Project no.:	**ICT-FP7-STREP-214755**
Project full title:	**Quantitative System Properties in Model-Driven Design**
Project Acronym:	**QUASIMODO**

Deliverable no.: D3.5

Title of Deliverable: Extended timed automata for scheduling

Contractual Date of Delivery to the CEC:	**Month 18**
Actual Date of Delivery to the CEC:	**Month 18 (July 1, 2009)**
Organisation name of lead contractor for this deliverable:	**CNRS**
Author(s):	**François Laroussinie, Martin Neuhäußer, Kim G. Larsen, Frits Vaandrager.**
Participant(s):	**P01 AAU, P02 ESI/RU, P03 CNRS, P04 RWTH, P05 SU**
Work package contributing to the deliverable:	**WP 3**
Nature:	**R+P**
Version:	**0.99**
Total number of pages:	**13**
Start date of project:	1 Jan. 2008 **Duration:** 36 month

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This deliverable reports on the works carried out inside the QUASIMODO consortium on the use of timed automata-based models for modelling and analyzing scheduling problems.

Keyword list: timed automata, stopwatches automata, (preemptive) scheduling, Markov chains, Markov decision processes.

Contents

Introduction

Scheduling concerns the allocation of resources to tasks (or jobs) over time in order to achieve some goals. Scheduling problems occur in many different domains and a vast amount of research has been carried out in this area. But very often in the literature, the scheduling problems are perfectly known: the number of jobs, their duration, their period (for periodic tasks), *etc.* But in practice these problems are driven by uncertainty and this makes their analysis much more difficult.

In this framework, we present three works:

- an extension of timed automata with stopwatches to model preemptive scheduling with uncertainty in the duration and the period of the tasks;
- a use of the tool UppAal-Tiga to model an industrial scheduling problem with uncertainty in the job arrivals;
- an algorithm to analyze continuous-time Markov decision processes (CTMDPs) that can be used to model stochastic job scheduling problems.

In this framework, one can consider two kinds of problem: the *schedulability analysis* where one aims at verifying that a given scheduling policy is correct for a given set of tasks, and the *synthesis of scheduling policy* where the objective is to build a correct scheduling policy from a given problem (a set of tasks, *etc.*).

Stopwatches automata (SwTAs)

A stopwatch is a clock that can be stopped and restarted. Contrary to clocks that progresses continuously with time elapsing, a stopwatch can for example be used to measure the accumulated time spent in a given set of locations during an execution. This extension is very powerful and leads to undecidability of reachability. An important application area of stopwatches automata is the modeling of preemptive scheduling problem: the current value of the clocks associated with a process can be stored after a preemption from the scheduler and the process can be resumed after some delay. SwTAs are then a natural formalism to handle schedulability analysis.

Timed games automata (TGAs)

This model extends standard timed automata by marking edges as either controllable or uncontrollable. This defines a two players game with on the one side the controller (mastering the controllable edges) and on the other side the environment (mastering the uncontrollable edges). Winning conditions of the game can be specified through TCTL formulas: then the controller has to ensure that the formula holds for every execution. A classical and simple case consists in considering reachability questions: the controller has to reach some given winning states (or conversely to avoid some bad state).

Continuous-time Markov chains (CTMCs) and Continuous-time Markov decision processes (CTMDPs)

Continuous-time Markov chains are one of the most important models in performance and dependability analysis. They are exploited in a broad range of applications, and constitute the underlying semantical model of a plethora of modeling formalisms for real-time probabilistic systems.

Continuous-time Markov decision processes (CTMDPs), also known as *controlled* Markov chains, have been used for, among others, the control of queueing systems, epidemic, and manufacturing processes. The analysis of CTMDPs is focused on determining optimal schedulers for criteria such as expected total reward and expected (long-run) average reward.

In the sequel, we present our recent results on all three models and their use for scheduling analysis.

1 Stopwatch Automata for Scheduling

1.1 Stopwatch Automata for Scheduling

1.1.1 Participants

- François Laroussinie, Eudes Petonnet, CNRS/LIAFA, University Paris 7, France

1.1.2 Contribution

Timed automata have been used to model scheduling problems in many papers [AAM06, FMPY06]. Here we consider the *verification* of a preemptive scheduling policy (given as an automaton) for a set of periodic tasks with uncertainty about their periods and their durations. The possibility to preempt a task (in order to restart it later) makes natural the use of stopwatches in the model of tasks.

