Analysis & Tools: Structure Exploitation

KUN

June 9, 2005

AMETIST DELIVERABLE 2.1.2

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: 39 months
Project home page: `http://ametist.cs.utwente.nl/`

In Task 2.1 the AMETIST consortium studied the application of abstraction, compositionality and structure exploitation as key techniques in controlling and reducing the complexity involved in analysing real-time system models.

Abstraction, compositionality and structure exploitation techniques are well established for model checking and theorem proving approaches to discrete state system verification. *Abstraction* — either based on generic principles (like from timed to untimed systems, infinite data domains to finite one) or on case-specific user suggested simplifications — serves to simplify the models by omitting aspects that are not relevant for analysis. *Compositionality* provides a divide-and-conquer approach to manage complexity in which for any global specification there exists a decomposed specification for each module that together allow the deduction of that global property. For the verification of finite-state systems a number of techniques for *exploiting the composite structure* of the model exists. Partial order reduction and compositional backwards reachability are methods which help to reduce search spaces for loosely coupled concurrent components. Likewise, methods exploiting symmetries and hierarchical structure have shown to be very successful. An overview of the project's results relating to abstraction and compositionality has been presented in Deliverable 2.1.1 [2]. The present deliverable focuses on structure exploitation.

Actually, it is hard to draw a firm line between the different techniques for fighting state space explosion that have been studied within Task 2.1. Symmetry reduction techniques exploit symmetries in the syntactic description of a model to statically determine an equivalence relation on the set of states. Such an equivalence relation, essentially, determines an abstraction (bisimulation) of the original system with a reduced number of states: during state space exploration it suffices to explore the successors of only one state in each (reachable) equivalence class. Similarly, partial order reduction techniques typically lead to an equivalence relation ("indenpendence") on events/transitions, thereby providing a different type of abstraction. Again, only one representative of each equivalence class needs to be explored, which reduces the effort of state space exploration. We have chosen to discuss in the present deliverable abstractions that can be computed statically and that exploit the composite structure of systems. For a discussion of abstractions that are computed dynamically (as e.g. in counterexample guided abstraction refinement) and abstractions that involve the value domain of clock variables (as e.g. in active clock reduction) we refer to Deliverable 2.1.1 [2].

It is also difficult to draw a firm line between the work in Task 2.1 and that within Task 2.3 on state space representations: only by choosing the right representation of states it becomes possible to compute abstractions and to exploit the structure of a model. The primary reason why OBDDs have been so successful in hardware verification is that these data structures manage to exploit the structure that is present in the description of models of many hardware components.

The main contributions of AMETIST that relate to structure exploitation concern: (a) partial order reduction, (b) symmetry reduction, (c) an approach to the verification of logic controllers given as Sequential Function Charts, and (d) hybrid systems. We will summarize these contributions below.

Partial Order Reduction

In a setting of (untimed) finite state machines, partial order reduction techniques have been very successfull in dealing with the state space explosion problem, and they are implemented in several state-of-the-art model checkers, e.g. SPIN. In an attempt to further improve the efficiency of timed automata model checkers, several researchers have tried to transfer partial order reduction methods to the timed setting.

Partial order methods basically try to fight the state space explosion problem by exploiting knowledge about the structure of the reachability graph, in particular *independence* of pairs of transitions of loosely related parts of a complex system. Such pairs a and b *commute*, i.e., a state s allowing a sequence ab of transitions to state s' also allows ba and this sequence also leads to s'. Hence, for reachability analysis only one of the interleavings has to be explored. However, this type of commutativity is lost in classical symbolic algorithms for timed automata, which represent sets of possible clock values by symbolic states. Consider two "independent" actions a resetting clock x to 0, and b resetting clock y to 0. Exectuting a first and then b means that afterwards (time may have elapsed) $x \geq y$ whereas executing b first and then a implies that afterwards $x \leq y$. The result of this is that in tools like Uppaal and KRONOS, ab and ba lead to *different* and in fact incomparable symbolic states.

