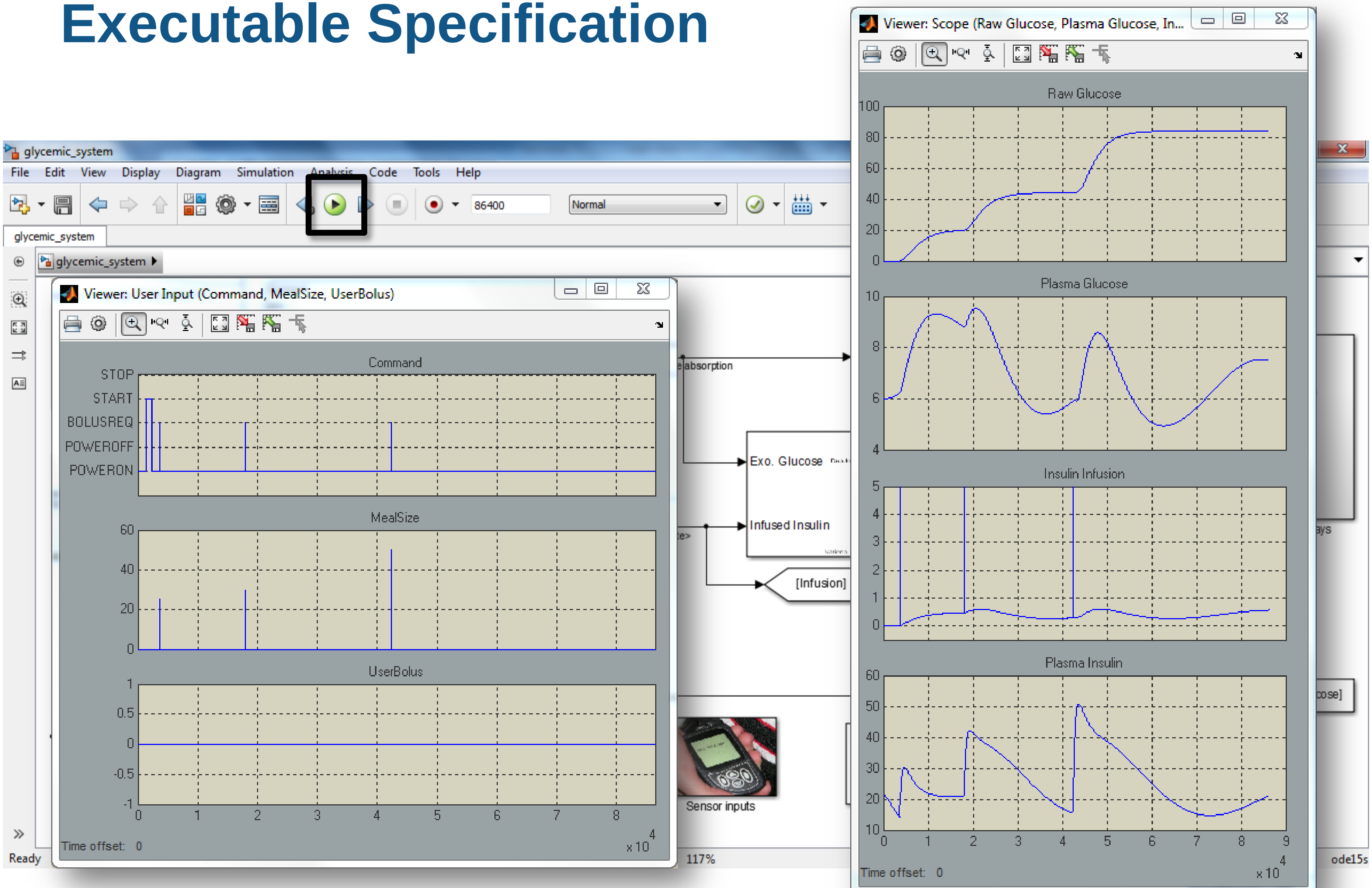
Auto-generate C code,

Deploy & Test on the embedded hardware

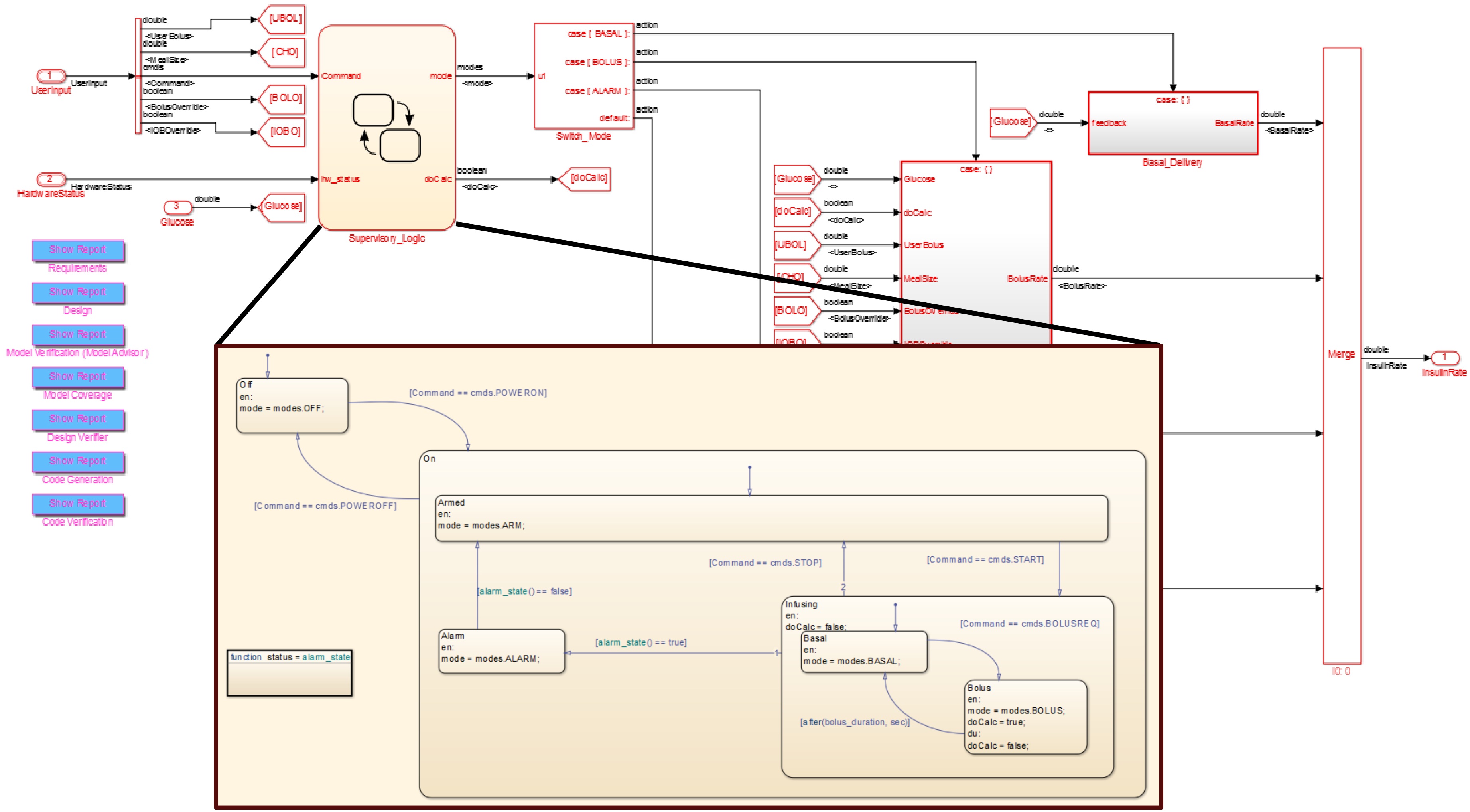
Example: Glycemic Control System



Executable Specification

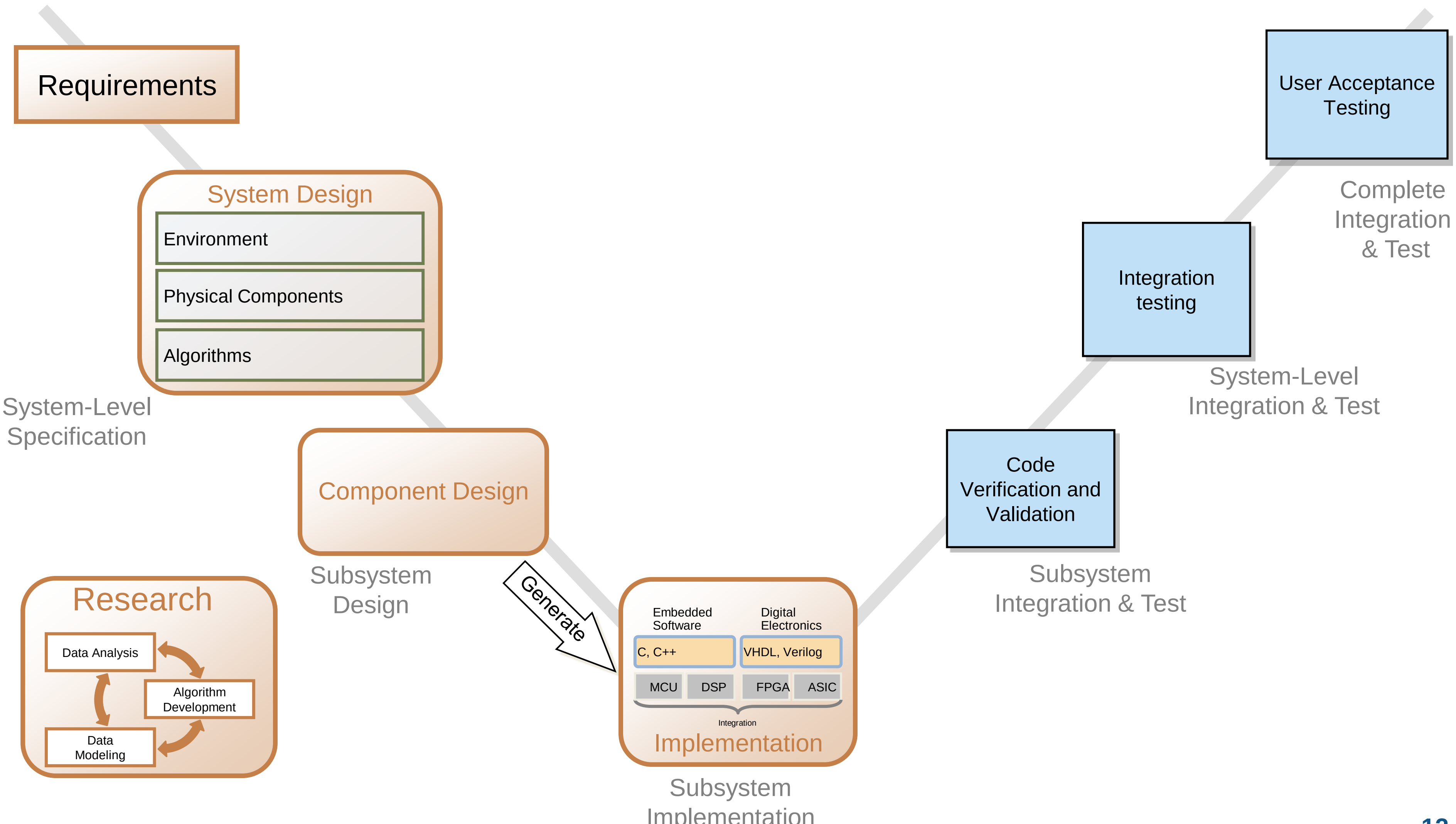


Component Design – Subsystems



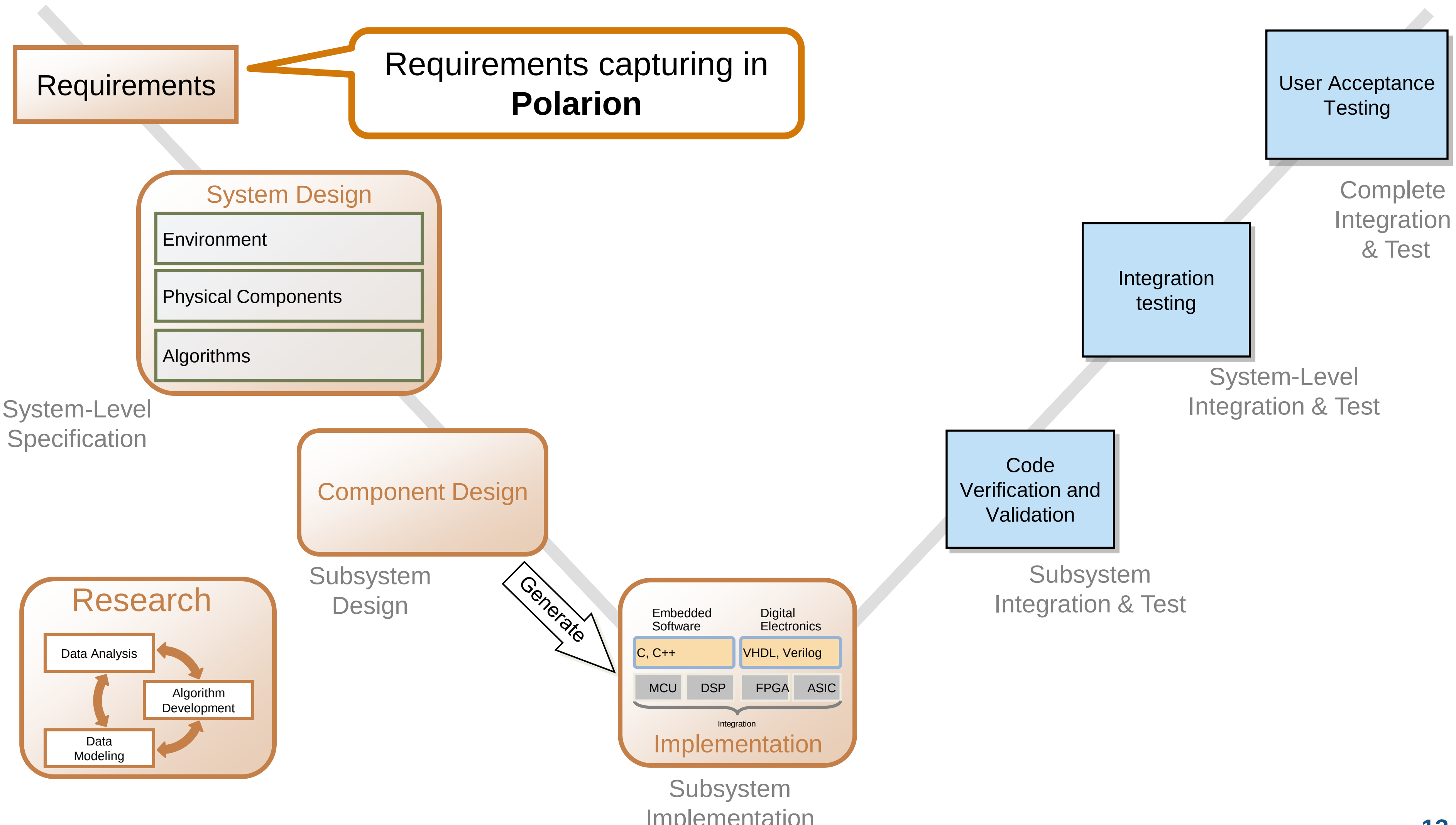
Model-Based Design

Development Process



Model-Based Design

Development Process



Polarion Connector for MATLAB Simulink

Polarion Connector for MATLAB® Simulink

SHARE

by Polarion

Polarion Connector for MATLAB® Simulink lets you link Simulink and State flow models with Polarion requirements and design items.