More precisely, we consider the case of n periodic tasks (or jobs) and *one* processor such that:

- the period of task T_i belongs to some interval $[p_i; P_i]$;
- every task T_i has a duration that may change in some interval $[d_i; D_i]$;

Moreover every task T_i has a deadline DL_i.

In this setting uncertainty is about the period *and* the duration of the tasks. We assume that the scheduling policy is given by an automaton S that may suspend and/or (re)start a task.

Task T_i can be modeled as a simple stopwatch automaton A_{T_i} with one clock c_i measuring the elapsed time since the beginning of the last period and one stopwatch w_i measuring the total execution time of T_i. Then a task can terminte when $w_i \in [d_i; D_i]$ and has to end before $x_i = DL_i$.

Given the automaton S and n automata A_{T_1}, ..., A_{T_n}, the problem consists in verifying whether $x_i < DL_i$ is always true along the executions (for any i).

Model-checking stopwatches automata is undecidable. Nevertheless the structure of the T_is is special and we can also distinguish several cases depending on the form of the scheduler S:

- S is a timed automaton;
- S is an *untimed* automaton that may suspend/start tasks when a new task is available or when a task terminates;
- S is active: if there exists some pending tasks, the processor is busy.
- S is well-founded: when S suspends T to start T', then T' will be completed before restarting T.

Given these different types of scheduler, we have shown the following results [Pet08]:

- if every task has a tight period (*i.e.* $p_i = P_i$ for any i), then the problem is decidable (even if S is a general timed automaton);
- if (1) the periods are not tight and (2) S is a timed automaton, then the problem is undecidable;
- if (1) the durations of the tasks are tight and (2) S is an untimed active and well-founded automaton, then the problem is decidable.

This work is an on-going work. Especially we are considering the case where the scheduler is untimed, active and well-founded and with uncertainty on periods and durations.

References

[AAM06] Yasmina Abdeddaïm, Eugene Asarin, and Oded Maler. Scheduling with timed automata. *Theor. Comput. Sci.*, 354(2):272–300, 2006.

[FMPY06] Elena Fersman, Leonid Mokrushin, Paul Pettersson, and Wang Yi. Schedulability analysis of fixed-priority systems using timed automata. *Theor. Comput. Sci.*, 354(2):301–317, 2006.

[Pet08] Eudes Petonnet. Application des automates temporisés en ordonnancement. Master's thesis, University Paris 7, 2008.

1.2 Scheduling using stopwatch automata with UPPAAL 4.1

1.2.1 Participants

- Thonmas Bøgholm, Alexandre David, Jacob Illum Rasmussen, Henrik Kragh-Hansen, Kim G. Larsen, Petur Olsen, Arne Skou, Bent Thomsen; CISS, Aalborg University.

1.2.2 Contribution

As a main contribution of last years deliverables, we offered in the release of UPPAAL 4.1. the availability of stop-watches (*i.e.*, clocks that may be stopped and restarted during behaviour of the timed automata) with the the support of an efficient, zone-based over-approximate state-space exploration. During the second year this extension has in a number of instances been applied to schedulability analysis (frameworks) of complex, multi-processor systems with preemptive scheduling principle. Also, as a prerequisite for performing schedulability analysis, the extension has been applied to providing safe bounds of WCET (worst-case execution time) of executable (C-)code.

Schedulability Analysis of Safety Criticial Hard Real-Time Java In [BKHO+08] we present a novel approach to schedulability analysis of Safety Critical Hard Real-Time Java programs. The approach is based on a translation of programs, written in the Safety Critical Java profile introduced in [SSTR07] for the Java Optimized Processor [Sch05], to timed automata models verifiable by the Uppaal model checker. Schedulability analysis is reduced to a simple reachability question, checking for deadlock freedom. Model-based schedulability analysis has been developed by Amnell et al. in the TIMES tool [AFM+03], but has so far only been applied to high level specifications, not actual implementations in a programming language. Experiments show that model-based schedulability analysis can result in a more accurate analysis than possible with traditional approaches, thus systems deemed non-schedulable by traditional approaches may in fact be schedulable, as detected by our analysis. Our approach has been implemented in a tool, named SARTS, successfully used to verify the schedulability of a real-time sorting machine consisting of two periodic and two sporadic tasks. SARTS has also been applied on a number of smaller examples to investigate properties of our approach. More information may be found on the SARTS home-page `http://sarts.boegholm.dk/`.

