**Sdl2pml:**

* **semi-automated** system abstraction
* **automated** model extraction, simulation, and formal verification of real-life complex SDL specification
* Implemented Sound **algorithms** preserve all properties of the SDL system.
* **Includes** our model of discrete time, abstraction, and support for all relevant SDL functionality and constructs such as:
* dynamic process creation,
* rational data types,
* and communication with more than one process instance.
* Formal **method**: model checking → SpinRCP
* the **first SDL to Promela** converter that **supports**:
* a full SDL system structure,
* predefined and userdefined data types,
* dynamic process creation and termination,
* timers with parameters,
* save construct,
* asterisk state,
* asterisk input,
* priority input,
* implicit transition,
* rational numbers,
* enabling condition,
* direct and indirect addressing,
* path limitations,
* procedures and
* some other less frequently used SDL concepts.
* Additionally, there is no notion of time in Promela, so they had to model it.
* a new model of discrete-time that was modeled entirely with standard Promela
* supports the inclusion of embedded C code for the ADT operators.
* supports all relevant SDL functionality and constructs such as
* dynamic process creation,
* communication with more than one pro-cess instance,
* charstring and
* rational data types,
* support for
* priority input,
* SAVE construct,
* abstraction,
* and others.
* They managed to automate some steps of the process, but there is still room for improvement.
* They are currently developing new algorithms for sdl2pml that will enable the use of the latest versions of Spin with included support for inline specification of LTL properties, multiple never claims, parallel breadth-first search algorithm, iterative and swarm verification, and other techniques.

TOOL:

* The sdl2pml implements algorithms from in C++:
* modelling of all predefined and user-defined SDL data types,
* support for the dynamic creation of processes,
* modelling of the priority signal,
* modelling of the implicit transition,
* modelling of the spontaneous transition,
* modelling of the save construct,
* modelling of the priority input,
* modelling of the enabling condition,
* modelling of the asterisk input,
* modelling of direct (PId) and indirect addressing (name of the process, name of the signal route),
* support for path limitations introduced by the via statement,
* dynamic monitoring of the associated channel,
* modelling of the asterisk state,
* support for timers with parameters,
* introduction of probes for monitoring the system's behaviour during formal verification of the model,
* new model of discrete time in Promela,
* modelling of the SDL rational data types REAL, TIME, and DURATION,
* modelling for procedures,
* semi-automatic support of ADT operators.
* In their future research:
* extend the sdl2pml using a graphic user interface,
* provide a hierarchical view of the model,
* include support for the multi-threaded execution,
* port it to other operating systems and prepare methodology that will help users to prepare a model of the system.