# HiLiTe Functionalities

# CyPhy-HiLiTe Verifier Capabilities

The Verifier takes a model written in CyPhyML, additional parameters describing the location of HiLiTe and its working directory in the file system as well as the input file names for the tool. Based on this information, the Verifier invokes HiLiTe, and the result of the execution is written back into the CyPhy model.

The Verifier uses the static analyzer capabilities of HiLiTe. The tool takes the ranges of the input variables of a Simulink model. In CyPhy, the Simulink model is represented as an URI to the Simulink model file, whereas the ranges are described by range nodes that are attached to wrappers pointing to Simulink input ports. CyPhy is able to describe the following parameters: Signal, DataType, Maximum, Minimum, Hard Maximum, Hard Minimum, Units, and Resolution. The Verifier is capable of translating these parameters into the input format of HiLiTe.

The output of HiLiTe is an XML describing the result of the execution along with reports for additional information. The Verifier can display these errors, and provides immediate traceability to the place of the error in the Simulink model. The Verifier can load and show the reports as well. The number of errors is fed back to CyPhy: the FormalRequirements model component used to invoke the Verifier will contain this high-level result of the execution.

# Limitations of CyPhy-HiLiTe Verifier

The Verifier by no means supports all the functionalities of HiLiTe. The tool is capable of performing dynamic analysis and automatic test generation besides the static analysis functions that the Verifier uses. The Verifier works with a fixed set of table columns, HiLiTe facilitates the configuration of them.

# HiLiTe-CyPhy Verifier Files

The install contains an executable only (*setup\_ISIS-HybridSALVerifierxxx.exe*). The setup installs the following groups of files.

* \bin\HiLiTeVerifierPlugin.dll: interpreter component
* \bin\HiLiTeVerifierPlugin.tlb: type library for the interpreter
* \doc\readme.txt: guide to run the tool and the case studies
* \src\\*: source (cs source files, snk signature, csproj and sln project files)
* \samples\\*: case studies: Model123, FlightControl, Display Logic