KnowledgeBytes

for the best ideas in information technology

www.KnowledgeBytes.net

Model-Based Development and Automated Testing Course

Course Overview

This course uses lecture and exercises to discuss recommended practices for requirement modeling and the associated concepts & implementation of a testing environment that supports formalizing textual requirements that are often captured in a requirement management tools such as DOORS or textual documents. The course discusses approaches for formalizing requirements in the form of models that can link to textual requirements stored in DOORS. The course covers concepts and the purpose of automated model analysis, which can identify requirement defects in complex systems that are difficult to identify in textual requirements through manual inspections and reviews. Model requirements are automatically translated to support automatic model-based test generation, and test execution based on a suite of tool referred to as the Test Automation Framework (TAF).

The integrated environment generically referred to as the Test Automation Framework (TAF) integrates commercially available model development and test generation tools. TAF integrates the DOORS® requirement management tool with the T-VEC Tabular Modeler (TTM) for requirement modeling. DOORS integrates also with Simulink/Stateflow®, which supports design-based models. TAF integrates requirement models with design models to provide full traceability from the requirements source to the generated tests, as reflected in Figure 1.Finally, TAF integrates with code coverage tools such as LDRA and VectorCAST too.

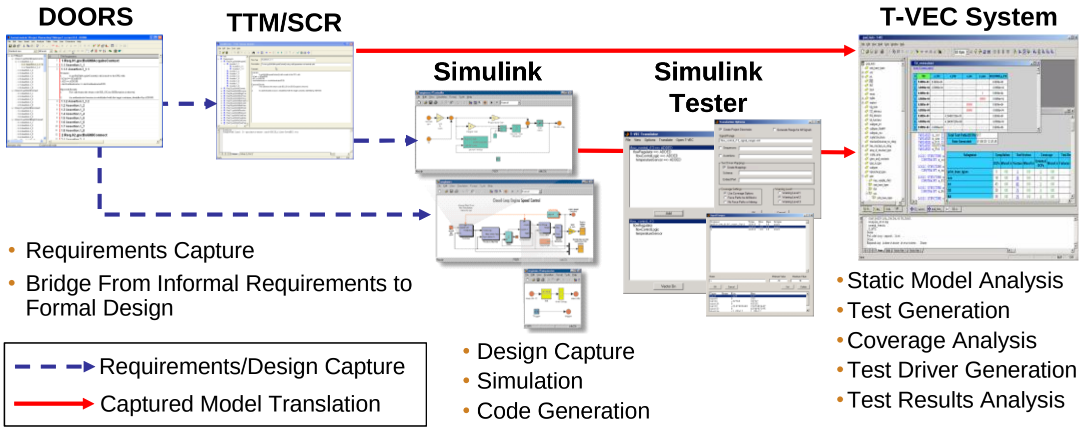

Figure 1. Modeling Tool Chain and Integrated Environment

Learning Objectives: Upon completion of this course, attendees will be able to:

- Understand basic and advanced modeling techniques
- Transforming textual requirements into requirement models
- Build models with the modeling tool
- Link textual requirements to requirement models
- Understand test selection principles that are most effective at exposing faults
- Refine high-level requirements into implementation-derived requirements and associated models
- Understand the concept of requirement defect and their relationship to software defects
- Understand requirement-to-test traceability
- Describe how to organize models and test infrastructure for team development
- Identify the advantages offered by full lifecycle testing using automated model-based testing techniques
- Describe basic requirement modeling used for requirement defect analysis and automated testing
- Automatically generate test vectors to outline a test plan and generate test reports
- Develop and use test drivers to automatically perform test execution and results analysis
- Reduce rework by reducing requirement defects
- Describe what artifacts are captured that should be configuration controlled
- Understand model-based metrics and measures
- How to get started with a pilot project

Highlights (3 –days):

- Introduction
- Modeling Approaches and Demonstration
- Fundamentals of Requirement-based Modeling as implemented by TTM
- Model Analysis/Assertions/Test Constraints and Exercise
- Model Management and Reuse Concepts
- Mode (State) Machine Concepts and Exercise
- Modeling Requirements and Exercise
- Model Organization for a Team, and Exercise
- Generating Test Reports and Exercise
- Generating Test Drivers and Exercise
- Modeling with Structures, Functions, and Message Passing Exercises
- Advanced Modeling and Exercises
- Modeling Concepts of Time and Exercise
- Full-lifecycle Exercise from Requirements through Test Execution

Resources

If possible, course attendees should bring a laptop with the following minimum configuration: Windows 2000, Windows XP, or Windows 7 with at least 256MB RAM and a 600MHz processor.

Who Should Attend

The intended audience for this course includes requirement engineers, developers, testing engineers, quality assurance personnel and others interested in learning about applying requirement and design modeling, and automated model-based testing technology.

Extensible to Simulink/Stateflow (1 extra day)

The course can be tailored or include additional training for students knowledge in Matlab/ Simulink/Stateflow tools. The focus of this course segment is on how to use the various translation and analysis features of the Simulink Tester and T-VEC Test Vector Generation System. The instructor will provide lectures and demonstrations that include the following topics:

- Overview of Simulink Tester And T-VEC Tools
  - Basic Tool Introductions
  - General Process Flows
  - First Demonstration
- Simulink Tester Front End Processes
  - Model Translation
  - Signal Range Specifications
  - Model/Test Coverage
  - Error Analysis
  - Test Sequences For Dynamic Properties
- Simulink Tester Back End Processes
  - Test Driver Generation
  - Test Execution and Results Comparison
- Simulink Tester integration with T-VEC to support model analysis, test generation, test execution and results analysis

Contact Us about Training:

Maureen Murtha
E-mail: maureen@knowledgebytes.net
Phone: 703.999.2734