Course Title:
Verification and Validation of Simulink Models

Course Purpose:
This one-day course describes techniques for testing and formally verifying Simulink® model behavior.

Topics include:
- Recalling the role of verification and validation in Model-Based Design
- Configuring Simulink models for testing
- Testing a Simulink model for accuracy and coverage
- Publishing test results
- Formally verifying model behavior

Pre-requisites:
MATLAB Fundamentals and Simulink for System and Algorithm Modeling
This course is intended for intermediate or advanced Simulink users. Familiarity with creating MATLAB scripts and functions is recommended.

Teaching method:
The course combines lectures, demonstrations and practical exercises in MATLAB, using original training books from MathWorks. The course is in Hebrew, but the training materials are in English.

Training Center Systematics - Contact information:
Phone number: 03-7660111 Ext: 5  Email: training@systematics.co.il
Website: http://www.systematics.co.il/mathworks
Course Objective:

Verification and Validation in Model-Based Design

Objective: Introduce verification and validation in the Simulink environment, and how it fits into a typical Model-Based Design project workflow.

- Continuous test and verification
- Types of verification
- Project environment
- System requirements
- Test plans

Analyzing Test Results

Objective: Analyze test results of a Simulink simulation, both during and after the simulation.

- Run-time analysis
- Logging data
- Saving data
- Automated analysis
- Simulation Data Inspector

Running Multiple Tests

Objective: Create repeatable tests, run groups of tests automatically, and collect coverage data on a model.

- Test process
- Self-contained tests
- Test suites
- Speeding up tests

Reporting Test Results

Objective: Discuss the methods of sharing test results for Simulink models.

- Publishing MATLAB files
- Standard reports
- Custom reports
Automatically Verifying Models

**Objective:** Generate model coverage from test cases, and use formal methods to automatically generate tests and prove properties.

- Model coverage
- Automatic test generation
- User-defined objectives
- Property proving