BTC Embedded Validator

Formal Specification and Formal Verification of TargetLink® Models

Safe Function Development
- Formal Requirement Specification
- Formal Model Verification
- Generation of Counter Examples
- Automatic Debugging Support
- Model Robustness Checking
- Detection of Dead Code
- ISO 26262 and IEC 61508 Support
- Smooth TargetLink® Integration
- Automatic Comprehensive Reporting

Overview
EmbeddedValidator provides a tool suite for formal, automatic, and model-based verification. It performs model checking for reactive embedded systems designed using Simulink, Stateflow, together with TargetLink. In contrast to conventional testing approaches, model checking technology is fully automated and complete in a mathematical sense, meaning that it can detect every logical design flaw and error in the model being verified.

The Challenge
Open-loop control systems usually have deep dependencies that designers cannot completely analyze with traditional methods. Present embedded control units have such enormous functionality such that designers and testers cannot ensure correct functioning under all environmental conditions by means of conventional testing methods. Hence, complexity has increased enormously, as has time-to-market pressure, and as a consequence the error rate is increasing.

The Technology
Model checking analyzes a model of a system with regard to arbitrary input scenarios and sequences and can thus be viewed as a “complete test” that is performed against a formally specified requirement. Such requirements may be simple Boolean expressions such as “an error state is never reached”, or more sophisticated ones expressing temporal and causal properties like “an output is set only after certain input values are observed”.

By applying EmbeddedValidator, designs are produced that meet their requirements regarding robustness and correctness, and decrease design faults and costs.

Additionally, obligations from the upcoming Standard ISO 26262 such as Formal Specification and Formal Verification are addressed by EmbeddedValidator.
Early Model and Code Verification

The combination of EmbeddedValidator and TargetLink enables early testing and verification. Applying EmbeddedValidator’s model checking method in early development phases not only verifies models, it is also a step towards production code verification. TargetLink builds the bridge between verified models and verified AutoCode. It is the only code generator for Simulink that has official formal verification support with EmbeddedValidator.

User-friendly Pattern Library

EmbeddedValidator provides a library of predefined patterns for specifying functional (safety- and mission critical) requirements. Patterns can be instantiated simply by filling the pattern parameters with Boolean expressions ranging over model elements.

Standard Analyses Checks

Range Violation Analyses

EmbeddedValidator dynamically checks for range violations, i.e., whether there is model behavior in which an overflow or underflow for a certain variable occurs.

Drive-to-Analyses

Using the Drive-to-State analysis, one can select one, some or all the states of a Stateflow chart and perform a reachability check. Input-stimuli traces are generated automatically for all reachable states. Drive-to-Configuration analysis extends the Drive-to-State analysis. EmbeddedValidator analyses whether two or more states in parallel sub-charts are reachable at the same point in time.

Drive-to-Property is a more generic check that constructively tries to reach a state of the system where a certain property holds. All types of observables (variables and states) of a selected Stateflow chart or a subsystem are displayed for the user to specify a property. The data objects are used to express an arbitrary Boolean property. Using the Drive-to-capabilities, the user can automatically obtain test vectors by driving the system into an arbitrary configuration. Also any alternative trace to reach a certain system configuration can be generated fully automatically, allowing to create any user-driven test scenario. All generated traces can be exported into various formats to be re-used in MiL, SiL, PiL and HiL test environments like BTC EmbeddedTester®.

Automatic Reporting

EmbeddedValidator provides detailed application reports. In addition to general information about the user, date, and design, each report also contains all the information on the profile, for example, defined analyses and defined proofs. Reports can be generated in several output formats like XML, HTML and TXT.