In [24, 16, 23, 17], we propose what we believe to be the first satisfactory partial order reduction technique for timed automata. Our goal is to preserve independence (commutativity of transitions) from the original timed automaton at the symbolic level, thus fully avoiding state splitting, yet avoiding problems of previous similar approaches with maximal bounds abstraction. We achieve this goal by (1) lifting the theory of Mazurkiewicz traces to timed words and symbolic state exploration, (2) examining symbolic path exploration from a formal language point of view, and (3) by splitting the concerns of (abstraction free) successor computation and zone comparison by a new abstraction related to maximal bounds. The theory results in data structures and algorithms that we have experimentally validated via the prototype tool ELSE [24] (see also Deliverable [5]), finding good reductions.

In [1], we demonstrate on the job-shop-scheduling problem, that a state based approach together with partial order reduction methods can result in efficient algorithms for scheduling, which we moreover link to the non-state-based approach using disjunctive graphs. The aim of the paper is to contribute to the understanding of the relation between a state based and a constraint based approach to scheduling.

Finally, we studied the use of partial order reduction techniques in the probabilistic setting of MDPs [6, 9]. These contributions are discussed in Deliverable [4].

Symmetry Reduction

In [12], we describe a prototype extension of the tool Uppaal with symmetry reduction. The symmetric data type scalarset was added to Uppaal's system description language to support the easy static detection of symmetries. Our prototype tool uses state swaps, described and proven sound in [11], to reduce the space and memory consumption of Uppaal.

Scientifically, the results of our work on symmetry reduction are not revolutionary: we have added symmetry reduction as used within Murphi, a well-established technique to combat the state space explosion problem, to the real-time model checking tool Uppaal. For researchers familiar with model checking it will come as no surprise that this combination can be made and indeed leads to a significant gain in performance. Still, the effort required to actually add symmetry reduction to Uppaal turned out to be substantial.

The soundness of the symmetry reduction technique that we developed for Uppaal does not follow trivially from the work done by Ip and Dill in the setting of Murphi since the description languages of Uppaal and Murphi, from which symmetries are extracted automatically, are quite different. In fact, the proof that symmetry reduction for Uppaal is sound takes up more than 20 pages in [11].

In the actual implementation of symmetry reduction, it is very important how the representatives of a symmetry class are computed. Computation of canonical representatives minimizes usage of memory but can be very expensive in terms of time. We identified and implemented a number of algorithms to compute representatives in a setting of timed automata which all turn out to be very effective.

Many timed systems exhibit symmetries that can be exploited by our methods. For all examples that we experimented with (both academic toy examples and industrial cases), we obtained a drastic reduction of both computation time and memory usage, exponential in the size of the scalar sets used. For quantitative results we refer to [12] and to Deliverable 2.5.a [3].

Verification of Logic Controllers given as SFC

The research described in the previous two sections aims at exploiting the structure in an arbitrary timed automaton model. A line of research pursued at Uni DO targets the verification of logic controllers given as Sequential Function Charts (SFC) and the exploitation of the inherent structure of (timed) automata models generated from these SFC.

Programmable Logic Controllers (PLC) are widespread in the manufacturing and processing industries to realize sequential procedures and to avoid safety-critical states. For the specification and the implementation of PLC programs, the graphical and hierarchical language Sequential Function Charts is increasingly used in industry. To investigate the correctness of SFC programs with respect to a given set of requirements, Uni DO advocates the use of formal verification.

In [8], the semantics of SFC is studied. Although syntactically specified in the IEC 6113-3 standard, SFC lack an unambiguous, complete semantic description. The authors point out a number of problems and explain how these lead to different interpretations in commercial programming environments. To remedy this situation, a parametrized formal semantics for SFC is introduced including many high-level programming features such as parallelism, hierarchy, actions, activity manipulations, time, clocks, and timed actions. In [7, 18], two different approaches are presented to convert SFC programs automatically into automata models that are amenable to model checking. While the first approach translates untimed SFC into the input language of the tool Cadence SMV, the second converts timed SFC into timed automata using a procedure based on graph grammars. The timed automata can then be composed with a timed automaton modelling the plant behaviour, and model checked with the tool Uppaal. The translation of [7, 18] takes the cyclic operation mode of the PLC hardware into account. Since the explicit representation of the cyclic mode of PLC can lead to complex TA models, [21] investigates to which extent the cyclic mode can be omitted, to obtain simplified models for which the verification effort is considerably smaller. In [10], finally, an SFC is transformed into a timed automaton, the latter is composed with a hybrid automaton modeling the plant, and safety properties of the composition are verified using model checking. To deal with the state space explosion, the approach of counterexample guided abstraction (see Deliverable 2.1.1 [2]) is used.