Model-based framework for schedulability analysis using UPPAAL 4.1 [DILS09] Embedded systems involve the monitoring and control of complex physical processes using applications running on dedicated execution platforms in a resource constrained manner in terms of for example memory, processing power, bandwidth, energy consumption, as well as timing behavior. Viewing the application as a collection of (interdependent tasks) various scheduling principles may be applied to coordinate the execution of tasks in order to ensure orderly and efficient usage of resources. Based on the physical process to be controlled, timing deadlines may be required for the individual tasks as well as the overall system. The challenge of schedulability analysis is now concerned with guaranteeing that the applied scheduling principle(s) ensure that the timining deadlines are met.

For single processor systems, industrial applied schedulability analysis tools include Sys Corporation, and RapidRMA from TriPacific based on Rate Monotonic Analysis. More recently SymTA/S has emerged as an efficient tool for system-level performance and timing analysis based on formal scheduling analysis techniques and symbolic simulation. These tools benefit from great succes in real-time scheduling theories; results that were developed in the 1970ies and the 1980ies, and are now well-established. However these theories and tools have become seriously challenged by the rapid increase in the use of multi-cores and multiprocessor system-on-chips (MPSoC). To overcome the limitation to single-processor architectures, applications of simulation have been pursued Though extremely useful for early design exploration by providing very adequate performance estimates for example memory usage and energy con- sumption as well as options for parallelizations, the use of simulation makes the schedulability analysis provided by these tools unreliable: though no deadline-violation may be revealed after (even extensive) simulation, there is no guarantee that this will never occur in the future. For systems with hard real-time requirements this is not satisfactory.

During recent years the use of real-time model checking has become an attractive and maturing approach to schedulability analysis providing absolute guarantees: if after model checking no

violations of deadlines have been found, then it is guaranteed that no violations will occur during execution. In this approach, the (multiprocessor) execution platform, the tasks, the interdependencies between tasks, their execution times, and mapping to the platform are modeled as timed automata allowing efficient tools such as Uppaal to verify schedulability using model checking. The paper [DILS09] offers a Uppaal modeling framework, that may be instantiated to suit a variety of scheduling scenarios, and that can be easily extended. In particular, the framework includes: A rich collection of attributes for tasks, including:

- off-set, best and worst case execution times, minimum and maximum interarrival time, deadlines, and task priorities.
- Task dependencies.
- Assignment of resources, for example processors or busses, to tasks.
- Scheduling policies including First-In First-Out (FIFO), Earliest Deadline First (EDF), and Fixed Priority Scheduling (FPS).
- Possible preemption of resources.

The combination of task dependencies, execution time uncertainty and preemption makes schedulability of the above framework undecidable. However, the recent support for stopwatch automata in Uppaal 4.1 leads to an efficient approximate analysis that has proved adequate on several concrete instances.

WCET Analysis of ARM Processors using Real-Time Model Checking In order to produce a reliable and efficient execution schedule for a real-time system (RTS), scheduling algorithms need safe and sharp worst-case execution times (WCETs) for the processes in the system. The developed method in [DOT+09] utilises real-time model checking performed by the model checker UPPAAL to determine these WCETs. The method is able to analyse program code found in real systems, and an extensive evaluation has been conducted using the benchmark programs published by Malerdalen WCET Research Group. The WCET for a process depends on the hardware platform that the process is executing on, thus the method is designed to allow a high degree of exibility. For example, support for new processors only requires a model of the new processor. Modern processors utilise techniques such as caching and pipelining, which increase the average number of calculations that can be executed per time unit. Since these techniques are also found in many processors intended for embedded devices, such as members of the widely deployed ARM7 and ARM9 families, a modern WCET analysis method must take them into account to be useful. The presented method models these techniques in a modular and independent fashion. It is important to be aware that caching and pipelining in certain instances can lead to timing anomalies, which complicate WCET analysis to a great extent. By introducing more non-determinism in the applied models, the presented method can be extended to handle the presence of timing anomalies. The method is implemented in the tool METAMOC that may be downloaded from `http://metamoc.martintoft.dk/`.