Added: Tue Jun 04 10:49:55 CEST 2013

Version: 1.0

Categories: Integration

Download

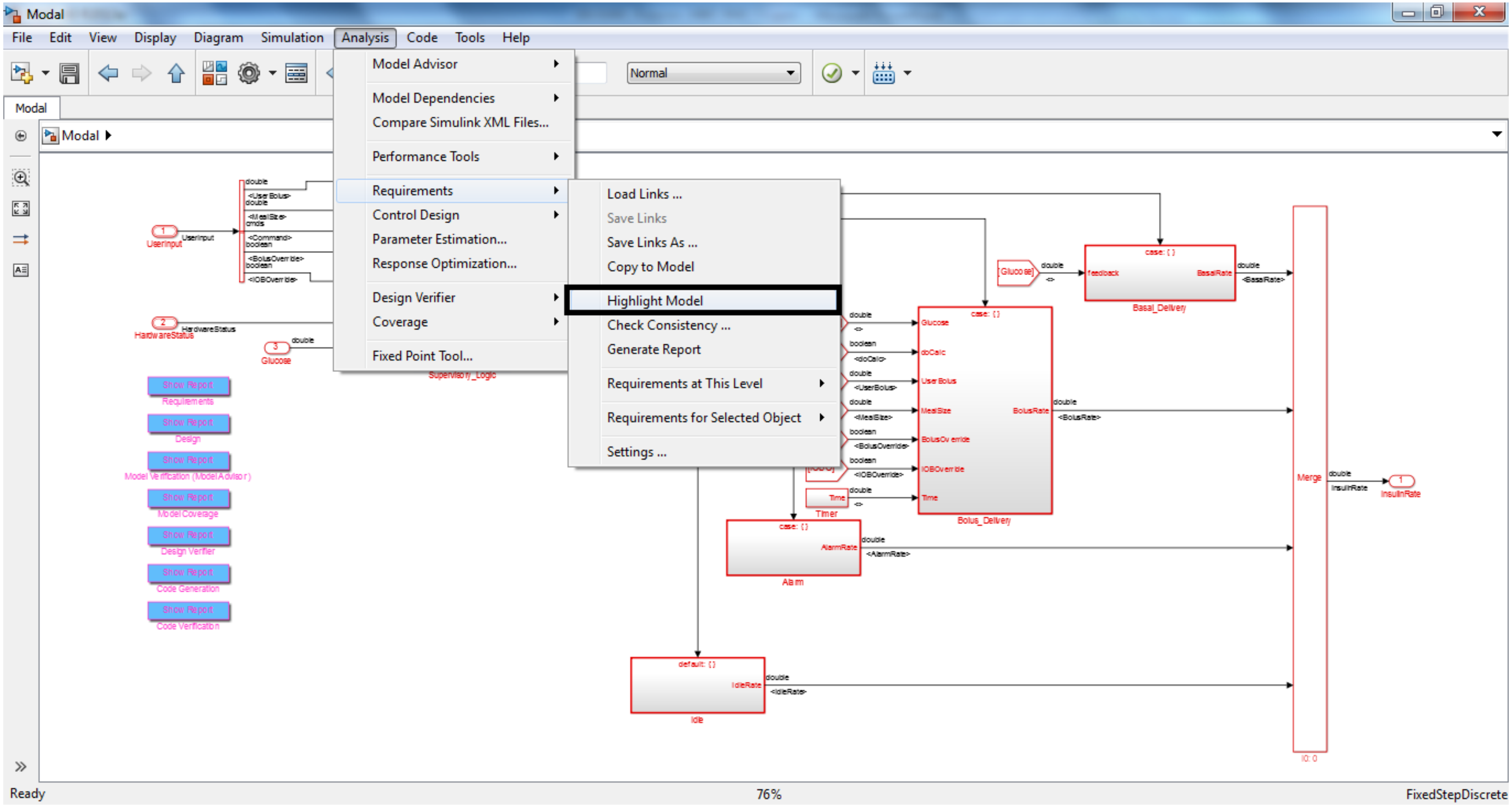
Certified Extension

See: www.polarion.com/connectors/matlab

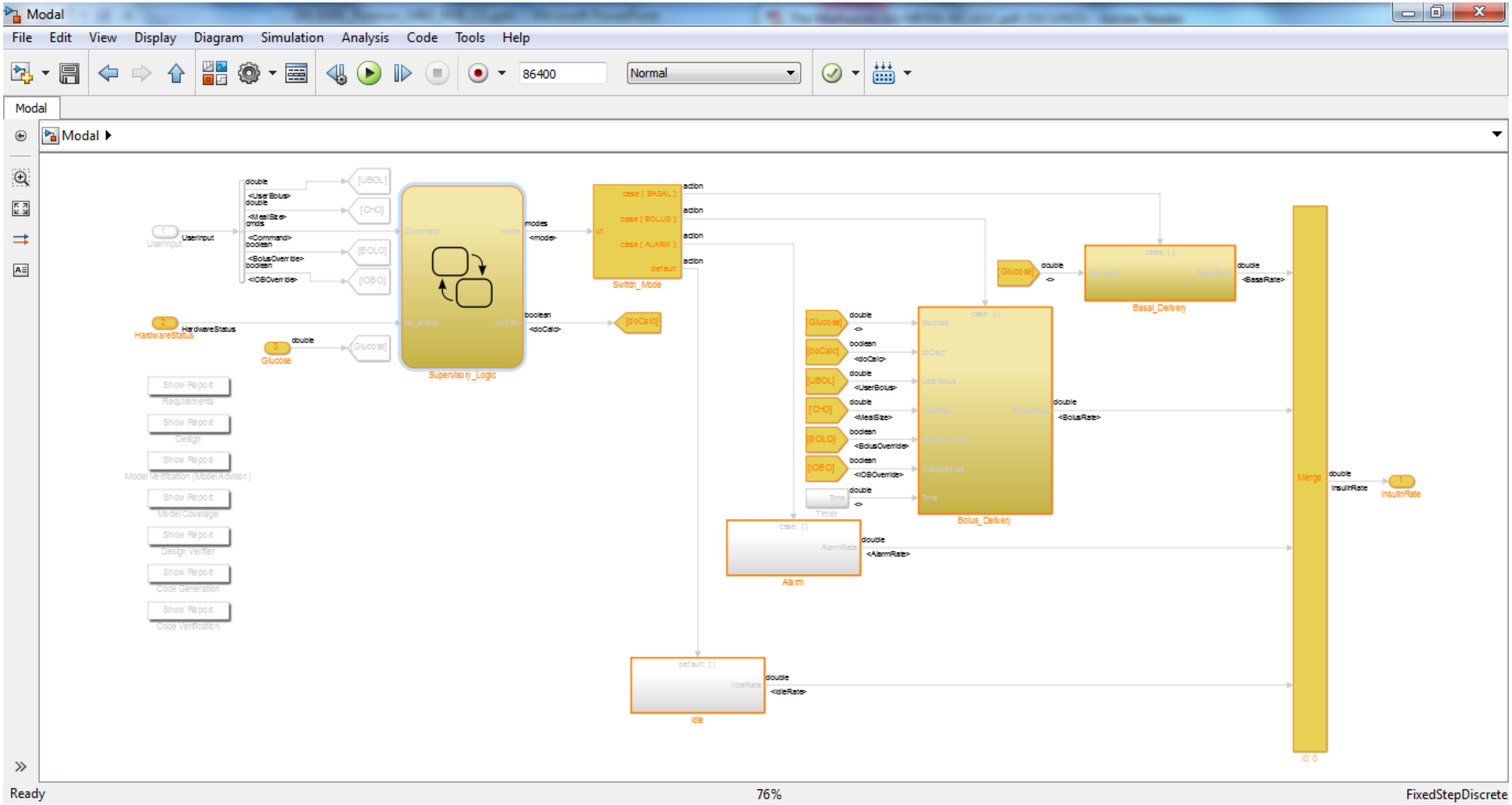
Extension Requirements

- Polarion 2013
- MATLAB/Simulink 2013a
- Simulink Validation and Verification Toolbox

