Automatic Validation of Simulink/Stateflow Models
Formal Verification of Safety-Critical Requirements

Dr. Udo Brockmeyer
OSC - Embedded Systems AG

The Challenge…
Innovation dominated by electronic components and SW
- exponential increase of functionality
- distribution of functions
- new technologies, e.g. X-by-wire, demand new complex technologies

Increased level of failures

The Challenge…
*to manage complexity within tough cost and time constraints!*

IAC 2004
Stuttgart, June 2004
Model-based Development Process

- Model-based development processes to cope with complexity
  - executable specifications
- Typical use cases
  - virtual system integration
  - concept validation
  - prototyping
  - auto code generation
- Matlab/Simulink/Stateflow wide-spread design tool for modeling, simulation, and code generation
  - improvements in quality while saving time
- Testing gap still exists!
  - classical testing approaches fail even in model-based processes
  - costs for testing and also failure rates still raise

The Testing Gap

Testing processes are based on skills of test engineers
- functional complexity grows exponentially
- design space for test generation grows exponentially

Test engineers need tool-support for V&V activities

Design of good and rich test suites is critical!
The Enhanced Model-based Development Process

- Requirement Capturing
- Specification
- Design
- Implementation
- Unit Testing
- Integration Testing
- System Test

Automatic Validation
- EmbeddedValidator
  to formally verify SL/SF models

Automatic Simulation
- Model violates requirement
- Model fulfills requirement

Functional Requirements

Simulink/Stateflow Model

Requirements
EmbeddedValidator – Integration in Matlab/Simulink

- drag-and-drop the EmbeddedValidator block from the Simulink library
- drag-and-drop selects subsystem to be verified
- launches EmbeddedValidator

EmbeddedValidator – Feature Support

- full support of Stateflow
- set of Simulink blocks for open-loop control
- EmbeddedValidator operates on C-Code generated from TargetLink (dSPACE)
EmbeddedValidator – Certification Mode

- Analysis and Proof Dictionaries
- Result management
- Report generation
- Powerful Pattern Library
- Automatic Matlab simulation to debug failures

Powerful Pattern Library

- large set of pre-defined patterns for requirement specification
- intuitive pattern usage based on forms
- used to specify functional and safety-critical requirements
**EmbeddedValidator – Static Failure Analysis**

- automatic waveform generation
- displaying input / output scenarios
- easy *static* failure analysis

**EmbeddedValidator – Dynamic Failure Analysis**

- fully automatic generation of test benches
- automatic generation of test scripts
- automatic simulation into failure situation
- easy *dynamic* failure analysis
EmbeddedValidator – Debugging Mode

- set of standard analysis for SL/SF models provided
- variable range violation, reachability of states, reachability of configurations

Summary

- Automatic Validation with formal verification is another key method to improve model-based processes
- EmbeddedValidator is smoothly integrated into Matlab
- EmbeddedValidator helps to find design failures early
- saves time, costs and increases quality
- EmbeddedValidator supports requirement certification and model debugging
- Automatic Validation with EmbeddedValidator can be complemented by model-based Automatic Test Generation and model-based Safety Analysis
Thank You for Your Attention!