The Real-Time Maude Tool

Peter Csaba Ölveczky[1] and José Meseguer[2]

[1] Department of Informatics, University of Oslo
peterol@ifi.uio.no
[2] Department of Computer Science, University of Illinois at Urbana-Champaign
meseguer@cs.uiuc.edu

Abstract. Real-Time Maude is a rewriting-logic-based tool supporting the formal specification and analysis of real-time systems. Our tool emphasizes expressiveness and ease of specification over algorithmic decidability of key properties, and provides a spectrum of analysis methods, including symbolic simulation, and unbounded and time-bounded reachability analysis and LTL model checking. Real-Time Maude has proved well suited to analyze both correctness and performance of large and complex real-time systems, including state-of-the-art schedulers, network protocols, and wireless sensor network algorithms.

1 Introduction

Real-Time Maude is a high-performance tool that extends the rewriting logic-based Maude system [1] to support the formal specification and analysis of real-time systems. The characteristic features of Real-Time Maude are:

- Its specification formalism emphasizes generality and expressiveness, yet is simple and intuitive.
- Real-Time Maude is particularly suitable for specifying distributed real-time systems in an object-oriented style, and provides advanced object-oriented features such as inheritance and dynamic object creation and deletion.
- It does not build in a fixed communication model; instead, the user has the flexibility to easily define the appropriate communication model.
- It supports a range of analysis methods, including symbolic simulation and unbounded and time-bounded reachability analysis and LTL model checking.

Real-Time Maude is particularly useful for specifying and analyzing advanced distributed object-based systems with novel forms of communications and/or complex and unbounded data types. Such systems are typically beyond the pale of timed-automaton-based tools, as well as formal tools and simulation tools that are based on a fixed model of communication. One example of an application domain with new forms of communications for which Real-Time Maude has proved useful is the rapidly emerging field of *wireless sensor networks*.

Real-Time Maude is implemented in Maude [1] as an extension of Full Maude. Since most commands are executed by translating them into Maude commands [2], Real-Time Maude's performance is in essence the good one of Maude.

C.R. Ramakrishnan and J. Rehof (Eds.): TACAS 2008, LNCS 4963, pp. 332–336, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The tool is available at `http://www.ifi.uio.no/RealTimeMaude`. The paper [2] describes in detail the semantic foundations of our tool. The enhancements in the tool and its applications since our previous tool paper [3] include: (i) development of conditions that guarantee soundness and completeness of Real-Time Maude analysis for many applications (Section 3); (ii) important new applications (Section 4); and (iii) improved support for object-oriented features such as subclasses, attributes, etc., in search and model checking commands.

2 Specification and Analysis in Real-Time Maude

A Real-Time Maude specification is a tuple (Σ, E, IR, TR), where

- (Σ, E) is a theory in *membership equational logic* [4], with Σ a signature and E a terminating and confluent set of conditional equations and membership axioms. (Σ, E) specifies the system's state space as an algebraic data type.
- IR is a set of *conditional instantaneous rewrite rules* specifying the system's *instantaneous* (i.e., zero-time) local transition patterns.
- TR is a set of *tick rewrite rules* which model the time elapse in the system and have the form

  $\texttt{\{}t\texttt{\}}$ `=>` $\texttt{\{}t'\texttt{\}}$ `in time` τ `if` *cond*

  where τ is a term denoting the *duration* of the rule, t and t' are terms, and `{_}` is an operator encapsulating the global state, so that the form of the tick rules ensures that time advances uniformly in all parts of the system. Intuitively, the tick rule says that it takes time $\sigma(\tau)$ to go from state $\texttt{\{}\sigma(t)\texttt{\}}$ to state $\texttt{\{}\sigma(t')\texttt{\}}$ for any substitution σ of the variables in t, t', and τ that satisfies the condition *cond*.

Real-Time Maude is particularly well suited to specify real-time systems in an object-oriented way. The state of a system is then represented by a term that has the structure of a *multiset* of objects and messages (with *delays*).