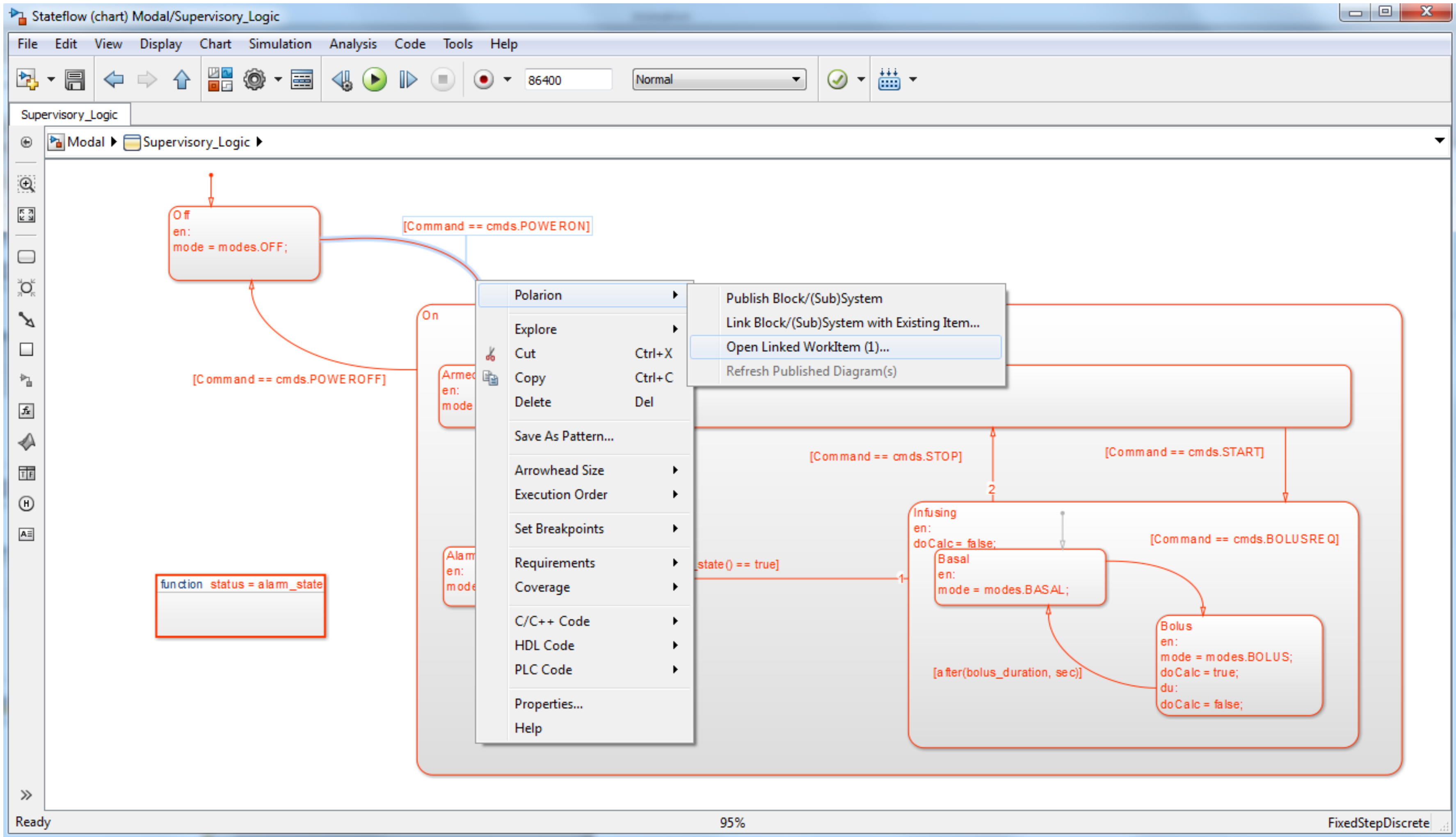
Highlight Requirements Inside



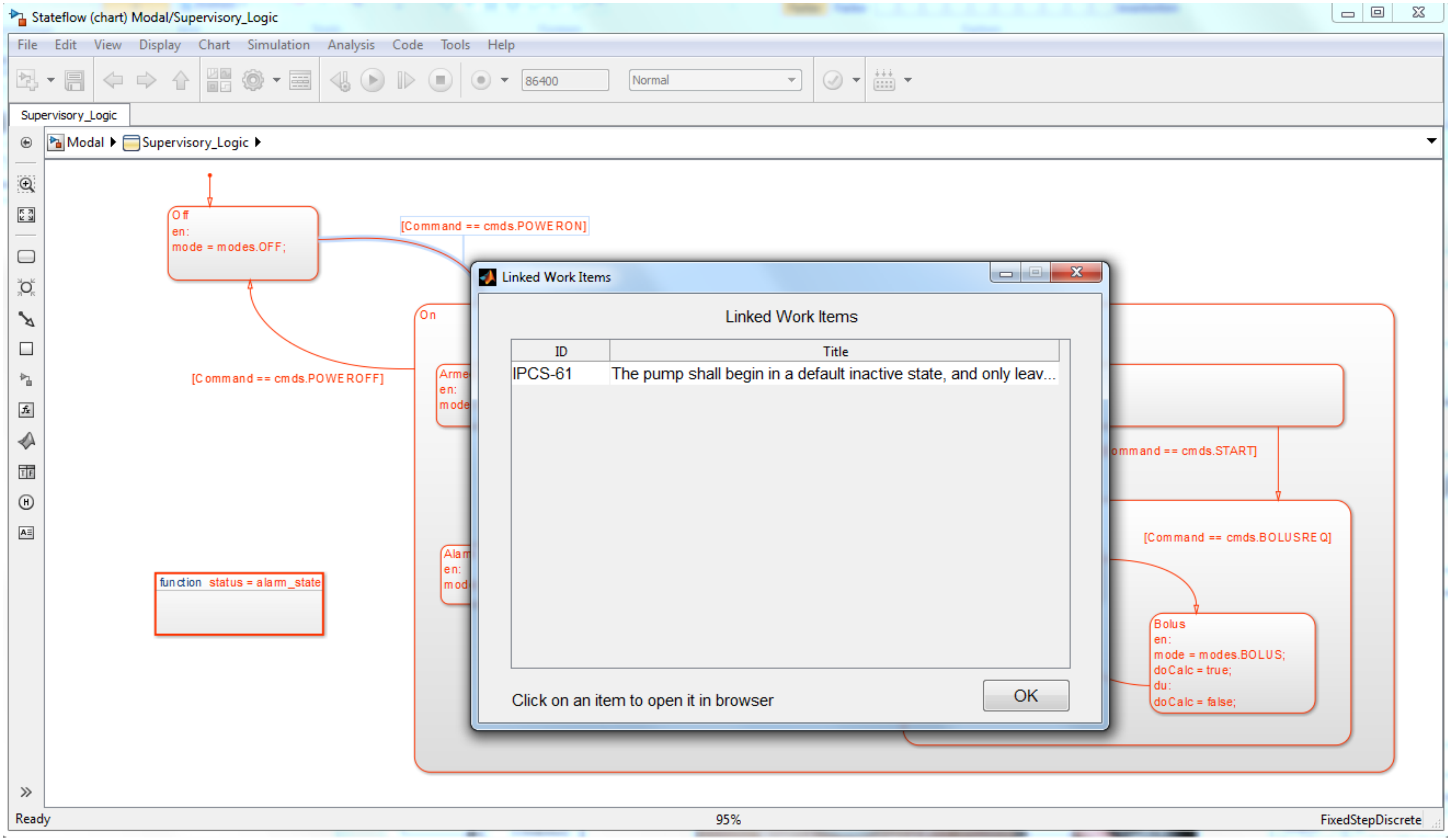
Highlight Requirements Inside



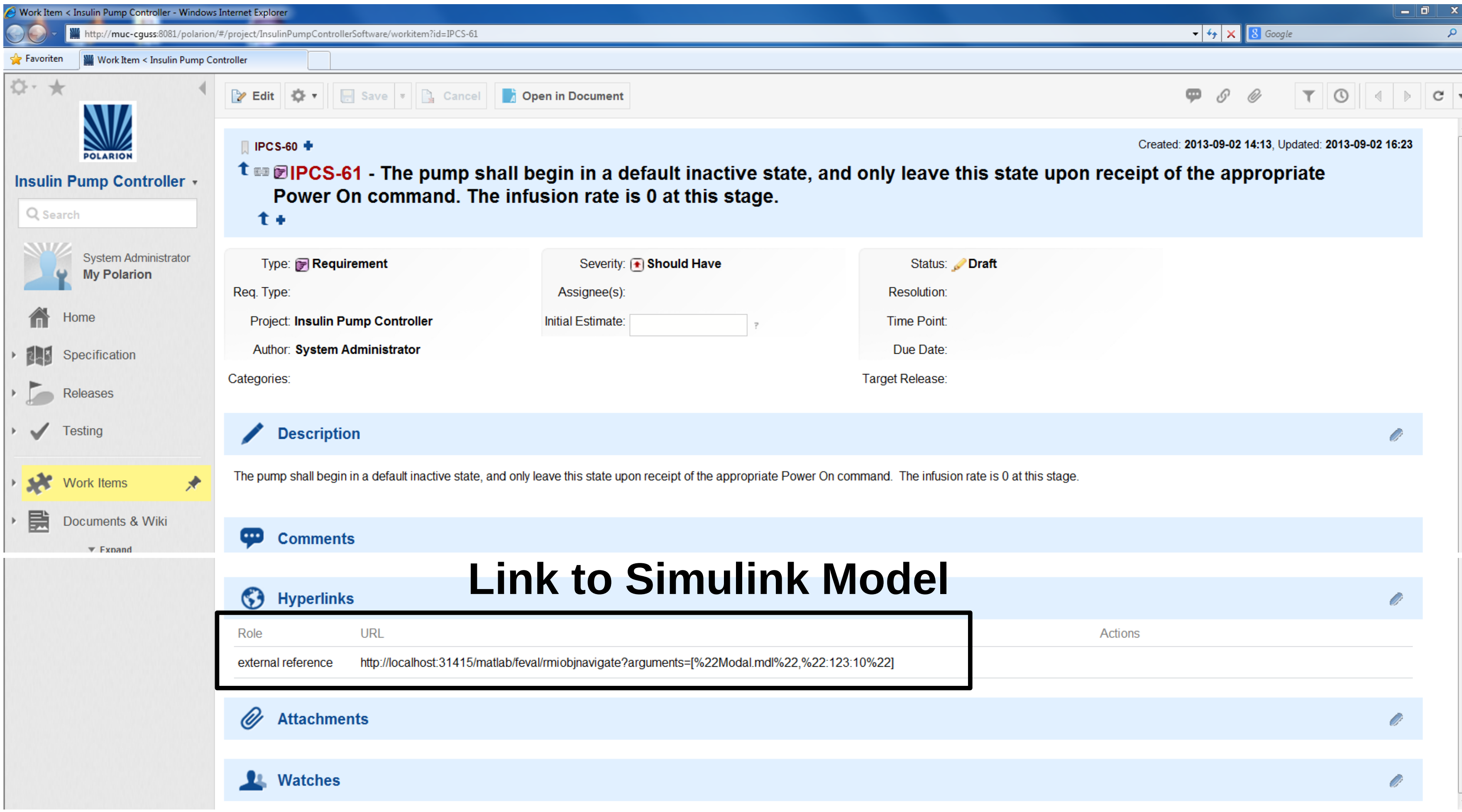
Traceability from Simulink to Polarion



Traceability from Simulink to Polarion



Traceability from Polarion to Simulink



Model-Based Design

Multi-Domain Modeling and Algorithm Development

Requirements

System De...

Environment

Physical Compo...

Algorithms

