Framework Report (v1)

VERIMAG

June 13, 2003

AMETIST DELIVERABLE 0.2.1

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/

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, an approach whose underlying principles 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 year 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 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 undelying 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 essential 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.
- Each component is modeled by an automaton, the archetypical model of discrete dynamical system. The state transitions are labeled 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.

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.

[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.

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 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 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 too concrete models.

Another feature of commonly-used approaches is the modeling of the system in a form which is already geared toward a certain 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, still there will be 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 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 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 Progress toward these Goals

The project moves toward these goals along several tracks. One is the treatment of real-life case-studies from some candidate application domains to see if, indeed, the proposed methodology is suited for them. The other direction involves new results and better algorithms to support the timed automaton framework, in order to cope with industrial-size problems. The third track, which will become more dominant as the project evolves, aims at synthesizing the accumulated results in order to assess the applicability of the vision and modify it according to feed-back from the field.

2.1 Case Studies

The case-studies were chosen because it was believed a-priori that they represent problems amenable to the kind of solutions the project intends to offer. Of course, during the actual work on them, it may turn out that their detailed nature is different from what was supposed and some adaptation of the work plan is needed in order to solve the problem using alternative solution techniques or transform them to problems that we know how to to solve. A short summary of the situation of each of the case-studies is given below.

2.1.1 The Terma Case-study

This case-study which belongs to the class of computer scheduling problems, fits well into the proposed framework. The functioning of the radar management system can be modeled faithfully with timed automata and indeed models of the case-study were built and analyzed using UPPAAL and SMV. The work on this case-study has led to some new insight on the design of the system which might result in better

designs. The feed-back from Terma on the work so far was very positive and more refined models, with some features not mentioned explicitly in the original description is planned.

2.1.2 The Bosch Case-study

The Bosch case-study, involving a car periphery supervision system belongs also to this class of computer controlled systems. However its preliminary presentation involves some more hybrid aspects such as the car dynamics or properties of the vision system. Consequently a lot of effort is still needed to cope with this hybrid aspects in order to finally abstract them into the timed automaton level. Some efforts in this direction have started with the positive outcome of forcing the engineers to put explicitly the tacit assumptions they use concerning the car physical aspects. In fact, this case-study demonstrates the problematics associated with the "separation of concerns" approach where the limited interaction between control and software engineers is not always sufficient in order to design working efficient systems.

2.1.3 The Cybernetix Case-study

On one hand this case-study exhibits the kind of complex discrete dynamics for which verification methodology has very good solutions. On the other hand it turned out that the quantitative timing aspects have no dominant role in the design. Our choice with respect to this case-study was to work on untimed problems and indeed it has been shown that standard verification tools, when applied to this problem, can derive automatically a solution developed (and patented) by Cybernetix. This is an important step toward the proliferation of verification techniques to a new application domain.

2.1.4 The Axxom Case-study

This case-study is in fact a family of increasingly more complex generic scheduling problems. For the problems introduced so far models based on timed automata proved to be very adequate and have led to good schedules for non-trivial problems. We are now expecting more complex problems in terms of size, performance specification and logical inter-dependencies among tasks. Such problem will induce extensions of the TA-based scheduling methodology to cope with richer problem descriptions.

2.2 Improvements in TA Analysis Technology

The project has produced a lot of improvements in the state-of-the-art of TA analysis. These improvements are necessary in order to cope with the state-explosion problem. Among these we mention symmetry reduction for timed automata, new data-structures for storing reachable states, new memory management policies during the exploration, partial-order methods to reduce redundancy introduced by the interleaving semantics, new abstraction techniques for timed automata and for Markov chains, and more.

In addition to the gradual improvements in the capabilities of the tools, we can note two important conceptual developments. The first is a crystallization of our understanding of various scheduling problems and the effective solutions the TA-based framework can deliver for them. This line of research increases the class of scheduling problems that can be formulated and solved (cost, preemption, uncertainty) and is also concerned with the integration of the methodology in the development chain of embedded software (for example, deriving timed models from programs). It has been, for example, discovered that for many scheduling and synthesis problems where the only uncertainty in the initial model is associated with the choices of the scheduler (or if it is of a purely discrete nature) one does not need to employ the "classical" approach for TA analysis (manipulation of zones) but rather use a kind of more efficient discrete time semantics.

The other direction is the better understanding of the merits and shortcoming of two different approaches for attacking timing problems, namely, the "classical" TA approach based on reachability analysis and fixed-point computation, and the bounded model-checking/optimization approach based on phrasing the existence of a run (resp. the optimal run) as a satisfiability (resp. constrained optimization problem, and

solving this problem using a constraint satisfaction solvers (resp. a MILP optimizer). Although no conclusive results on this issue are presented for the first year, better intuition about the inherent differences of the two approaches have been gained, partly due to the multi-disciplinary nature of the consortium. We intend to pursue this issue (which is also a very active one in untimed verification) further in the coming years.

To conclude, the project attacks timing related problem is several frontiers and several levels (technical, algorithmic, conceptual) and is progressing steadily toward the realization of the vision of establishing timed automata as a unified framework for solving a wide variety of scheduling, planning and other time-dependent problems.