In [7, 8, 18, 21, 10] the proposed approach is applied to some simple examples including an evaporation system and a batch laboratory plant in which two products are simultaneously produced from three raw materials. When applying the approach, an important issue is to employ a plant model of sufficient accuracy. At least for chemical processing systems, the conclusion is that finite state models (e.g. defined with Cadence SMV) are often not sufficient to verify the exclusion of safety-relevant plant states. The use of TA models is appropriate if the transition times between certain events can be estimated (or measured) accurately and conservatively. If this is not possible one can start from a hybrid dynamical model and try to derive TA models algorithmically.

Analysis of Hybrid Systems

The main thrust of Ametist is to advance the state-of-the-art in timed automata verification. However, several of the partners also have a strong interest in the closely related area of hybrid systems modeling and analysis (Uni DO, VERIMAG, UT, KUN). Clearly, there is a strong and fruitful interaction between the two areas. Below we report on three pieces of work in which specific subclasses of hybrid systems are proposed which have some additional structure to make analysis possible.

In [14, 15], a stability analysis approach for a class of hybrid automata is presented. It is assumed that the dynamics in each location of the hybrid automaton is linear and asymptotically stable, and that the guards on the transitions are hyperplanes in the state space. For each pair of ingoing and outgoing transitions in a location a conservative estimate is made of the gain via a Lyapunov function for the dynamics in that location. It is shown how the choice of the Lyapunov function can be optimized to obtain the best possible estimate. The calculated conservative gains are used in defining a so-called gain automaton that forms the basis of an algorithmic criterion for the stability of the hybrid automaton.

In [13], we investigate a new technique to determine whether an open continuous system behaves correctly for all admissible input signals. This technique is based on a discretization of the set of possible input signals, and on storing neighborhoods of points reachable by trajectories induced by those signals. Alternatively, this technique, inspired by automata theory, can be seen as an attempt to make simulation a more systematic activity by finding the smallest set of input signals such that the behaviors they induce "cover" the whole reachable state space.

Design techniques for controlled hybrid systems often have to account for the interaction of continuous and discrete degrees of freedom. The presence of the latter leads to a drastic increase of complexity with the problem size such that the applicability of many methods proposed recently was only demonstrated for relatively small systems. In [22], the limits of applicability for MILP-based techniques are investigated, and those parts of the problem that contribute most to complexity are isolated. The investigation is carried out exemplarily for a well-known approach to optimal control of hybrid systems where the control task is transformed into a mixed-integer programming problem. The influence of different indicators of the problem size on the computational effort is investigated theoretically and empirically for a scalable example. The results reveal which parameters are most critical for improving the practical solvability. Based on these results, an alternative approach was developed which combines graph search techniques with the principles of optimal control. The main idea is to embed nonlinear programming and hybrid system simulation into a graph search algorithm that fixes the discrete degrees of freedom. In order to obtain a good coverage of the hybrid search space, the method employs the notion of adjacency of intermediate hybrid state, i.e. only locally optimal trajectories are selected from sets of almost identical evolutions. The adjacency criteria can either be used as a search heuristics or, if a near-optimal solution is sufficient, to prune the search graph [19, 20].

Future Work

In all four areas discussed above, we see a lot of potential. Much research remains to be done but clearly the ideas that we developed help to further advance the state-of-the-art in verification of timed and hybrid systems. Especially in the field of symmetry reduction there is a lot of "low hanging fruit".

