Formal Verification of UML Statecharts with Real-time Extensions*

Alexandre David[1], M. Oliver Möller[2], and Wang Yi[1]

[1] Department of Information Technology, Uppsala University,
adavid@docs.uu.se
[2] BRICS Basic Research in Computer Science, Aarhus University,
omoeller@brics.dk.

Extended Abstract

We present a framework for formal verification of a real-time extension of UML statecharts. This formalism is based on hierarchical state machines, that can be put in parallel at any level of composition. It features powerful event-communication, synchronization mechanisms, and actions triggered on entry or exit of components. Industrial modeling tools like Rhapsody or Rational Rose support UML statecharts and are applied in large-scale projects.

We restrict ourselves to the subset, where synchronization happens only in hand-shake fashion and actions are always associated with transitions. Then we enrich this subset with with real-time constructs, namely clocks, timed guards, and invariants. This makes the modeling language appropriate for systems, where the correct behavior is dependent on time constraints and subtle synchronization of individual components via delays. In the formal analysis, it is then e.g. not only possible to establish, that every action a is necessarily followed by an action b, it can also be proven (or refuted) that this happens within some fixed time bound.

We formally describe the such obtained modeling language as *hierarchical timed automata* (HTAs), and equip it with a rule-based formal semantics.

Properties of a model in this formalism can be expressed in real-time logics like timed computation tree logic (TCTL). To realize automated verification, we give a translation of HTAs into a network of flat timed automata with hand-shake synchronization, basic data types, and committed locations. This can serve as input to the UPPAAL model checking tool [ABB+01].

We report on an XML based implementation of this translation. The well-known pacemaker example is used to illustrate our technique: we are able to model-check safety and (unbounded) response properties. It turns out that the validity of a safety property is strongly dependent on the right choice of parameters. Here, the results of the formal verification can be used to make correcting adjustments in the model. For translation and verification we give run-time data.

An detailed exposition of this work is available in a technical report [DM01].

References

[ABB+01] Tobias Amnell, Gerd Behrmann, Johan Bengtsson, Pedro R. D'Argenio, Alexandre David, Ansgar Fehnker, Thomas Hune, Bertrand Jeannet, Kim G. Larsen, M. Oliver Möller, Paul Pettersson, Carsten Weise, and Wang Yi. UPPAAL - Now, Next, and Future. In F. Cassez, C. Jard, B. Rozoy, and M. Ryan, editors, *Modelling and Verification of Parallel Processes*, number 2067 in Lecture Notes in Computer Science Tutorial, pages 100–125. Springer–Verlag, 2001. available at http://www.brics.dk/~omoeller/papers/movep2k.[ps.gz|pdf]; see also http://www.uppaal.com.

[DM01] Alexandre David and M. Oliver Möller. From Hierarichcal Timed Automata to UPPAAL. Research Series RS-01-11, BRICS, Department of Computer Science, University of Aarhus, March 2001.

* Supported by the European AIT-WOODDES project, No IST-1999-10069.