Extended Abstract : Multi-Timed Bisimulation for Distributed Timed Automata

Ortiz Vega, James Jerson; Schobbens, Pierre-Yves

Published in:
Open problems in real-time computing

Publication date:
2014

Document Version
Early version, also known as pre-print

Link to publication

Citation for pulished version (HARVARD):
Ortiz Vega, JJ & Schobbens, P-Y 2014, Extended Abstract : Multi-Timed Bisimulation for Distributed Timed Automata. in *Open problems in real-time computing.* http://actriss.org/2evenements/passes/9OPRTC-ULB/, Brussels, Belgium, 4/04/14. <http://actriss.org/2evenements/passes/9OPRTC-ULB/>

Multi-Timed Bisimulation for Distributed Timed Automata

James Ortiz and Pierre-Yves Schobbens

Computer Science Faculty, University of Namur
jor@info.unamur.ac.be, pys@info.unamur.edu.be

1 Extended Abstract

Distributed and reactive systems can be modeled in a modular manner using operations such as sequential or parallel composition in order to build larger modules. Each component of a modular system can be replaced by behaviorally equivalent components without changing the properties of the modules. The properties can be preserved by means of semantic equivalences. Branching bisimulation preserves Computation Tree Logic (CTL)[4] and language inclusion preserves Linear Temporal Logic (LTL)[11]. The idea behind bisimulation is to regard two states as equivalent if they can engage in the same observable transitions (transitions with identical labels) and after performing similar transitions end up in equivalent states again. If one chooses to think of every transition as observable, the relation is called a strong bisimulation. If one abstracts from internal transitions, it is called weak bisimulation or branching bisimulation.

The timed properties can be preserved by means of timed semantic equivalences. Timed bisimulation preserves timed properties expressed in timed modal logics such as Timed Computation Tree Logic (TCTL) [16], L_ν [12]. Timed bisimulation was shown decidable for Timed Automata (TA) [3]. Efficient algorithms checking for timed bisimilarity have been discovered [12] and implemented in tools for automatic verification [6]. TA are used to model real-time systems and have become popular as modeling language for several model checkers [17][13] because of their simplicity and tractability. A timed automaton is a finite automaton augmented with real-valued clocks. Clocks are variables that increase at the same rate in order to register time progress. Constraints on these clocks are used to restrict the behaviors of the automaton. The model of TA assumes perfect clocks: all clocks have infinite precision and are perfectly synchronized. Transitions are labelled with constraints on clocks, called guards, that indicate when such transition may take place. Usually TA are used to model real-time systems with hard constraints. In this case, TA are equipped with an invariant, which is a constraint on clocks that limits time progress in each control state [9]. A established bridge has been provided between TA and timed modal logic, by the notion of characteristic formula. A characteristic formula for TA is a formula in a timed temporal logic L_ν that completely characterizes the behavior of an automaton modulo some chosen relation [12]. A solution has first been proposed in [12] for TA, providing formulas in the timed modal logic L_ν. Then, these results have been improved in [1], yielding linear constructions.

Some variants of TA called Distributed Timed Automata DTA [10] and icTA [2][15] are used to model real-time distributed systems, where the clocks are not necessarily synchronized. Constraints on the clocks are used to restrict the behaviors of the automaton in the different processes. The clocks belonging to one process can be read by another process, but a clock can only be reset by its owner process. The clocks in icTA can be in different processes and we cannot assume that they are perfectly synchronized. DTA and icTA can been used to model systems such as the Controller Area Network (CAN) [14], WirelessHART Networks [5], and the ARINC-659 protocol [7].

Our first contribution is to extend the definition of Timed Labelled Transition Systems (TLTS), icTA [2], timed languages and timed bisimulation removing the assumption of clock synchronization, the idea that the clocks can advance independently if they are in different processes, yielding the multi-timed bisimulation. We will show that multi-timed bisimulation is decidable (more exactly, EXPTIME-complete). Our second contribution is a logical characterization of multi-timed bisimulation for icTA, based on timed modal logic ML_ν, an extension of L_ν [12] and Hennessy-Milner logic [8].

References

1. L. Aceto, A. Ingólfsdóttir, M. L. Pedersen, and J. Poulsen. Characteristic formulae for timed automata. *ITA*, 34(6):565–584, 2000.
2. S. Akshay, B. Bollig, P. Gastin, M. Mukund, and K. N. Kumar. Distributed timed automata with independently evolving clocks. In F. van Breugel and M. Chechik, editors, *CONCUR*, volume 5201 of *LNCS*, pages 82–97. Springer, 2008.
3. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
4. M. C. Browne, E. M. Clarke, and O. Grumberg. Characterizing finite kripke structures in propositional temporal logic. *Theor. Comput. Sci.*, 59:115–131, 1988.
5. M. De Biasi, C. Snickars, K. Landernäs, and A. Isaksson. Simulation of process control with wirelesshart networks subject to clock drift. In *Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference*, COMPSAC '08, pages 1355–1360, Washington, DC, USA, 2008.
6. J. C. Godskesen, K. G. Larsen, and K. Cerans. User's manual for epsilon, 2000.
7. D. A. Gwaltney and J. M. Briscoe. Comparison of communication architectures for spacecraft modular avionics systems. Technical report, 2006.
8. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985.
9. T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Inf. Comput.*, 111(2):193–244, 1994.
10. P. Krishnan. Distributed timed automata. *Electr. Notes Theor. Comput. Sci.*, 28, 1999.
11. L. Lamport. What good is temporal logic? In *IFIP Congress*, pages 657–668, 1983.
12. F. Laroussinie, K. G. Larsen, and C. Weise. From timed automata to logic - and back. In J. Wiedermann and P. Hájek, editors, *MFCS*, volume 969 of *Lecture Notes in Computer Science*, pages 529–539. Springer, 1995.
13. The Liège Automata-based Symbolic Handler (LASH). Available at `http://www.montefiore.ulg.ac.be/~boigelot/research/lash/`.
14. A. Monot, N. Navet, and B. Bavoux. Impact of clock drifts on CAN frame response time distributions. In *16th IEEE International Conference on Emerging Technologies and Factory Automation - ETFA 2011*, Toulouse, France, Sept. 2011.
15. J. J. Ortiz, A. Legay, and P.-Y. Schobbens. Distributed event clock automata - extended abstract. In B. Bouchou-Markhoff, P. Caron, J.-M. Champarnaud, and D. Maurel, editors, *CIAA*, volume 6807 of *Lecture Notes in Computer Science*, pages 250–263. Springer, 2011.
16. S. Tripakis and S. Yovine. Analysis of timed systems using time-abstracting bisimulations. *Formal Methods in System Design*, 18(1):25–68, 2001.
17. The UPPAAL tool. Available at `http://www.uppaal.com/`.