MISCELLANEOUS CASE STUDIES — Second Year Report

UT, all CRs

April 28, 2004

AMETIST DELIVERABLE 3.5.2

Project acronym: AMETIST
Project full title: Advanced Methods for Timed Systems
Project no.: IST-2001-35304

1 Introduction

This is the Ametist second year's deliverable of Task 3.5 on Miscelleneous Case Studies. This Task has been created to collect and publish internal case studies of the CRs pertinent to the Ametist project.

Although most of the project's resources have been directed to case studies provided by the Ametist associate partners, reported under Tasks 3.1–3.4, a number of internal case studies have been carried out. They are reported in this deliverable.

A substantial part of the reported work has been funded from other sources than the Ametist project, but all of the reported work is strongly related to Ametist in terms of application area or techniques and tools used, as well as the involved researchers. It is expected that the work on these internal case studies will be relevant for the remaining of the Ametist project.

2 Internal Case Studies

2.1 On the correctness of an intrusion-tolerant group communication protocol

Intrusion-tolerance is the technique of using fault-tolerance to achieve security properties. Assuming that faults, both benign and Byzantine, are unavoidable, the main goal of Intrusion-tolerance is to preserve an acceptable, though possibly degraded, service of the overall system despite intrusions at some of its sub-parts. In [8] a correctness proof is presented of the Intrusion-tolerant Enclaves protocol via an adaptive combination of techniques, namely model-checking, theorem-proving and analytical mathematics. Murphi is used to verify authentication, then PVS to formally specify and prove proper Byzantine Agreement, Agreement Termination and Integrity, and finally robustness of the group key management module is mathematically proved.

2.2 Verification and improvement of the sliding window protocol

The well-known Sliding Window protocol caters for the reliable and efficient transmission of data over unreliable channels that can lose, reorder and duplicate messages. Despite the practical importance of the protocol and its high potential for errors, it has never been formally verified for the general setting. In order to fill this gap, a fully formal specification and verification of an improved version of the protocol is given in [4]. The protocol is specified by a timed state machine in the language of the verification system PVS. This allows a mechanical check of the proof by the interactive proof checker of PVS. The modelling is very general and includes such important features of the protocol as sending and receiving windows of arbitrary size, bounded sequence numbers and channels that may lose, reorder and duplicate messages.

2.3 Modeling and verifying a Lego car using hybrid I/O automata

In [6], the hybrid I/O automata framework of Lynch, Segala & Vaandrager is applied to model and analyze the behavior of a simple Lego car with caterpillar treads. Constraints are derived on the values of the parameters that occur in the hybrid model that guarantee that the car will always move forward along a black tape, and will never get off the tape or move backward. To simplify the analysis, the correctness proof is carried out on a transition system that abstracts from the hybrid automaton in a rather drastic manner, but still preserves the validity of the properties of interest. Even though the original model does not involve any disturbances, the general parametric analysis of the system allows to exend the results in a trivial manner to a hybrid model in which several disturbances are allowed (mistakes in measurements of lengths, drift and jitter of the hardware clock, velocity, and distance between the two caterpillar treads).

2.4 Deadlock avoindance policy for a wafer scanner

Martijn Hendriks and Frits Vaandrager (KUN) were involved in a case study proposed by end-user panel member ASML. For a new wafer scanner that is currently being designed by ASML, they showed how model checking techniques can be used to compute (i) a simple yet optimal deadlock avoidance policy (with Cadence SMV), and (ii) an infinite schedule that optimizes throughput in the absence of errors (with UPPAAL). These results were obtained within two weeks, which confirms once more that model checking techniques may help to improve the design process of realistic, industrial systems. The specific deadlock avoidance policy that was synthesized by Hendriks and Vaandrager will be part of a patent that is currently being filed by ASML. Once the patent has been filed, a paper describing this case study will be made publically available.

2.5 Analysis of the IEEE 1394 root contention protocol

The article [5] reports on the automatic verification of timed probabilistic properties of the IEEE 1394 root contention protocol combining two existing tools: the real-time model-checker Kronos and the probabilistic model-checker Prism. The system is modelled as a probabilistic timed automaton. Kronos is first used to perform a symbolic forward reachability analysis to generate the set of states that are reachable with non-zero probability from the initial state, and before the deadline expires. This information is then encoded as a Markov decision process to be analyzed with Prism. This technique is applied to compute the minimal probability of a leader being elected before a deadline, for different deadlines, and to study the influence on this minimal probability, of using a biased coin, and considering different wire lengths.