System-Level Specification

Co...

Subs... De...

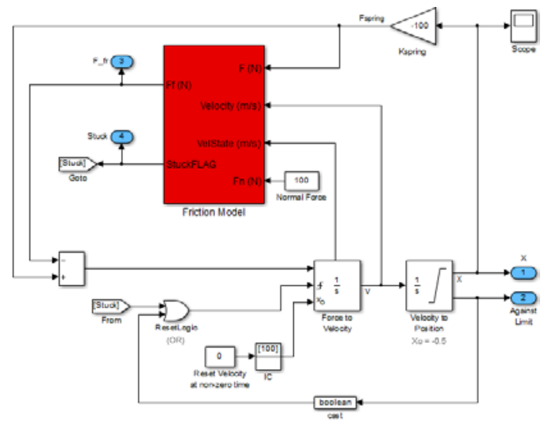
Research

Data Analysis

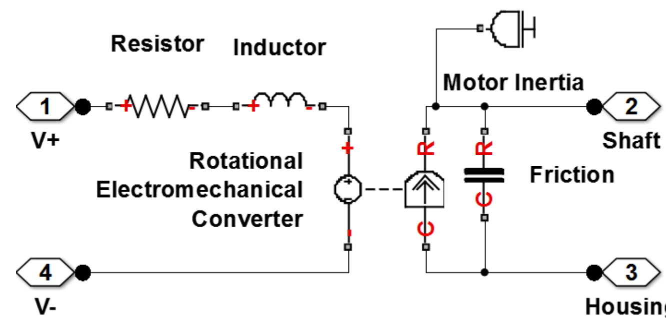
Algorithm Development

Data Modeling

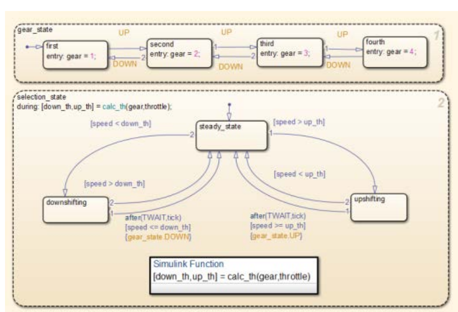
Methods for modeling systems in different domains



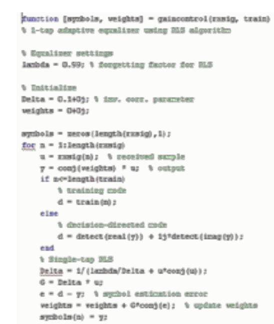
Data Flow (Block diagram)



Physical Modeling (Schematic)



Modeling of Event-Driven Systems (State - Machines)



Programing Language (Textual)

