Modeling: Controller Synthesis

VERIMAG

April 1, 2003

AMETIST DELIVERABLE 1.5

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/`

While verification is concerned with proving that a system is correct with respect to all external disturbances, synthesis attempts to mechanize the choice between design options for the system in order to produce a system which is provably correct with respect to all disturbances. Clearly this problem is more difficult than verification although the underlying algorithmic principles are similar.

The first step in formulating a design problem as controller synthesis is to identify the various "players" in the game, that is the components of the system and classify them as *controller* (the part that we want to design), and *environment* (the thing that we do not control and that our designed system has to cope with). In the context of scheduling and resource allocation problems, the controller will be a scheduler which makes decisions concerning resource allocation at certain situations while the environment consists of tasks or jobs that may arrive according to some under-specified constraints and terminate after some under-specified duration. In other contexts the environment actions may include certain faults and the controller role is to guarantee correctness and some acceptable performance albeit those faults.

After composing the model of the controller and its environment we obtain a transition system with two types of transitions, controllable (those of the controller) and uncontrollable (those of the environment). These objects are also known as game graphs, AND-OR graphs, alternating automata, etc. Algorithms for controller synthesis work on such graphs combining techniques from optimization (choose the best among several choices) and verification (evaluate choices according to *all* possible consequences related to uncontrollable transitions).

As verification, synthesis suffers from the state-explosion problem and most of the research effort is to find techniques and tricks to avoid it. Since the interest in synthesis is much more recent than in verification we face another problem: the non-existence of efficient synthesis tools that incorporate the technological progress achieved by the verification community during the last decade in terms of efficient algorithms and data-structures for treating systems of a huge size. Consequently one has to choose between developing new synthesis tools which, by definition, will be less mature then existing verification tools, or to try to realize synthesis algorithms indirectly by "hacking" with existing verification tools. The simplest way to do it is to form the negation of the synthesis problem as a verification query and to obtain the synthesized controller as a "counter-example".

During the first year of Ametist the two approaches has been taken. For case-studies CS1 and CS2, controllers have been synthesized using the verification tool SMV. Finally, a whole framework for treating scheduling problem as controller synthesis has been developed at Verimag, specializing for various problems and accompanied by prototype tools. Below one can find a short summary of each of these developments with references for more details.

- Controller synthesis for CS1 (KUN)

  Using the Berkeley SMV symbolic model checker we synthesize, under certain error assumptions, a controller for the smart card personalization system. The controller that we synthesize, and of which we prove optimality, has been previously patented. Due to the large number of states (which is well-beyond 10^6), this control synthesis problem appears to be out of the scope of existing tools for controller synthesis, which typically use some form of explicit state enumeration. Our result provides new evidence that model checkers can be useful to tackle industrial sized problems in the area of scheduling and control synthesis. [6]

- Controller synthesis for CS2 (Weizmann)

  In [10] we describe a scheduler synthesis for the memory interface card offered by Terma as a case study. A model of the system in the SMV modeling language is introduced. This model can be used to produce valid schedules for systems with small number of buffers. A method to synthesize an optimal (in terms of buffer sizes) scheduler for the full system is also given. The schedule is verified by the SMV model checker using the above- mentioned model. Some extensions to cards with other parameters are discussed.

- Timed control with partial observability (Aalborg)

  Until now all work dealing with the problem of control synthesis for a timed plant and a timed specification have assumed that the controller has complete information about the plant. In [5] the natural extension of the synthesis problem to partial observatbility is investigated. In this setting the plant can have internal unreadable clocks and actions that are not readable by the controller. Our main result is that in this wider framework, if the resources of the controller are fixed, the control synthesis problem is decidable for specifications describing the undesired behaviors. On the other hand, if the resources of the controller are not a priori fixed, the control synthesis problem becomes undecidable even in the case of deterministic specifications. This situation is in contrast to the setting of complete observability, where the problem is decidable for deterministic specifications.

- Scheduling under Uncertainty (Verimag, LIF)

  The problem of job-shop scheduling under temporal uncertainty (tasks are known but their duration is uncertain) has been studied in [2, 1]. A new optimality criterion which can be seen as an adaptive generalization of worst-case has been defined along with an implemented synthesis algorithm optimal in the above sense. The algorithms uses the zone library of the verification tool IF/Kronos. In addition the case of probabilistic uncertainty in task durations has been studied and an optimal synthesis algorithm for this special type of a continuous-time Markov decision process has been developed. Currently work is being conducted on another type of uncertainty, where the environment can choose between several tasks according the the results of previous tasks. It is hoped that efficient forward synthesis algorithms (similar to those developed already in [4]) could be developed. This work also raises some fundamental problems concerning the analysis of game automata, problems that have not been treated by AI or Game Theory.

