Framework Report (v2)*

VERIMAG

May 2, 2004

AMETIST DELIVERABLE 0.2.2

Project acronym: AMETIST
Project full title: Advanced Methods for Timed Systems
Project no.: IST-2001-35304
Project Co-ordinator: Frits Vaandrager
Project Start Date: 1 April 02
Duration: 36 months
Project home page: http://ametist.cs.utwente.nl/

*This is a revised version of deliverable 0.1.2 from the first year. The introductory section is almost identical.

1 General

The purpose of this report is to summarize the developments that took place within the project and put them in a larger scientific and technological context. We start with a general overview of model-based design and analysis of systems and of the approach we advocate. The principles underlying our approach are coming from the verification of reactive computer systems. We then move to the more specific goal of the project, namely to export this approach to a wide class of systems where quantitative timing information plays a major role and which sometimes are treated by techniques that have not assimilated the computer science way of looking at such problems. Finally we mention the major achievements in the first two years of the project and assess their contribution toward fulfilling that vision.

A large part of engineering and applied mathematics is concerned with building mathematical models of systems and using these models to validate the correct functioning of the system and to choose between design alternatives in order to optimize system performance. The nature of the system in question determines the types of mathematical models that are useful for its analysis. The internal operation of a combustion engine will be modeled by partial differential equations, the car dynamics — by ordinary differential equations and an automatic driver by, say, a finite-state automaton. The fact that a class of models is used in an application domain is not always correlated with its adequacy for solving the domain's problem. In many cases it is a combination of the latter with historical and cultural coincidences. Practitioners, in general, do not have the time to build new clean and rigorous models. They either use what was invented by theoreticians in the past or develop ad-hoc (and, sometimes, ingenious) models that allow them to solve the concrete problems they face in a short time. It usually requires the intervention of theoreticians in order to clean and generalize these models. Academics, on the other hand, tend to live in an imaginary world and have the privilege of being allowed to ignore the feed-back from reality about the relevance of their models. Hence they can publish inertial papers that solve problems whose only significance is internal[1]. This is not the goal of our project. What we want is to establish timed automata as an underlying model for a large class of problems and application domains in the same sense that differential equations underly a large part of physics and traditional engineering, or that transition systems are used in software and hardware engineering.

The class of models that we advocate has its origins in the domain often called *formal verification* whose goal is to prove that certain systems behave correctly for all the external contexts in which they can find themselves. Essentially these are models of discrete dynamical systems whose most notable features are:

- The different *components* of the system in question are clearly identified and the model is given in terms of a composition of simpler models.
- The interaction and mutual influence between the components is easily visible from the interconnection scheme of the components.
- The external environment of the system is also modeled as one or more components of the same kind.
- Each component is modeled by an automaton, the archetypical model of discrete dynamical system. The state transitions are labeled (and enabled) by interaction conditions between the components and by input and output events.
- Putting all components together yields a global automaton in which the origin of each transition can be traced back to the participating components.
- The global system model is the basis of all design activities such as validation, evaluation and optimization. These activities are supported by tools that provide for automatic and semi-automatic analysis.

[1] This is not meant to undermine fundamental purely-theoretical research. Certain (but not all) mathematical objects are worth being studied for their own internal sake.

This approach has been extended in the last decade to treat time-dependent behaviors using the timed automaton model, proposed by Alur and Dill. The project aims at establishing this framework as a unifying model for a large class of timing related problems.

To contrast this approach we discuss the ways these problems are treated today in industry and academia. Like every straw man, this description caricaturizes a bit but we feel it represents the spirit of the state-of-the-art. In many application domains the models are of a more ad-hoc nature without making a distinction between the essential and the accidental details of the problem. The latter may be the syntax of a given programming language, scheduling policies of a given operating systems, etc. This approach may lead to practical solutions but not always to solutions which are scalable to more general problems. The success of formal verification comes partly due to the use of the transition system model (automaton) which abstracts away from such details. Of course, one needs at the end to treat such details in order to solve concrete problems, but much of the insights obtained on the more abstract model could not have been reached on models cluttered with details.