Within Ametist we have not invested in extensions of timed automata modeling and verification tools with a hierarchical notion of state as in statecharts, even though from a user perspective this would have been very interesting. Consequently, we also did not address hierarchy in Task 2.1. (In the work on SFC we did consider hierarchy but this was flattened in the translation to timed automata.)

Compositional backward reachability (CBR) analysis is a proof technique which has proven to be extremely useful for the practical verification of embedded software. A group of Danish researchers (involving 4 members of the AAU team) demonstrated in 2000 that they can exhaustively verify even the largest industrial applications — comprising more than 1,000 finite state machines — in a few minutes on a standard PC. In the Ametist Technical Annex CBR was mentioned as one of the research topics for Task 2.1. We tried to lift CBR to the setting of timed automata, but this turned out to be difficult. The reason for this is that time is a global phenomenon, and thus it becomes complicated to reason about the behaviour of one component without also including the other components. A second problem is that CBR works extremely well when combined with an ROBDD representation, and an equivalent representation for TA is not yet well-established. Finally, doing enumerative TA exploration backwards limits what one can do with integer variables in the model.

Nevertheless, we still consider these questions as interesting research topics for the future.

References

[1] Y. Abdeddaïm and P. Niebert. On the use of partial order methods in scheduling. In *Ninth International Conference on Project Management and Scheduling (PMS 04)*, 2004. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/pms04.pdf`. to be published as online abstract.

[2] AMETIST. Analysis and tools: Abstraction and compositionality, May 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DELIVERABLES/del2.1.1.pdf`. Deliverable 2.1.1 from the IST project AMETIST.

[3] AMETIST. Analysis and tools: Tools and tool interaction, May 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DELIVERABLES/del2.5.a.ps`. Deliverable 2.5.a from the IST project AMETIST.

[4] AMETIST. Analysis and tools: Stochastic analysis, 2005. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DELIVERABLES/del2.4.b.pdf`. Deliverable 2.4.b from the IST project AMETIST.

[5] AMETIST. Analysis and tools: Tools and tool interaction, 2005. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DELIVERABLES/del2.5.b.ps`. Deliverable 2.5.b from the IST project AMETIST.

[6] C. Baier, P.R. D'Argenio, and M. Größer. Partial order reducction for probabilistic branching time. In *3rd Workshop of Quantitative Aspects of Programming Languages,* QAPL'05, 2005.

[7] N. Bauer, S. Engell, R. Huuck, S. Lohmann, B. Lukoschus, M. Remelhe, and O. Stursberg. Verification of plc programs given as sequential function charts. In *Integration of Software Specification Techniques for Applications in Engineering*, volume 3147 of *LNCS*, pages 517–540. Springer, 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/BEH+04.pdf`.

[8] N. Bauer, R. Huuck, B. Lukoschus, and S. Engell. A unifying semantics for sequential function charts. In *Integration of Software Specification Techniques for Applications in Engineering*, volume 3147 of *LNCS*, pages 400–418. Springer, 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/BHLE04.pdf`.

[9] P.R. D'Argenio and P. Niebert. Partial order reduction for concurrent probabilistic programs. In B.R. Haverkort et al., editor, *Proc. of the 1st International Conference on Quantitative Evaluation of Systems,* QEST 2004, pages 240–249. IEEE Computer Society Press, 2004.

[10] S. Engell, S. Lohmann, and O. Stursberg. Verification of embedded supervisory controllers considering hybrid dynamics. *Int. Journal of Software Engineering and Knowledge Engineering*, 15(2):307–312, 2005. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/ELS05.pdf`.

[11] M. Hendriks. Enhancing Uppaal by exploiting symmetry. Report NIII-R0208, Nijmegen Institute for Computing and Information Sciences, University of Nijmegen, October 2002. Available from World Wide Web: `http://www.cs.kun.nl/research/reports/info/NIII-R0208.html`.