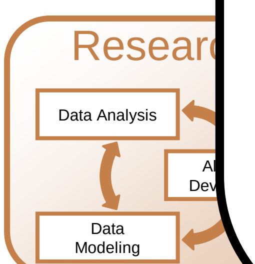
Model-Based Design

Early Concept Verification

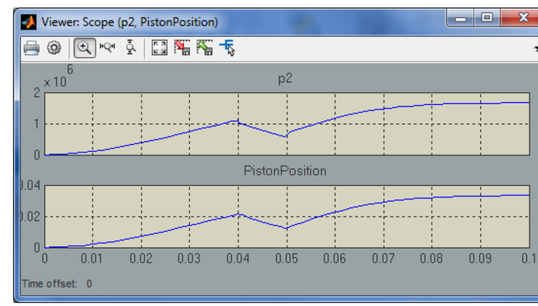
Requirements

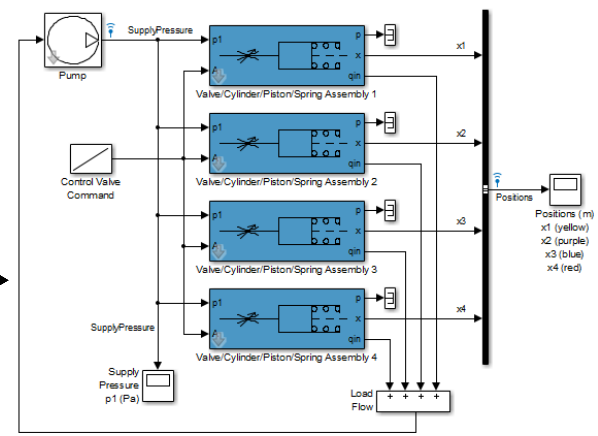
User Acceptance Testing

System-Level Specification

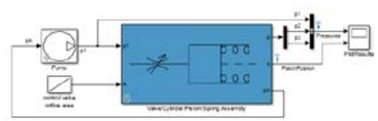


- Executable specifications
- Predict dynamic system behaviour by simulation
  - System & environment models
  - Less physical prototypes
- Use of simulation results for system design
  - Fast What-/If studies
  - Short iteration cycles