2.6 Testing conformance of real-time applications by automatic generation of observers

A methodology for testing conformance of an important class of real-time applications in an automatic way, is proposed in [3]. The class includes all applications for which a specification is available and can be translated into a network of timed automata. The method relies on the automatic generation of an observer from the specification, on one hand, and on the instrumentation of the system to be tested, on the other hand. The testing process consists in feeding the traces generated by the instrumented system to the observer, which is a testing device, used to check conformance of a trace to the specification. This approach was applied to the NASA K9Rover case study.

2.7 Synthesis of safe, QoS extendible, application specific schedulers for heterogeneous real-time systems

This paper [7] presents a new scheduler architecture, which permits adding QoS policies to the scheduling decisions. It also presents 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 analysing the task interactions and the overall system behaviour. The 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 synchronisation primitives. Finally, a prototype implementation of this scheduler architecture and related mechanisms on top of an open-source OS for embedded systems are presented.

2.8 Modelling and analysis of a leader election algorithm for mobile ad hoc networks

This article shows the correctness of an algorithm that, while very simple, exhibits the interesting features of more complicated distributed algorithms. It uses modelling and model-checking with Uppaal. The problem was posed by Leslie Lamport, and is inspired by a classic algorithm of Radia Perlman. The original algorithm constructs a spanning tree and maintains that tree by having the root periodically propagate an "I am alive" message down it. A new tree is constructed if a failure caused some node to time out before receiving the message. The simple algorithm assumes an arbitrary network of nodes. Each node can send messages to its neighbors. The network need not be connected. The leader of a connected component is defined to be the lowest-numbered node in the component. The goal of the algorithm is for each node n to learn the leader of its connected component. Correctness of this algorithm means that if no failure or repair has occurred for a sufficiently long period of time, then every node knows its leader. [2]

2.9 Notes on a Uppaal model of the Welch/Lynch clock synchronization protocol

A simplified algorithm for synchronized clock by Welch and Lynch is modelled with Uppaal in [1]. This model is too complex to let the Uppaal tool prove the essential of the model. It is henceforth reduced to a simpler one that is used to let Uppaal prove the properties on it.

References

[1] L. Aceto, G. Behrmann, J.F. Groote, and K. Larsen. Notes on a Uppaal model of the Welch/lynch Clock synchronization protocol, 2004. Available from World Wide Web: `http://www.cs.auc.dk/~kgl/AcetoBehrmannGrooteLarsen.pdf`. Unpublished note.

[2] G. Behrmann, K. Larsen, and A. Skou. Modelling and analysis of a leader election algorithm for mobile ad hoc networks, 2003. Available from World Wide Web: `http://www.cs.auc.dk/~kgl/leader.xml`. Modelling carried for on request by Leslie Lamport.

[3] S. Bensalem, M. Bozga, M. Krichen, and S. Tripakis. Testing conformance of real-time applications by automatic generation of observers. In *Runtime Verification (RV'04)*, 2004.

[4] D. Chkliaev, J. Hooman, and E. de Vink. Verification and improvement of the sliding window protocol. In *Proceedings TACAS'03*, pages 113–127. Lecture Notes in Computer Science 2619, Springer-Verlag, 2003. Available from World Wide Web: `http://www.cs.kun.nl/~hooman/SWP.html`.

[5] C. Daws, M. Kwiatkowska, and G. Norman. Automatic verification of the IEEE 1394 root contention protocol with KRONOS and PRISM. *Software Tools for Technology Transfer*, 5(2–3):221–236, 2004. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/UTPublications/daws-STTT04.ps.gz`.

[6] A. Fehnker, F.W. Vaandrager, and M. Zhang. Modeling and verifying a Lego car using hybrid I/O automata. In *Third International Conference on Quality Software (QSIC 2003), m Dallas, Texas, USA, November 6 - 7*, pages 280–289. IEEE Computer Society Press, 2003. Available from World Wide Web: `http://www.cs.kun.nl/ita/publications/papers/fvaan/LEGO.html`.

[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] M. Layouni, J. Hooman, and S. Tahar. On the correctness of an intrusion-tolerant group communication protocol. In *Proceedings 12th Conference on Correct Hardware Design and Verification Methods (CHARME 2003)*, pages 231–246. Lecture Notes in Computer Science 2860, Springer-Verlag, 2003. Available from World Wide Web: `http://www.niii.kun.nl/~hooman/CHARME03.html`.