References

[AFM+03] Tobias Amnell, Elena Fersman, Leonid Mokrushin, Paul Pettersson, and Wang Yi. Times: A tool for schedulability analysis and code generation of real-time systems. In Kim Guldstrand Larsen and Peter Niebert, editors, *FORMATS*, volume 2791 of *Lecture Notes in Computer Science*, pages 60–72. Springer, 2003.

[BKHO+08] Thomas Bøgholm, Henrik Kragh-Hansen, Petur Olsen, Bent Thomsen, and Kim Guldstrand Larsen. Model-based schedulability analysis of safety critical hard real-time java programs. In *JTRES*, pages 106–114, 2008.

[DILS09] A. David, Jacob Illum, K.G. Larsen, and A. Skou. *Model-based Framework for Schedulability Analysis using UPPAAL 4.1.*, chapter 1. CRC Press, 2009.

[DOT+09] A. Dalsgaard, M.C. Olesen, M. Toft, R.R. Hansen, and K.G. Larsen. Wcet analysis of arm processors using real-time model checking. In *Doctoral Symposium on Systems Software Verification (DS SSV'09), Real Software, Real Problems, Real Solutions (technical report)*, 2009.

[Sch05] M. Schoeberl. *JOP: A Java Optimized Processor for Embedded Real-Time Systems*. PhD thesis, Vienna University of Technology, 2005.

[SSTR07] M. Schoeberl, H. Sondergaard, B. Thomsen, and A. P. Ravn. A profile for safety critical java. In *ISORC07: Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, pages 94–101, 207.

2 Optimal Adaptive Scheduling with Uppaal Tiga

2.1 Participants

- Israa AlAttili, Fred Houben, Georgeta Igna, Steffen Michels, Feng Zhu, Frits Vaandrager : ESI, Radboud University Nijmegen, the Netherlands

2.2 Contribution

This work is about the computing of optimal scheduling strategies for an industrial case study where the uncertainty is crucial. This analysis has been done with the tool UppAal-Tiga: it extends the model-checker UppAal [LPY97] for timed automata with an algorithm to deal with timed games [CDF+05].

Overview of the method. The method that has been used is based on timed model-checking. Components of a system are modeled as dynamical systems with a state space and a well-defined dynamics. All that can happen is expressed in terms of behaviors that can be generated by the dynamical systems; these constitute the semantics of the problem. Verification, optimization, synthesis and other design activities explore and modify system structure so that the resulting behaviors are correct, optimal, etc. Using this approach, it is sometimes possible to derive schedules that are of comparable quality as those that were provided by an industrial tool [BBHM05]. Nevertheless dealing with uncertainty makes the problem much more difficult to solve and this motivates the use of new techniques.

Scheduling and UppAal-Tiga. In a scheduling context, uncertainty can be modeled using uncontrollable edges of timed games. Uppaal Tiga is then able to synthesize strategies for controlling the system such that scheduling objectives are met irrespective of the timing of uncontrollable edges. In order to demonstrate the practical usefulness of Uppaal Tiga for solving scheduling problems with uncertainty, we have applied the tool to an industrial case study from Océ Technologies that concerns the scheduling of a state-of-the-art image processing pipeline of a printer. In [IKY+08], an initial version of this scheduling problem has been described and analyzed using three different modeling frameworks: timed automata (Uppaal), colored Petri nets and synchronous dataflow. None of the models in [IKY+08] incorporated uncertainty and in particular it was assumed that the arrival times of new jobs are known in advance. In reality, of course, the arrival time of new printer jobs is typically unknown (arrival times are the most significant source of uncertainty in this application domain).

In a new work [AHI+09], we use Uppaal Tiga to generate optimal scheduling strategies for scenarios in which the arrival times of certain jobs is unknown. Previous industrial applications of Uppaal Tiga deal with the synthesis of controllers (for example [CJL+09]). The present work describes the first application to an industrial scheduling problem.

References

[AHI+09] I. AlAttili, F. Houben, G. Igna, S. Michels, F. Zhu, and F.W. Vaandrager. Adaptive scheduling of data paths using uppaal tiga. In *Proceedings First Workshop on Quantitative Formal Methods: Theory and Applications (QFM'09), Eindhoven, the Netherlands, 3rd November 2009*, volume 13 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–12, 2009.