Idea



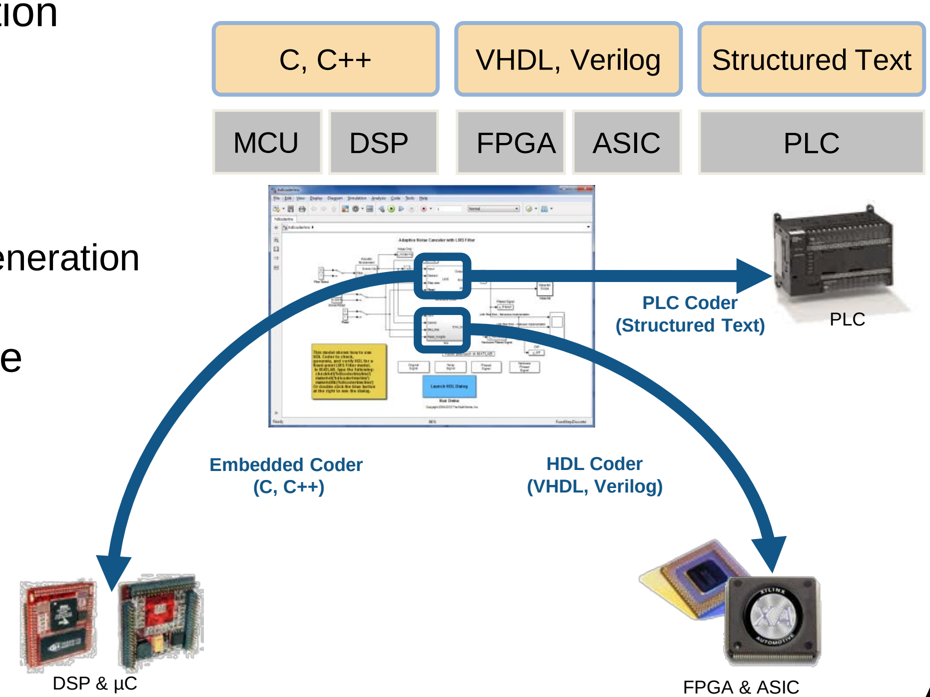
Simple Model

Detailed Model

Model-Based Design

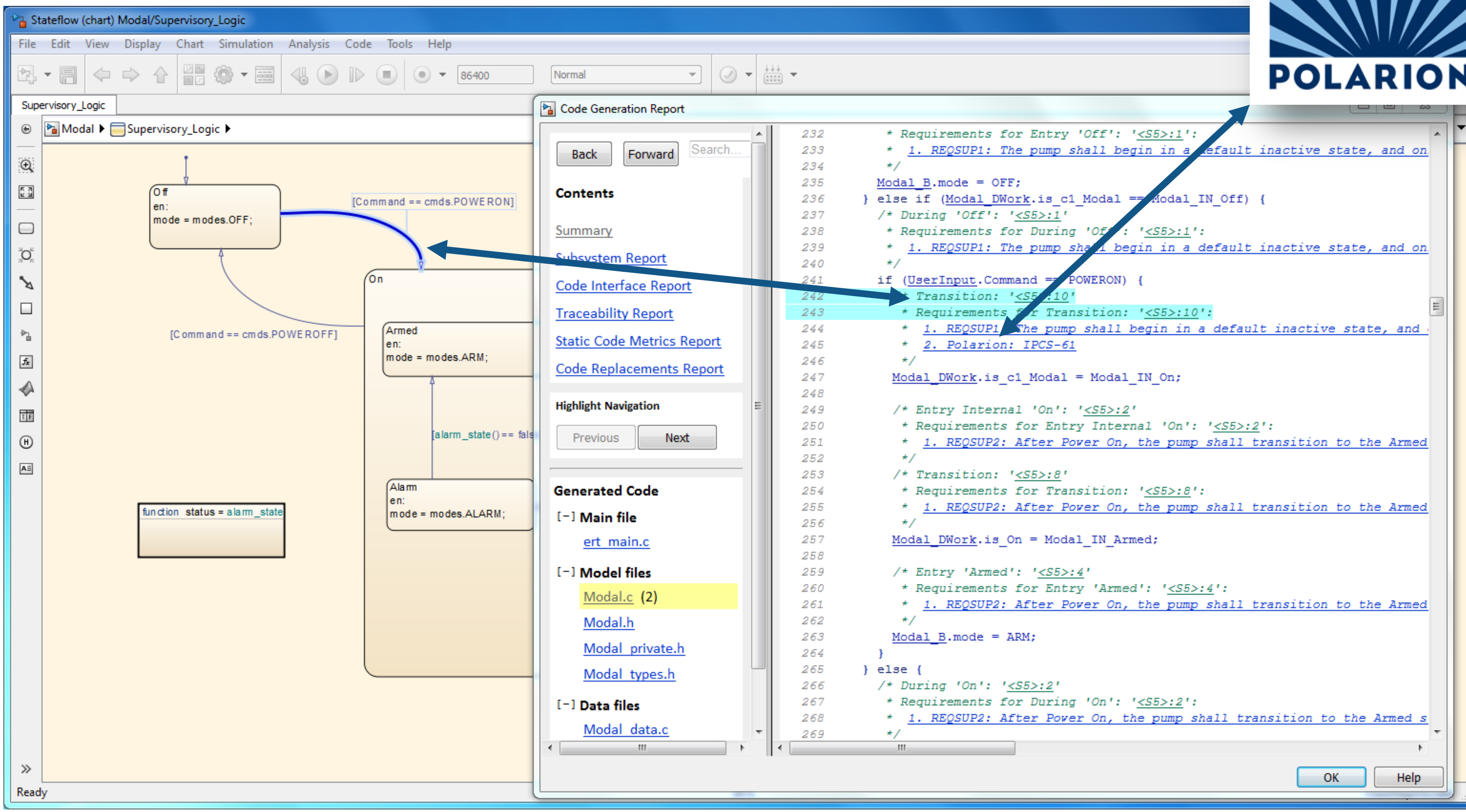
Automatic Code Generation

- 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



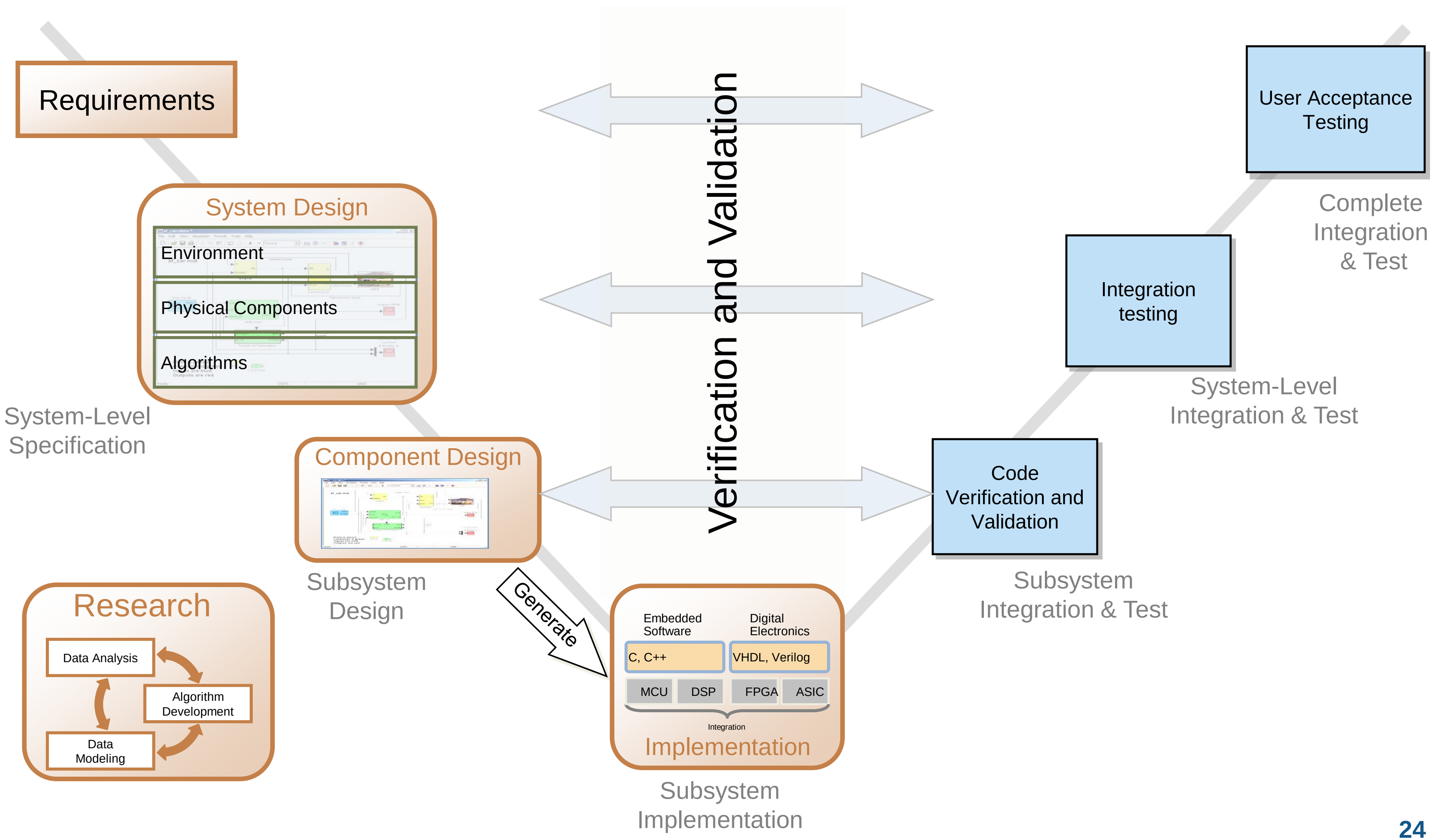
Development
Data Modeling
Integration
