Bosch Case Study – Final Report

KUN, Bosch

June 3, 2005

AMETIST DELIVERABLE 3.3.4

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/`

The original case study that was proposed by Bosch as a challenge for the AMETIST project concerned the analysis of a Car Periphery Supervision system (CPS). At the beginning of Year 2 of the AMETIST project, Bosch decided to put the further development of the CPS system to a halt. Consequently, it was decided decided that also within AMETIST the work on the CPS system would be finished after completion of a paper summarizing the results on this case study [7]. An alternative case study was proposed by Bosch concerning the development of a SRS (Supplemental Restraint System) for an airbag ECU (Electronic Control Unit) to be tackled by the consortium during the remainder of the project.

In this this document we will summarize the results obtained for both case studies.

1 The CPS Case Study

First, we briefly summarize the work carried out within the AMETIST project on the case study "Real-time service allocation for Car Periphery Supervision". Part of the text below is taken from Deliverable 3.3.3 [2]. For a more extensive overview of the results obtained for this case study, we refer to [7].

The term *Car Periphery Supervision (CPS)* refers to technology for obtaining information about the environment of a car. Such technology can serve as the basis for many driver assistance services, e.g., parking assistence, pre-crash detection, blind spot supervision, lane change assistance, etc., most of which are still under development. There are different sensor technologies available for CPS realizations, e.g., ultrasonic, radar, lidar, infrared and video. In the case study we concentrate on Short Range Radar (SRR) technology. In Deliverable 3.3.1 [9], a preliminary description of the CPS system was provided by Bosch, as well as a list of timing related problems.

1.1 Results

An initial study by partners UT and KUN made it clear that the initial description in [9] contained insufficient information to make formal analysis possible. In particular, the system requirements, assumptions about the timing behavior of the various system components, and a listing of the assumptions on the environment (e.g., number and relative speed of approaching objects) were unclear. Based on the discussions with the domain experts from Bosch, study of relevant literature, and use of MatLab to visualize the sensor visibility areas and to analyse possible trajectories of approaching objects, we were able to supplement the preliminary description of [9] with the necessary information to permit formal modelling and analysis [5, 6]. Our results show once more the importance of such an analysis: they allow us to make the right abstractions in the dynamical model of the system, and to interpret correctly the results obtained by analyzing this model. The next step we took was to actually construct a formal dynamical model using UPPAAL. In this model the environment, the sensors and the ECU are all modelled as timed automata. This is achieved by splitting the value domains of nonlinear continuous variables into a finite number of regions, and overapproximating (with the help of MatLab) the possible behavior of the continuous variables in the timed automata model. In Deliverable 3.3.2 [1], we reported on our modelling efforts.

The efforts during the second year of the project concentrated on verification. By encoding the various assumptions on the environment within our UPPAAL model, we were able to establish some key correctness properties of the system (for specific choices of the system parameters):

1. The ECU has a sufficiently accurate view on what is happening in the environment.
2. When an object reaches the pre-crash region, the ECU knows about this within a small (specific) number of time units.
3. The ECU avoids false alarms.
4. The system is deadlock free.

Using the convex-hull approximation feature of UPPAAL, the time required to verify these properties was less than a minute. A paper describing our model and results has been presented at WODES'04 [7].

1.2 Evaluation/Assessment

The original objective of the case study — as stated in the Technical Annex — was to use the timed automata framework to specify precisely the logical and timing requirements for the CPS system, and to design and verify (or alternatively, automatically synthesize) a dynamical resource allocation scheme on this basis. The original objective has been achieved, although the model of [7] is probably somewhat more abstract than what was originally intended, and our efforts were restricted to verification (rather than design and/or synthesis). Building more detailled models of the CPS system, and contributing to the design and synthesis of resource allocation schemes, made no sense given that the work on CPS had been abandoned by Bosch.

The basic approach that was taken in this case study — over-approximation of a hybrid system by a timed automaton — is well known and has been studied by several authors (see [7] for references). Nevertheless, there are not too many practical, realistic problems to which it has been applied. Our work demonstrates the practicality of the approach on the basis of a nontrivial, industrial case. The input language of UPPAAL turned out to be sufficiently expressive to allow for convenient modeling of the CPS system. The case study could have been carried out with the version of UPPAAL that existed before the start of AMETIST. So the technology and tools developed as part of AMETIST were not essential to solve this problem. However, in the model the feature of "broadcast channel" has been used, which helped to make the model more natural and compact. Use of the convex-hull approximation was essential for the successful verification with UPPAAL: without it the model becomes intractable due to the different time scales of the various components. An alternative verification approach (but is a topic for future research) is to apply a variation of the exact acceleration approach developed by Hendriks and Larsen [8] in Task 2.3 of the project.

