

SVM is a tool developed with funding from the Defense Advanced Research Projects Agency (DARPA). SVM can manage Requirements, Models and Tests and relationships among them. SVM facilitates model-based verification of embedded control system software and provides guidance to engineers for performing Verification and Validation (V&V). SVM provides an extensible environment designed to support the use of multiple modeling, simulation, and verification tools without imposing constraints on the Models or methods used for design and implementation. SVM can support different system design processes using multiple views. Engineers may create Requirements and Tests and then create new Models which can be verified against the Requirements or they may use legacy Models (which may undergo change) and the properties (Requirements) of the model can be checked using Tests. Emmeskay, Inc. can provide consulting services to support the customization of SVM for V&V.

### Core Features of SVM

#### Requirements Management

- Create and organize Requirements and Tests
- Traceability between Requirements, Models and Tests
- Filtered views

#### Verification Methods and Activities

- Define, parameterize, and instantiate Verification Methods
- Perform simulations, formal verifications, test case generation, compilation, and regression testing
- Clone Tests over parameter ranges or for several data sets
- Reuse Tests and verification criteria across development cycle

#### Results Management

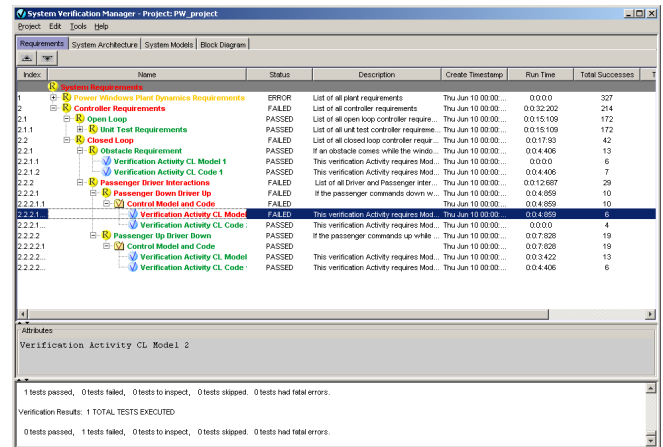
- Automatically updates the status of verifications after Completion
- Provides methods to log and access text, plots, files, etc. generated by verification activities
- Guides the user to pass results of verification methods to processing algorithms

#### Change Management

- Detect changes in Test inputs, Models, Requirements or Verification Methods and reset verification status

#### External Interfaces

- Import/export Requirements with MS Excel
- Generate reports of verifications in MS Excel



ID	Name	Status	Description	Create Timestamp	Run Time	Total Successes
1	Power Windows Plant Dynamics Requirements	ERROR	List of all plant requirements	Thu Jun 10 00:00:00	0:00:00	327
2	Controller Requirements	FAILED	List of all controller requirements	Thu Jun 10 00:00:00	0:03:20	214
2.1	Open Loop	PASSED	List of all open loop controller require.	Thu Jun 10 00:00:00	0:01:10	172
2.1.1	Unit Test Requirements	PASSED	List of all unit test controller require.	Thu Jun 10 00:00:00	0:01:10	172
2.2	Closed Loop	FAILED	List of all closed loop controller require.	Thu Jun 10 00:00:00	0:01:50	42
2.2.1	Obstacle Requirement	PASSED	If an obstacle comes within the window.	Thu Jun 10 00:00:00	0:04:46	13
2.2.1.1	Verification Activity CL Model 1	PASSED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:00:00	6
2.2.1.2	Verification Activity CL Code 1	PASSED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:04:46	7
2.2.2	Passenger/Driver Interactions	FAILED	List of all Driver and Passenger inter.	Thu Jun 10 00:00:00	0:05:07	29
2.2.2.1	Passenger Down Driver Up	FAILED	If the passenger commands down w.	Thu Jun 10 00:00:00	0:04:09	10
2.2.2.1.1	Control Model and Code	FAILED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:04:09	10
2.2.2.1.2	Verification Activity CL Model	FAILED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:00:00	4
2.2.2.1.2.1	Verification Activity CL Code	PASSED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:07:50	19
2.2.2.1.2.2	Passenger Up Driver Down	PASSED	If the passenger commands up while	Thu Jun 10 00:00:00	0:07:50	19
2.2.2.2	Control Model and Code	PASSED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:03:42	13
2.2.2.2.1	Verification Activity CL Model	PASSED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:03:42	13
2.2.2.2.2	Verification Activity CL Code	PASSED	This verification Activity requires Mod.	Thu Jun 10 00:00:00	0:04:46	6

SVM: Requirements driven verification

### SVM Use Scenarios in Product Development

- Structural verification of work products from a distributed team
- Functional verification of work products from a distributed team
- Regression Testing (Minimum Acceptance Testing) before release
- Batch Mode Testing
- Testing on multiple platforms (desktop and real-time)
- Management of System Models, Requirements and Test information

### References

- "Development of an Automated Verification and Validation Platform Using Hardware-In-the-Loop Simulation for a Solid Oxide Fuel Cell Control System"  
J. Absmeier, T. Das, S. Gopalswamy, R. Paike, *4th International ASME Conference on Fuel Cell Science, Engineering and Technology*, June 2006, Irvine, CA. (Coauthored with Delphi)
- "Managing Verification Activities Using SVM"  
B. Aldrich, A. Fehnker, P. H. Feiler, Z. Han, B. H. Krogh, E. Lim and S. Sivashankar, *Sixth International Conference on Formal Engineering Methods (ICFEM)*, Seattle, November 8-12, 2004. (Coauthored with Carnegie Mellon University, SEI and The MathWorks)