[12] M. Hendriks, G. Behrmann, K.G. Larsen, P. Niebert, and F.W. Vaandrager. Adding symmetry reduction to Uppaal. In *Proceedings First International Workshop on Formal Modeling and Analysis of Timed Systems (FORMATS 2003),* September 6-7 2003, Marseille, France, volume 2791 of *LNCS*. Springer Verlag, 2004. Available from World Wide Web: `http://www.cs.kun.nl/ita/publications/papers/fvaan/symmetry.html`. Full version available as Technical Report NIII-R0407, NIII, University of Nijmegen, February 2004.

[13] J. Kapinski, O. Maler, O. Stursberg, and B. H. Krogh. On systematic simulation of open continuous systems. In *Hybrid Systems: Computation and Control, LNCS 2623*, pages 283–297. Springer, 2003. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/hscc03_simu.ps`.

[14] R. Langerak, J.W. Polderman, and T. Krilavičius. Stability analysis for hybrid automata using conservative gains. In *In proc. IFAC Conference on Analysis and Design of Hybrid Systems (ADHS03),* St. Malo, France, 2003. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/UTPublications/LangerakPoldermanKrilavicius_Stability.ps`.

[15] R. Langerak, J.W. Polderman, and T. Krilavičius. Stability analysis for hybrid automata using optimal lyapunov functions. In *Proc. International Conference on Dynamical Systems Modeling and Stability Investigation,* May 27-30, 2003, Kyiv, Ukraine, 2003. 1 page abstract, to appear.

[16] Denis Lugiez, Peter Niebert, and Sarah Zennou. A partial order semantics approach to the clock explosion problem of timed automata. In Kurt Jensen and Andreas Podelski, editors, *Tools and Algorithms for the Construction and Analysis of Systems: 10th International Conference, TACAS 2004*, volume 2988 of *LNCS*, pages 296–311. Springer-Verlag, 2004. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/tacas04.pdf`.

[17] Denis Lugiez, Peter Niebert, and Sarah Zennou. A partial order semantics approach to the clock explosion problem of timed automata. In Kurt Jensen and Andreas Podelski, editors, *accepted for Theoretical Computer Science - special issue on selected papers of TACAS 2004*, volume 2988 of *LNCS*, page 40 pages. Springer-Verlag, 2005. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/specialissue.pdf`.

[18] M.P. Remelhe, S. Lohmann, O. Stursberg, S. Engell, and N. Bauer. Algorithmic verification of logic controllers given as sequential function charts. In *Proc. IEEE Conf. on Computer-Aided Control System Design*, 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/RLSEB04.pdf`.

[19] O. Stursberg. Dynamic optimization of processing systems with mixed degrees of freedom. In *Proc. 7th Int. Symposium on Dynamics and Control of Process Systems*, page ID: 164, 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/DYCOPS04.pdf`.

[20] O. Stursberg. A graph search algorithm for optimal control of hybrid systems. In *Proc. 43rd IEEE Conf. on Decision and Control*, pages 1412–1417, 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/Stu04b.pdf`.

[21] O. Stursberg, S. Lohmann, and S. Engell. Improving dependability of logic controllers by algorithmic verification. In *16th IFAC World Congress Prague*, 2005. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/SLE05.pdf`. accepted.

[22] J. Till, S. Engell, S. Panek, and O. Stursberg. Applied hybrid system optimization - an empirical investigation of complexity. *Control Engineering Practice*, 12(10):1269–1278, 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DORTMUNDPublications/TEPS04.pdf`.

[23] Sarah Zennou. *Methodes d'ordre partiel pour la verification de systemes concurrents et temps reel*. PhD thesis, Universite de Provence, Laboratoire d'Informatique Fondamentale de Marseille, 2004. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/thesezennou.pdf`.

[24] Sarah Zennou, Manuel Yguel, and Peter Niebert. *ELSE*: A new symbolic state generator for timed automata. In Kim G. Larsen and Peter Niebert, editors, *Proceedings of the 1st International Workshop on Formal Modelling and Analysis of Timed Systems, FORMATS 2003*, volume 2791 of *LNCS*, pages 263–270. Springer-Verlag, 2003. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/else_update.ps`.