Another feature of commonly-used approaches is the modeling of the system in a form which is already geared toward a particular solution technique, although this is not the most natural and suitable model for the problem at hand. To take an example, if someone is used to linear programming he or she will tend to model problems as linear optimization and if reality persists and refuses to be modeled that way, there will be still an attempt to keep the models close to linear programming, e.g. by adding integer variables. This approach, summarized by the proverb *"When you have a hammer, everything looks like a nail."*, may be useful for short-term resolution of problems, but cannot be recommended at times of "paradigm shift" where new phenomena are to be modeled. We strongly believe that *phenomena come first* and that it is more useful to understand them and devise new formal models whose semantics corresponds faithfully to these phenomena rather than to rush and translate them into one's favorite computational problem. An attack of the computational problems associated with the system analysis should follow only *after* the nature of the problem is understood.

It should be noted that when we say "semantics" or "timed automata" we do not refer to 90% of the work done in these domains. By semantics we do not mean fancy formalisms full of Greek letters and complex definitions. We pick from it the basic idea that a system description, any system description, denotes a set of behaviors, and that these behaviors are the objects of study, those according to which we evaluate system correctness and performance. Likewise, by timed automata, we do not necessarily refer to a particular definition, analysis method or a tool, but rather to the more essential mathematical model of a discrete dynamical system with clock variables.

2 Highlights of the First Two Years

After the first two years of the AMETIST project we can try to summarize the achievement along two major axes, *expressiveness* and *performance*. Expressiveness, is the big promise of TA technology, which is claimed to facilitate the rigorous formulation of complex planning and scheduling situation which cannot be expressed naturally within the standard academic models of operations research, and which are solved in practice using somewhat ad hoc methods. Performance is the other side of the coin, an ongoing fight against the complexity inherent in TA-based analysis and synthesis techniques, a complexity which grows as models become larger and richer.

2.1 Expressiveness

The work on enhancing the expressiveness of the TA framework can be roughly partitioned into two subclasses, model-directed and application-directed. By model-directed enhancement we refer to extensions of the TA model by new features and to the characterization of new scheduling problems and their translation to TA. While these extensions are inspired by practical problems, the main outcomes of this type of work are new rigorous models and algorithms for analysis and synthesis for these models. By application-directed work we refer to the modeling efforts intended to cope with the particularities of specific case-studies and commercial tools in order to provide complete solutions to real-life problems in a form acceptable by their

intended end users. These two directions are complementary and can be viewed as digging a tunnel starting from both sides of the mountain, hoping that they meet in the middle.

2.1.1 Model-directed Extensions

Priced Timed Automata

In ordinary timed automata essentially only time can be optimized as it can be represented by an additional clock that measures its value. There are many situations in which the relevant cost functions are richer, involving cost variables that grow at different rates at different states or are associated with certain transitions. Some motivating examples are memory and power consumption in computers, setup and machine occupation costs in manufacturing as well as non uniform penalties for missing deadlines in systems with soft constraints. Priced timed automata constitute a natural extension of TA with such cost variables. Although the dynamics of these variables renders the automata *hybrid* rather than timed, these variables do not really participate in the rest of the system dynamics (they do not appear in transition guards) and many problem are still decidable for this model. Some new significant decidability results and algorithms for priced timed automata have been obtained in the second year of the project.

Stochastic Extensions

Probabilities can be added to timed automata in two ways. The first is by adding probabilities to transitions, en extensions which transform reachable sets from unions of zones into probabilities on zones. The second and more challenging extension is to replace delay bounds by probabilities on delays, similar to the way this done in continuous time Markov chains. Topics related to both extensions were investigated within the project.

New Scheduling Problems

In the first year one class of problems of scheduling under uncertainty was explored, where the uncertainty is associated with task durations. This year the complementary problem of scheduling under discrete uncertainty has been tackled. It covers the situation where the choice of tasks that need to be executed may depend on the *results* of other tasks, results that become known only after the termination of these tasks. Such situations are very common in scheduling of real time programs, but can also be found in manufacturing, for example when certain production steps may terminate successfully or fail. A modeling framework for this problem using *conditional dependency graphs*, transformed into timed automata with discrete adversaries have been developed along with an algorithm for synthesizing optimal and sub-optimal scheduling policies.