To cover the entire time domain (which can be either discrete or dense), tick rules typically have the form $\texttt{\{}t\texttt{\}}$ `=>` $\texttt{\{}t'\texttt{\}}$ `in time` X `if` $X \leq u \wedge$ *cond*, for X a variable not occurring in t. To execute such rules, Real-Time Maude offers a choice of heuristic-based *time sampling strategies*, so that only *some* moments in time are visited. The choice of such strategies includes:

- Advancing time by a fixed amount Δ in each application of a tick rule.
- The *maximal* strategy, that advances time to the next moment when some action must be taken. That is, time is advanced by u time units in the above tick rule. This corresponds to *event-driven simulation*.

Real-Time Maude offers a spectrum of analysis commands, including:

- *Timed rewriting* that simulates *one* behavior of the system, possibly up to a time limit, from an initial state.

- Explicit-state breadth-first *search* for *reachability analysis*. This command analyzes all possible behaviors of the system, relative to the selected time sampling strategy, to check whether a state matching a *pattern* and satisfying a *condition* can be reached from the initial state. Paths leading to the (un)desired state can be exhibited. Search may be limited to search only for states reachable from the initial state in a desired time interval.
- Explicit-state *linear temporal logic (LTL) model checking*. Although our tool does not support model checking of *metric* LTL properties, it offers model checking of "clocked" properties that involve both the states and the *durations* in which the states can be reached. Model checking may be unbounded, or consider only behaviors up to a given duration. Since, relative to a time sampling strategy, the number of states reachable from an initial state in a finite time interval should be finite, time-bounded LTL model checking is possible also when an infinite number of states can be reached *without* a time bound. Our tool gives a counterexample if a formula does not hold.
- Finding the *shortest* and the *longest* time it takes to reach a state pattern.

3 Soundness and Completeness of the Analysis

The behaviors obtained by applying the tick rules according to a time sampling strategy is a subset of all possible behaviors in a system. Therefore, Real-Time Maude search and model checking analyses are *sound* in the sense that any counterexample to the validity of an invariant or LTL property found by such analysis is a valid counterexample in the system. However, Real-Time Maude analyses are in general *incomplete* for dense time, since there is no guarantee that the selected time sampling strategy covers all interesting behaviors.

In [5] we investigate under what circumstances *maximal time sampling* analyses are *sound and complete*. For object-oriented systems, we give a set of simple and easily checkable conditions for completeness of such analyses. These conditions are satisfied by useful classes of systems encountered in practice, including the large and complex AER/NCA and OGDC case studies mentioned below, that fall outside the class of dense-time systems for which well known decision procedures exist. For such systems, *time-bounded* search and model checking relative to the maximal time sampling strategy become sound and complete decision procedures for the satisfaction of, respectively, invariants and LTL properties *not including the next operator* $\bigcirc$. For discrete time, our results justify using maximal time sampling instead of visiting each time instant, which can drastically reduce the state space to make search and model checking analyses feasible.

4 Some Real-Time Maude Applications

The following is a sample of advanced Real-Time Maude applications in which the tool has been useful both as a simulation tool and as a model checking tool. Other Real-Time Maude applications include: time-sensitive cryptographic

protocols [6], parts of a multicast protocol developed by IETF [7], other state-of-the-art wireless sensor network algorithms [8], and power management algorithms for multimedia systems [9].

The AER/NCA Active Network Protocol Suite [10]. AER/NCA is a suite of active network protocols that aim to make network multicast scalable, fault-tolerant, and congestion-avoidant. The definition of the protocol mandated that aspects such as the capacity, speed, and propagation delay of each link, as well as the size of the packets, be modeled. Thanks to the ease with which we could experiment with different values of such parameters, we could use Real-Time Maude simulation and model checking to discover: (i) all flaws known by the protocol developers that we were not told about; and (ii) non-trivial design errors that were *not* known by the protocol developers, and that were not found during traditional network simulation and testing.

A Modification of the CASH Scheduling Algorithm [11]. The CASH algorithm is a sophisticated state-of-the-art scheduling algorithm with advanced features for reusing unused execution budgets. Because the number of elements in the queue of unused resources can grow beyond any bound, the CASH algorithm poses challenges to its formal specification and analysis. Using search, we found subtle behaviors in the proposed modification of CASH that lead to missed deadlines. Furthermore, by using a pseudo-random function, we could generate tasks with "random" arrival and execution times, and used rewriting to perform "Monte-Carlo simulations." Extensive such simulation indicated that it is unlikely that the missed deadline could be found by simulation alone.

