The DIADEM-KRONOS Connection: Bridging the Gap between Implementation and Verification of Hybrid Systems

Akash Deshpande*and Sergio Yovine†
akash@path.berkeley.edu, sergio@path.berkeley.edu
California PATH
University of California at Berkeley
Richmond Field Station Bldg. 452
1301 S. 46th St., Richmond CA 94804

April 24, 1997

1 Introduction

Design and development of complex systems involves several steps: analysis, synthesis, verification, simulation and implementation. These steps are often visited several times in the system design life-cycle. With each visit, the system description moves from the abstract to the concrete, and the modeling frameworks used also get successively finer.

The final product of system design is the implementation. The implementation often interacts with and controls critical equipment. Design flaws that lead to equipment malfunctioning can rarely be tolerated. Hence, it is crucial that the implementation be as close to the verified design as possible.

In the standard methodology used in verification, the system description is given in an abstract specification language. Then algorithmic tools are used to check whether the system models satisfy the desired properties. Table 1 shows sample language-tool combinations available today.

Language	Tools
LOTOS	CAESAR/ALDEBARAN
Verilog, VHDL	COSPAN, SMV with translators
PROMELA	SPIN
Timed and Hybrid Automata	KRONOS, Timed COSPAN, HyTech

Table 1: Languages and Tools for Algorithmic Verification.

The traditional approach has been successful in synchronous hardware design and asynchronous protocol design. The main reasons for this success are as follows.

1. The language semantics are close to the application domain

*Also at Amravan Group, Berkeley, CA.

†Also at Verimag, Cetre Equation, 2 Ave. de Vignate, 38610 Gieres, France.

2. The models are finite

3. The tools perform well even for large systems

4. Good translators exist for mapping between the abstract specification languages and the tool input and output formats

Control system designs have not yet met with the same level of success in verification. The main reasons for this are as follows.

1. Mismatch between the modeling frameworks used for control (linear or nonlinear systems) and required for verification (finite state machines)

2. Undecidability of the verification problem for the more general modeling frameworks

While theoretical results exist for approximating arbitrary nonlinear systems by finite state machines [8], no practical approaches have yet emerged. In practice, control engineers use design tools such as MATLAB and implement the systems in programming languages such as C or C++. This approach is entrenched due to substantial investment in existing code libraries, tools support and training. At the same time, as automatic control systems take over more and more aspects of daily life, their safety is a prime consideration.

To address this joint problem of control system implementation and verification, a design framework is needed to bridge the gap between the two disciplines. We propose a such a design framework and expose the associated tools and methods. We illustrate the use of our approach by applying it to a practical problem. We use the DIADEM system[5] as the platform for control system implementation. We use the KRONOS tool [3, 2, 4], as the platform for verification. Both systems are based on variations of the hybrid automaton model [1].

DIADEM is an object-oriented framework for implementing real-time, event-driven, distributed multi-agent control systems. The underlying model is a dynamic network of multi-rate hybrid automata with the following innovations: object-orientation, auxiliary states, mixed synchronous and asynchronous events, classification of transitions into proactions, reactions and responses, dynamic creation and destruction of components and dynamically changing data and event propagation dependencies between components. At its heart, DIADEM consists of a sophisticated, high-performance real-time event scheduling algorithm based on the fine computation granularity approach[6]. DIADEM provides a graphical modeling environment which automatically generates C++ code to be linked with DIADEM's real-time scheduling system to obtain the implementation. In choosing multi-rate hybrid automata as DIADEM's modeling framework, a careful tradeoff was made between expressiveness and decidability.

KRONOS is a tool for formally verifying correctness of real-time systems. One major design objective of this tool is to provide a powerful and flexible verification engine to be integrated into real-time system design environments in a wide range of application areas. In KRONOS, systems are modeled as networks of timed automata. Properties are usually specified as formulas of the temporal logic TCTL. It is the case in practice that only a subset of such a general specification language is used. For this reason, the KRONOS user-interface provides a variety of complementary features that can be used either independently or in cooperation to analyze the behavior of a system. Such features include computation of sets of safe initial conditions (using backward reachability analysis), generation of unsafe traces (using forward reachability analysis), and optimization of the number of clock variables.