2 The Airbag ECU Case Study

Much of the work on this case study was carried out by Marko Auerswald from Bosch, who provided an initial timed automata model of the airbag ECU with additional information about the behavior of each of the components [3]. Following the second review it was decided to limit the resources for this case study to a minimum. As a result, only Conrado Daws from KUN spent a couple of weeks on verification of the timed automata model. KUN intends to continue working on the case outside the scope of AMETIST.

2.1 Objectives

The goal of the second Bosch case study is to explore — in the course of a pilot project — the suitability of using timed automata-based modeling for increasing product quality in a cost effective way. The pilot project is attached to a current platform development project within Bosch of an airbag ECU. Within the development life-cycle, Bosch identified four potential areas to be explored by the project:

1. **Specification of real-time behavior**. During ECU architecture definition, components and their behavior have to be specified. The benefit of using timed automata as the formal description of the behavior of such components is to serve as documentation, replacing natural language specifications, providing more clarity and comprehensibility, thus simplifying the communication between development teams.

2. **Validation of the architecture specification**. The timed automata models of the components can be used to simulate system behavior for specific scenarios of the environment. A scenario-based analysis of the architecture is a common approach to validate architecture decisions.

3. **Verification of safety properties in a distributed real-time control system**. Safety critical functionality usually is implemented in a redundant way with diverse channels. These channels have a logical as well as temporal control of their own. In the example of the airbag ECU, several logical or physical plausibilization channels exist to avoid inadvertent deployment of the airbag. Nevertheless, these shall not hinder correct deployment.

   Modeling the real-time behavior of the plausibilization channels, and the relevant behavior of the environment, will allow to check whether there exist specific sequences in the environment which could lead to conditions under which plausibilization channels hinder the correct deployment of restraint devices.

4. **Validation of the implementation**. From formal specifications of the real-time behavior of components, test cases can be derived automatically. An appropriate coverage criteria based on the behavior model can be formulated. Testing the implementation with this generated test set ensures that the implementation really implements the specified behavior and thus also fulfills the verified safety properties.

2.2 Results

In the course of our work, we contributed to the first three of the stated objectives.

2.2.1 Modelling

The timed automata model of the airbag ECU provided by Bosch served as a clear and easy to understand documentation of the case study for the other partners of the project. The fact that the industrial partner came up with a good model using UPPAAL shows that the tool user-friendly graphical interface and its rather intuitive semantics makes UPPAAL accessible to people from the industry.

2.2.2 Scenario Analysis

One of the questions Bosch had was about the possibilty of using UPPAAL to analyse certain scenarios of the environment which had been identified as being able to exhibit an undesired behavior of the system. It was possible to verify with UPPAAL using a very abstract model of the ECU, that the undesired situation could emerge, but without providing a compact and valuable characterization of the type "when event A is followed by event B within between x and y time units, then the undesired behavior occurs". UPPAAL provides a concrete counterexample, i.e., a step-by-step error scenario, but leaves it to the user to extract a characterization of the "essence" of the problem. Such a characterization would be valuable not only to give a human readable interpretation of the problem, but also to assess the likelihood of such a scenario. Clearly, this is an interesting topic for future research.

2.2.3 Verification of Properties

We considered three properties that the airbag ECU model should satisfy:

- **Property 1:** it is possible to fire the airbag.
- **Property 2:** the system does not deadlock (this a property of the model rather than of the actual system itself).

- **Property 3:** when both the microcontroller and the approver enable the firing of the airbag, then the firing should not be hindered, that is, the firing stage should be unlocked.

The first property, for which it suffices to find an execution that leads to the firing of the airbag, was easily verified with UPPAAL. The two other properties require an *exhaustive exploration* of the state space of the system, which is highly memory- and time-consuming. The verification of these properties pushed UPPAAL to its limit, and it was first necessary to consider an untimed version of the model (i.e. the same model with the timing contraints removed) to get a result, and then showed that it was possible to carry out the verification on the constrained model on a machine with sufficient memory.

Moreover, by applying the convex-hull approximation, which computes an over-approximation of the set of reachable states, it was possible to verify both properties almost instantly and on machines with little memory. It is known that over-approximation preserves the validity of both invariant and absence of deadlock properties, so it can be concluded that the model satisfies properties 2 and 3. Table 1 shows the memory and time consumption of UPPAAL for the verification of each property using exhaustive analysis and over-approximation on a PC with a Pentium(R) 4 processor at 3.40GHz and 1518 MB of memory.

Table 1: Memory and time consumption

	exhaustive		over-approximation	
	mem (KB)	time (s)	mem (KB)	time (s)
P1	276	0.10	0	0.10
P2	33156	91.10	2772	0.92
P3	32984	77.82	2600	0.41

2.3 Future work

Although the initial model of the airbag ECU was succesfully verified with UPPAAL, several issues of interest to Bosch remain to be considered in order to assess the benefit of applying the timed automata modelling and analysis techniques to an industrial system.