- Software Scheduling as Controller Synthesis (Verimag)

  In [3] the problem of synthesizing schedulers for real-time software has been addressed and a general framework for phrasing it as a controller synthesis problem has been developed. In [9] the more general question concerning the place of the synthesis algorithm in the development cycle of real-time embedded systems has been considered, in particular the automatic translation from annotated real-time programs architecture descriptions to scheduler synthesis problems.

- Synthesizing Safe Schedulers with Guaranteed Quality of Service (Verimag)

  In [7] we present a new scheduler architecture, which permits adding QoS policies to the scheduling decisions. We also present a new scheduling synthesis method which allows a designer to obtain a safe scheduler for a particular application and at the same time helps him in analyzing the task interactions and the overall system behavior. Our scheduler architecture and scheduler synthesis method have not been developed for a particular application model and, therefore, can be used for heterogeneous applications, where there are periodic tasks, event-driven ones and tasks which are always enabled and where the tasks communicate through various synchronization primitives. Finally, we present a prototype implementation of this scheduler architecture and related mechanisms on top of an open-source OS for embedded systems.

- Control with Bounded Resources (Verimag)

  In [8] we propose models that capture the influence of computation on the performance of computer-controlled systems, and make it possible to employ computational considerations in early stages of the design process of such systems. The problem of whether it is possible to meet performance requirements given resource constraints is phrased as a problem of synthesizing switching controllers for hybrid automata, for which we give algorithms that in some cases are guaranteed to terminate and in others can solve the problem in an approximate manner.

To summarize, the project has advanced the state-of-the-art and the understanding of the issue of automatic controller synthesis for discrete and timed systems, and will contribute during the second year to the development and improvement of synthesis algorithms.

References

[1] Y. Abdeddam, E. Asarin, and O. Maler. On optimal scheduling under uncertainty. In H. Gargamel and J. Hatcliff, editors, *Proc. TACAS*, volume 2619 of *LNCS*. Springer. Available from World Wide Web: `http://www-verimag.imag.fr/˜maler/Papers/uncertain.ps`.

[2] Yasmina Abdeddam. *Scheduling with Timed Automata*. PhD thesis, INPG Grenoble, November 2002. Available from World Wide Web: `http://www-verimag.imag.fr/˜maler/Papers/thesis-yasmina.ps`.

[3] K. Altisen, G. Goessler, and J. Sifakis. Scheduler modeling based on the controller synthesis paradigm. *Journal of Real-Time Systems, special issue on ”Control Approaches to Real-Time Computing”*, 23:55–84, 2002. Available from World Wide Web: `http://www-verimag.imag.fr/˜sifakis/paper_final.pdf`.

[4] K. Altisen and S. Tripakis. Tools for controller synthesis of timed systems. In *RT-TOOLS*, 2002. Available from World Wide Web: `http://www-verimag.imag.fr/˜tripakis/final-rttools02.pdf`.

[5] P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. In *To appear in Proceedings of CAV 2003*, Lecture Notes in Computer Science. Springer Verlag, 2003. Available from World Wide Web: `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-5.rr.ps`.

[6] B. Gebremichael and F.W. Vaandrager. Control synthesis for a smart card personalization system using symbolic model checking. Report NIII-R0312, Nijmegen Institute for Computing and Information Sciences, University of Nijmegen, May 2003. Available from World Wide Web: `http://www.cs.kun.nl/ita/publications/papers/fvaan/smart.html`. Submitted.

[7] Ch. Kloukinas and S. Yovine. Synthesis of safe, qos extendible, application specific schedulers for heterogeneous real-time systems. In *Proceedings of ”5th Euromicro Conference on Real-Time Systems (ECRTS'03)”*, Porto, Portugal, July 2003. Available from World Wide Web: `http://www-verimag.imag.fr/PEOPLE/Christos.Kloukinas/IN2P3/kloukinas_yovine.pdf`.

[8] O. Maler, B. Krogh, and M. Mahfoudh. On control with bounded computational resources. In W. Damm and E-R Olderog, editors, *FTRTFT'02*, volume 2469 of *LNCS*, pages 147–164. Springer. Available from World Wide Web: `http://www-verimag.imag.fr/PEOPLE/Oded.Maler/Papers/resources.ps`.

[9] J. Sifakis, S. Tripakis, and S. Yovine. Building models of real-time systems from application software. In *Proceedings of the IEEE Special issue on modeling and design of embedded*, pages 91(1):100–111, January 2003. Available from World Wide Web: `http://ieeexplore.ieee.org/iel5/5/26369/01173177.pdf?isNumber=26369&prod=JNL&arnumber=1173177&arSt=+3&ared=+10&arAuthor=Sastry%2C+S.%3B+Sztipanovits%2C+J.%3B+Bajcsy%2C+R.%3B+Gill%2C+H.`

[10] Gera Weiss. Modeling smart-card personalization machine with LSCs. Research report, Weizmann, 2003. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/WISPublications/cybernetix.zip`.