Implementation
Subsystem Implementation

Traceability: Requirement ←→ Model ←→Code
HTML Code Generation Report



Model-Based Design

Continuous Verification and Validation

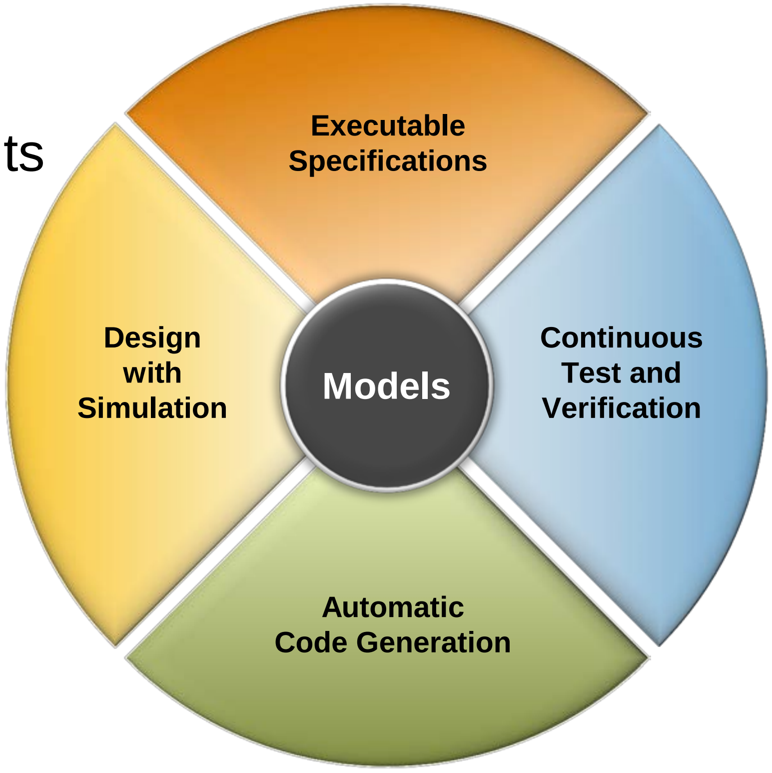


Benefits of Model-Based Design

- 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*