The HybridSAL tool suite

The HybridSAL relational abstractor is a tool that constructs a relational abstraction of a hybrid system modeled in the HybridSAL language. The HybridSAL language is a formal modeling language for systems with both discrete behavior (e.g. state machines and logic) and continuous dynamics (e.g. differential equations). The HybridSAL language can also model certain formal requirements, or specifications, of a dynamical system. A typical example is safety properties that are modeled using textual syntax for Linear Temporal Logic. The HybridSAL tool suite provides tools that can prove, or refute, the specified property for the specified system. The HybridSAL relational abstractor is one key component of the HybridSAL tool suite.

The CyPhy-HiLiTE Verifier

The Verifier takes a model written in CyPhyML, and invokes HiLiTE, a static analyzer. The analysis results are propagated back into the CyPhy model. The Verifier can display model errors and provides immediate traceability to the place of the error in the Simulink model. The CyPhy-HiLiTE Verifier can be used on a set of models where outputs of one model can feed into inputs of another. User can repeatedly invoke CyPhy on successive models while range constraints on intermediate signals between models are maintained in a single consolidated range file. Thus, the range bounds/constraints determined by the computations in one model can influence the design property correctness in downstream models.

The CyPhy HybridSAL Verifier

The Verifier takes a Simulink model in ESMoL format, and converts it to a hybrid automaton in the format of HybridSAL. Additionaly a CyPhy FormalRequirement model element contains the LTL expression to be verified by HybridSAL. The Verifier executes the tool, and then the retrieved result is added to the formal requirement.

Network Fault and Performance Analysis Tool Suite

The Network Fault and Performance Analysis Tool Suite supports analyzing the design of systems and network architecture in the context of failures and performance. The tool suite allows the joint analysis of performance (latency/timing properties, and utilization/bandwidth) and failure (fault modes and fault propagation) starting from the same network model. The rationale behind this tool suite is the insight that the trade-offs in the network architecture design space can be comprehensively explored only when both dimensions are systematically explored in conjunction. Further, both fault and performance requirements are irrevocably linked such that any design change in one dimension impacts the other. The tool consists of a Network Editor to create network models, a Fault Analyzer, and a Performance Analyzers. The latter two analyzers can be invoked from the Network Editor.