[BBHM05] Gerd Behrmann, Ed Brinksma, Martijn Hendriks, and Angelika Mader. Production scheduling by reachability analysis - a case study. In *19th International Parallel and Distributed Processing Symposium (IPDPS 2005), 4-8 April 2005, Denver, CA, USA*. IEEE Computer Society, 2005.

[CDF+05] Franck Cassez, Alexandre David, Emmanuel Fleury, Kim G. Larsen, and Didier Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *Proc. of the 16th Int. Conf. on Concurrency Theory (CONCUR'05)*, volume 3653 of *Lecture Notes in Computer Science*, pages 66–80. Springer, August 2005.

[CJL+09] Franck Cassez, Jan J. Jessen, Kim Guldstrand Larsen, Jean-François Raskin, and Pierre-Alain Reynier. Automatic Synthesis of Robust and Optimal Controllers – An Industrial Case Study. In *Proc. of the 12th International Conference on Hybrid Systems: Computation and Control (HSCC'09)*, volume 5469 of *Lecture Notes in Computer Science*, pages 90–104, San Francisco, CA, USA, April 2009. Springer.

[IKY+08] Georgeta Igna, Venkatesh Kannan, Yang Yang, Twan Basten, Marc Geilen, Frits W. Vaandrager, Marc Voorhoeve, Sebastian de Smet, and Lou J. Somers. Formal modeling and scheduling of datapaths of digital document printers. In *Formal Modeling and Analysis of Timed Systems, 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings*, volume 5215 of *Lecture Notes in Computer Science*, pages 170–187. Springer, 2008.

[LPY97] Kim G. Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. *Journal of Software Tools for Technology Transfer (STTT)*, 1(1–2):134–152, 1997.

3 Analysis of time-bounded reachability probabilities in continuous-time Markov decision processes

3.1 Participants

- Martin Neuhäußer, RWTH Aachen, Germany;
- Lijun Zhang, Saarland University, Germany.

3.2 Contributions

As part of the QUASIMODO project, we developed a discretization technique which allows to analyze time-bounded reachability probabilities in continuous-time Markov decision processes (CTMDPs). In these models, the sojourn time distribution of the current state is exponentially distributed and influenced by an action that is chosen nondeterministically. So far, the analysis of randomly timed systems that exhibit nondeterministic choices has received scant attention. In this part of the QUASIMODO project, we overcome this shortcoming in the theory and provide an efficient and quantifiably precise model checking algorithm for a broad class of CTMDPs and time- and history-dependent schedulers.

More precisely, we use a technique to compute the maximum probability to reach a set G of goal states within a given time bound z under all schedulers. For this result we prove that for time-bounded reachability, it suffices to consider the class of total time positional deterministic schedulers, which base their decision only on the elapsed time and on the current state. As a consequence, we are able to characterize the maximum time-bounded reachability probability as the least fixed point of a higher-order operator which involves integration over the time domain. This allows to reduce the time-bounded reachability problem for CTMDPs to the problem of computing step-bounded reachability probabilities in discrete-time MDPs. More precisely, we approximate the behavior of the CTMDP up to an a priori specified error bound $\varepsilon > 0$ by defining its discretized MDP such that its maximum step-bounded reachability probability coincides (up to ε) with the maximum time-bounded reachability probability of the underlying CTMDP.

In this way, we derive a quantifiably correct approximation method that solves the time-bounded reachability problem for CTMDPs by reducing it to the step-bounded reachability problem in MDPs. The latter is a well studied problem and can be solved efficiently by linear programming techniques, policy iteration or value iteration algorithms. Hence, our approach is also efficient from a complexity theory point of view.

Although we present all results only for maximum time-bounded reachability probabilities, it can easily be adapted to the dual problem of determining the minimum time-bounded reachability probability and even further, to model checking large fragments of the continuous stochastic logic.

This result is related to the scheduling problem because CTMDP are very convenient to model scheduling problems with uncertainty. Indeed we consider the stochastic job scheduling problem (sJSP) from [BDF81] and we illustrate our method on it in [NZ09].

References

[BDF81] J. L. Bruno, P. J. Downey, and G. N. Frederickson. Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the ACM*, 28:100 – 113, 1981.

[NZ09] Martin R. Neuhäußer and Lijun Zhang. Time-bounded reachability in continuous-time markov decision processes. Technical Report 2009-12, RWTH Aachen, Department of Computer Science, 2009.