The first issue concerns the suitability of timed automata to obtain a model which is as close as possible to the real system without recourse to artificial modelling tricks imposed by the specification language, and how clear the modelling of such a complex system would be without a higher level description language with, for instance, hierarchical concepts. Another aspect of the airbag ECU that Bosch would like to address is whether the verified properties are preserved if non-deterministic processing delays and jitter are added to the system model. A third issue, which was already mentioned above, is to develop algorithms that extract the "essence" from a counterexample.

Finally, we have not considered yet the application of formal techniques for the generation and execution of test cases generated from a timed automata specification to validate an ECU implementation. However, formal black-box timed testing techniques were recently incorporated into the real-time model-checker UPPAAL[10] and the test tool TORX[4]. Prototype implementations of the airbag ECU provide good targets on which to apply these techniques, where tests generated from the (verified) timed automata specification are executed on the ECU prototypes. As the models in the case study are not just models of software but also ASIC behaviour, such an approach could only be applied using hardware-in-the-loop tests with ECU prototypes. This means that the testing has to be carried out on site, and it remains to be seen whether it would be possible to do it in the business unit of Bosch.

KUN is willing to continue the collaboration with Bosch on some of the challenging open issues of this case study, outside the scope of AMETIST.

2.4 Evaluation/Assessment

Also the second Bosch case study could have been carried out with the version of UPPAAL that existed before the start of AMETIST, and so the technology and tools developed as part of AMETIST were not essential for solving the problem. Only very limited resources (< 1.5PM) from AMETIST have been invested in this case study. Nevertheless, we believe the case study is a very interesting and appealing. When the confidentiality conditions are no longer applicable and the model can be published, this will definitely be considered to be a nice illustration of industrial use of formal methods, and potentially as a challenging benchmark for the application of model based test generation methods. A notable feature of the model is that verification becomes extremely hard because of the timing constraints, even though there are only two clocks in the system.

References

[1] AMETIST. Bosch case study: First year report, June 2003. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DELIVERABLES/del3.3.2.pdf`. Deliverable 3.3.2 from the IST project AMETIST.

[2] AMETIST. Bosch case study: Second year report, May 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/DELIVERABLES/del3.3.3.pdf`. Deliverable 3.3.3 from the IST project AMETIST.

[3] M. Auerswald. SRS ECU Behaviour Modelling, December 2004. Confidential.

[4] H. Bohnenkamp and A. Belifante. Timed testing with TorX. In *Formal Methods*, volume 3582 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005. To appear.

[5] B. Gebremichael, H. Hermanns, T. Krilavičius, and Y.S. Usenko. Hybrid modeling of a vehicle surveillance system with real-time data processing. In *Proc. Int. Conf. on Dynamical Systems Modeling and Stability Investigation*, page 419, Kyiv, Ukraine, May 2003. Available from World Wide Web: `http://www.cs.kun.nl/ita/publications/papers/biniam/gkhu.html`.

[6] B. Gebremichael, T. Krilavičius, and Y.S. Usenko. Real-time service allocation for car periphery supervision: Requirements and environment analysis, 2003. Available from World Wide Web: `http://www.cs.kun.nl/ita/publications/papers/biniam/gku2.html`. Internal Ametist document.

[7] B. Gebremichael, T. Krilavičius, and Y.S. Usenko. A formal analysis of a car periphery supervision system. In J. Zaytoon, V. Carre-Mennetrier, X. Cao, and C. Cassandras, editors, *Seventh International Workshop on Discrete Event Systems WODES'04, Reims, France*, pages 433–439, September 2004. Available from World Wide Web: `http://www.cs.kun.nl/ita/publications/papers/biniam/gku.html`. Also available as Technical Report NIII-R0418, NIII, University of Nijmegen.

[8] M. Hendriks and K.G. Larsen. Exact acceleration of real-time model checking. *Electronic Notes in Theoretical Computer Science*, 65(6), April 2002. Available from World Wide Web: `http://www.cs.kun.nl/ita/publications/papers/martijnh/TPTS02.pdf`.

[9] S. Kowalewski and M. Rittel. Real-time service allocation for car periphery supervision, 2002. Available from World Wide Web: `http://ametist.cs.utwente.nl/RESEARCH/AMETIST_CPSPrelimDescription_1_0.pdf`. Deliverable 3.3.1 from the IST project AMETIST.

[10] Kim Larsen, Marius Mikucionis, and Brian Nielsen. Online Testing of Real-time Systems using Uppaal. In Jens Grabowski and Brian Nielsen, editors, *International workshop on Formal Approaches to Testing of Software*, Co-located with IEEE Conference on Automates Software Engineering 2004, Linz, Austria., September 2004. Available from World Wide Web: `http://www.cs.auc.dk/~bnielsen/Published/fates04.ps`.