The OGDC Wireless Sensor Network Algorithm [12]. OGDC is a sophisticated algorithm for wireless sensor networks that tries to maintain sensing coverage of an area for as long as possible. Wireless sensor networks pose many challenges to their modeling and analysis, including novel communication forms (area broadcast with delays for OGDC), treatment of geometrical areas, and the need to analyze both correctness and performance. The OGDC developers used the ns-2 simulator to show that OGDC outperforms other coverage algorithms.

To the best of our knowledge, the Real-Time Maude analysis of OGDC was the first formal analysis of an advanced wireless sensor network algorithm. Using a pseudo-random function to place sensor nodes in pseudo-random locations, we performed a series of simulations of OGDC with up to 800 sensor nodes. The Real-Time Maude simulations gave performance figures quite similar to the ns-2 simulations when we did *not* consider transmission delays. Since the OGDC developers told us that they probably did not include delays in their simulations, this indicates that the Real-Time Maude simulations provided fairly accurate performance estimates of OGDC. Nevertheless, since the definition of the OGDC algorithm does take transmission delays into account, such delays should be modeled. Real-Time Maude simulations *with delays* showed that then the performance of OGDC is more than twice as bad as in the ns-2 simulations. We could also point to a flaw in OGDC that explains this difference in performance.

These facts seem to indicate that Real-Time Maude simulations provide more reliable performance estimates for OGDC than the network simulation tool ns-2.

Embedded Car Software. Real-Time Maude has been used by a Japanese research institute to find several time-dependent bugs in embedded car software used by major car makers. The time sampling approach of Real-time Maude was crucial to detect the bugs, which could not be found by the usual model-checking tools employed in industry.

References

1. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C.: All About Maude - A High-Performance Logical Framework. LNCS, vol. 4350. Springer, Heidelberg (2007)
2. Ölveczky, P.C., Meseguer, J.: Semantics and pragmatics of Real-Time Maude. Higher-Order and Symbolic Computation 20(1-2), 161–196 (2007)
3. Ölveczky, P.C., Meseguer, J.: Specification and analysis of real-time systems using Real-Time Maude. In: Wermelinger, M., Margaria-Steffen, T. (eds.) FASE 2004. LNCS, vol. 2984, pp. 354–358. Springer, Heidelberg (2004)
4. Meseguer, J.: Membership algebra as a logical framework for equational specification. In: Parisi-Presicce, F. (ed.) WADT 1997. LNCS, vol. 1376, Springer, Heidelberg (1998)
5. Ölveczky, P.C., Meseguer, J.: Abstraction and completeness for Real-Time Maude. Electronic Notes in Theoretical Computer Science 176(4), 5–27 (2007)
6. Ölveczky, P.C., Grimeland, M.: Formal analysis of time-dependent cryptographic protocols in Real-Time Maude. In: IPDPS 2007, IEEE, Los Alamitos (2007)
7. Lien, E.: Formal modelling and analysis of the NORM multicast protocol using Real-Time Maude. Master's thesis, Dept. of Linguistics, University of Oslo (2004)
8. Katelman, M., Meseguer, J., Hou, J.: Formal modeling, analysis, and debugging of a wireless sensor network protocol with Real-Time Maude and statistical model checking. Technical report, Dept. of Computer Science, University of Illinois at Urbana-Champaign (In preparation, 2008)
9. Kim, M., Dutt, N., Venkatasubramanian, N.: Policy construction and validation for energy minimization in cross layered systems: A formal method approach. In: IEEE RTAS 2006 (2006)
10. Ölveczky, P.C., Meseguer, J., Talcott, C.L.: Specification and analysis of the AER/NCA active network protocol suite in Real-Time Maude. Formal Methods in System Design 29, 253–293 (2006)
11. Ölveczky, P.C., Caccamo, M.: Formal simulation and analysis of the CASH scheduling algorithm in Real-Time Maude. In: Baresi, L., Heckel, R. (eds.) FASE 2006. LNCS, vol. 3922, pp. 357–372. Springer, Heidelberg (2006)
12. Ölveczky, P.C., Thorvaldsen, S.: Formal modeling and analysis of the OGDC wireless sensor network algorithm in Real-Time Maude. In: Bonsangue, M.M., Johnsen, E.B. (eds.) FMOODS 2007. LNCS, vol. 4468, pp. 122–140. Springer, Heidelberg (2007)