The purpose of the DIADEM-KRONOS connection is to automate the verification of system designs. To this end, we automatically translate DIADEM application models into the KRONOS input language, which has been appropriately upgraded with auxiliary variables to support such a translation. In addition to automating the translation, we must also automate the verification. For this, we impose certain restrictions on the DIADEM model: (1) dynamic creation of components is not allowed, (2) continuous variables may have any rates different from zero, and (3) guards are described by linear inequalities.

2 Case Study

We illustrate our approach by implementing and verifying the FDDI data link protocol. FDDI (Fiber Distributed Data Interface) [7] is a high performance fiber optic token ring Local Area Network. The model implemented by us is taken from [2]. We consider a system composed of N identical stations and a ring network on which the stations send high and low priority messages.

The DIADEM model of a station is shown in Figure 1. The station is sub-classed from DIADEM's `Component` class. It has four auxiliary states: *high-priority-time*—the maximum time allowed to send high priority messages, *TTRT*—the target token rotation time, *station-id*—identifier of the station and *network-process*—identifier of the network to which the station is connected. The station has two continuous states: *TRT* and *THT*. The station uses the timer *TRT* to measure the total amount of time elapsed since the last reception of the token, and it uses *THT* to measure the token rotation delay of the previous round.

The station can be in one of the following three control locations: *idle*, *high-priority*, *low-priority*. In *idle*, the station waits to get the token from the network; in *high-priority*, it sends high priority messages; and in *low-priority*, it sends low priority messages. DIADEM automatically defines the *stop* state for each component, and when it is entered, the component is removed from the system. The station never enters the stop state.

The transition *get-token* from *idle* to *high-priority* is a *reaction*, which means that it is executed asynchronously as soon as the station gets the token from the network. The passage of the token back and forth between the station and the network is modeled using DIADEM's built-in `Alert` mechanism. When taking this transition the station performs two actions: (1) it sets the value of the timer *THT* to the current value of the timer *TRT*, and (2) it sets the value of *TRT* to zero.

At location *high-priority*, the station can spend at most *high-priority-time* time units. Since DIADEM semantics are deterministic, we need two transitions to model this behavior. The *proaction force-end-high-priority-transmission* is taken as soon as *TRT* >= *high-priority-time*, modeling the expiration of the transmission time for high priority messages. The reaction *end-high-priority-transmission* models the termination of the high priority transmission phase because of the absence of messages to send. This is the typical implementation of a timeout mechanism.

Once at location *low-priority*, the station may stay there while the following two conditions hold: (1) there are low priority messages to send, and (2) the value of *THT* is less than or equal to the global constant *TTRT*. Again, this behavior is modeled using two transitions: a proaction *force-end-low-priority-transmission* taken as soon as *THT* >= *TTRT*, and a reaction *end-low-priority-transmission* which models the absence of messages.

3 Conclusions

In the full paper we will provide complete descriptions of the tools and their underlying theories, the case study and the lessons learned by performing this exercise.