2.1.2 Application-directed Extensions

Real life problems, like those faced by Axxom customers, do not fall exactly within a stylized class of problems like the job shop. Such problems have additional constraints concerning the relative distance between tasks, occupation of certain resources, such as mixing tasks, during the execution of several steps, different penalties for missing deadlines, etc. Many of these details details are hidden inside various Excell tables used by the Orion PI tool. Much effort have been put this year in trying to understand these features, give them a rigorous semantics and devise a language to express them in a way that translates smoothly into timed automata. While this work is less heroic than inventing new mathematical models or algorithms, it is of great importance for the future acceptance of TA based methods. Another extension related to the Axxom case study is concerned with using probabilities to model machine failures, hopefully in a more refined way than the current “macro level” treatment of failures in Axxom tools.

2.2 Performance

The efforts in scaling TA technology can be divided into three classes. The first seeks help from existing techniques of optimization and constraint satisfaction. The second direction is concerned with improvements in the building blocks of existing TA verification tools while the last one attempts to specialize the algorthimics of TA for the special sub class of automata associated with optimal scheduling problems.

2.2.1 Optimization and Constraint Satisfaction

During the second year the applicability of mixed integer linear programming (MILP) as a tool for TA analysis has been explored. An interesting observation is that relaxed models (with "integers" interpreted as reals) may sometimes give more interesting lower bounds on the costs that extend a partial solution and hence can be used for this purpose with reachability based algorithms. For bounded horizon problems, questions related to timed automata are transformed into satisfiability problems for difference logic for which a "second generation" SAT solver was developed.

2.2.2 Improvements in TA Technology

The basic cycle of TA verification tools consists of taking a zone (a conjunction of difference constraints on clocks) and applying to it some operations in order to produce its successors on which the same process is iterated. The number of these zones and the size of their representations is a major bottleneck for TA verification. Zones are typically represented as difference bounds matrices (DBMs) of size quadratic in the number of clocks, and it has already been known that their dimensionality can be reduced in each state to the number of clocks active in that state. More recent work in AMETIST shows that performing a finer analysis of the structure of the TA, may yield for some states DBM representations which can be as good as linear in the number of clocks. It is expected that the integration of these results in a new version of Uppaal will contribute to a significant performance improvement. Among the other important contributions to improving performance of TA tools we mentioned ideas inspired by partial order methods, symmetry reduction and more clever memory management during exploration.

2.2.3 Specialized Technology

The zone based technology has its roots in verification where the temporal uncertainty is viewed as coming from the external environment and the system should be correct with respect to *all* environment choices. Around the beginning of the project it was observed that when uncertainty is associated with the scheduler decisions, for example in deterministic scheduling problems, sometimes there is a unique successor among the uncountably many which gives the optimum (non-lazy schedules). Consequently the problem can be solved without using zones at all but rather by using vectors of clock variables. This way certain problems can be formulated as shortest paths in discrete weighted graphs and be solved much more efficiently. They can also benefit from existing search algorithms for on game graphs in order to find sub-optimal schedules for scheduling with discrete uncertainties.

With all this accumulated effort, there are reason to believe that at the end of the project TA technology will find itself in a much better shape and quite closer to industrial acceptance than it was.

3 Toward a Unifying Framework

In the paper [1], which can be seen as an appendix to this deliverable, a unifying framework for posing and solving problems of optimal control in the presence of adversaries has been proposed. Among the topics treated in that paper we can find a clarification of the notions of set-theoretic (worst-case) vs. probabilistic (average-case) performance criteria and a discussion of three generic approaches for solving such problems (static optimization for bounded horizon problems, backward value iteration and forward best-first search). The second part of the paper shows how problems of scheduling under uncertainty can fit into this framework.

It now seems that one of the major obstacle for gaining acceptance for TA technology lies in TA being a quite non-standard model from the engineer's point of view. Purely continuous systems (differential equations) as well as discrete time models (difference equations, automata) are very well understood, while discrete systems working on "asynchronous" dense time are much less intuitive to grasp upon a first encounter. Perhaps some more pedagogical considerations should be employed while writing papers on the domain, rather than addressing a small community of experts.

The final framework report to be delivered at the end of the project will extend [1] with a more detailed discussion of the state-space approach to dynamic scheduling problems.

References

[1] O. Maler. On optimal and sub-optimal control in the presence of adversaries. 2004. Available from World Wide Web: `http://www-verimag.imag.fr/~maler/Papers/wodes.ps`. Invited talk at WODES'04.