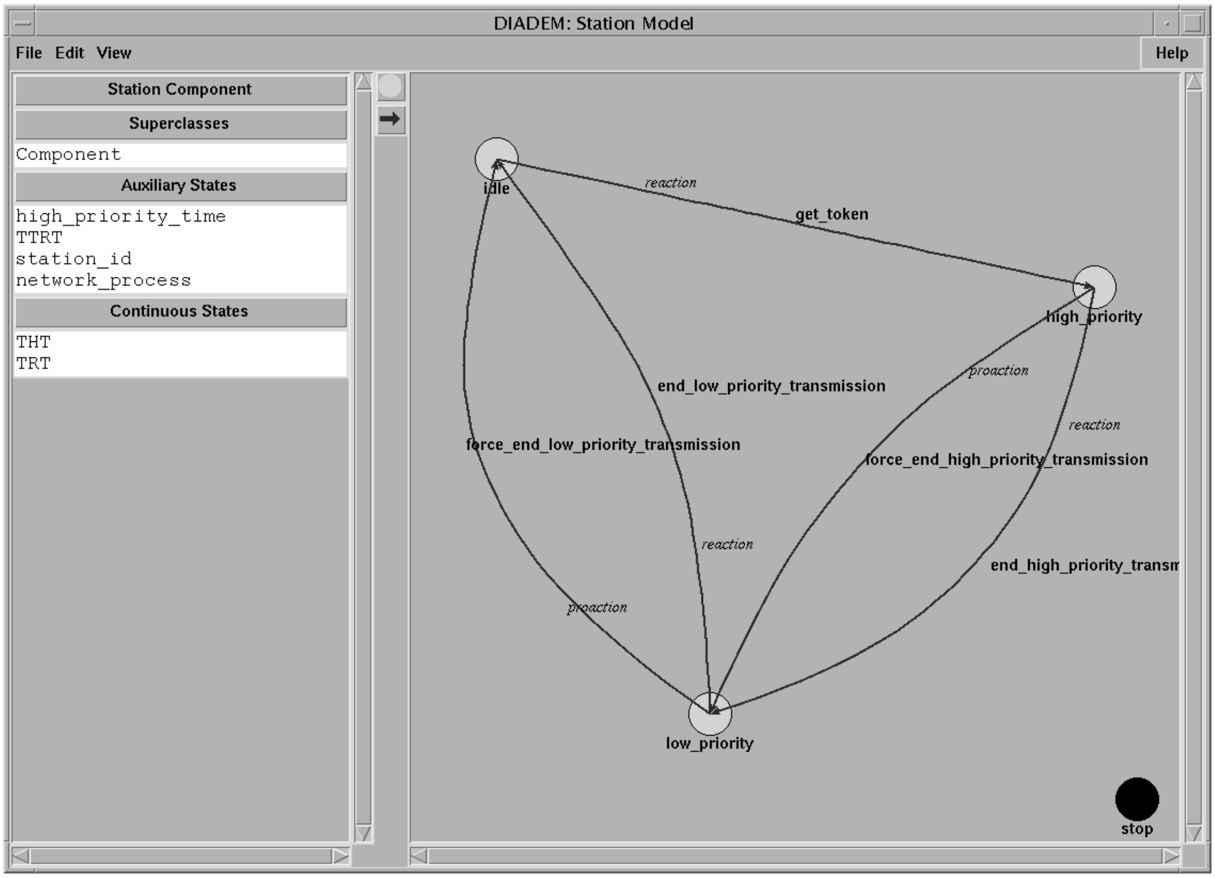

Figure 1: The prototypical station component.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[2] C. Daws, A. Olivero, S. Tripakis, and S. Yovine. The tool KRONOS. In *Hybrid Systems III, Verification and Control*, pages 208–219. Lecture Notes in Computer Science 1066, Springer-Verlag, 1996.

[3] C. Daws and S. Yovine. Two examples of verification of multirate timed automata with KRONOS. In *Proc. 1995 IEEE Real-Time Systems Symposium, RTSS'95*, Pisa, Italy, December 1995. IEEE Computer Society Press.

[4] C. Daws and S. Yovine. Reducing the number of clock variables of timed automata. In *Proc. 1996 IEEE Real-Time Systems Symposium, RTSS'96*, Washington, DC, USA, December 1996. IEEE Computer Society Press.

[5] A. Deshpande. The DIADEM System for Real-Time Operations Management. Submitted to the *Journal of Formal Methods in System Design.*

[6] A. Deshpande. DIADEM's Fine-grained Computation Approach to Real-time Event Scheduling. Submitted to the *18th IEEE Real Time Systems Symposium.*

[7] R. Jain. *FDDI handbook: high-speed networking using fiber and other media.* Addison-Wesley, 1994.

[8] A. Puri, V. Borkar and P. Varaiya. ϵ-Approximation of Differential Inclusions. In *Proceedings of the 34th IEEE Conference on Decision and Control (CDC 95).*