Cybernétix Case Study – Final Report

LIF, CYR

May 30, 2006

AMETIST DELIVERABLE 3.1.4

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/`

1 Introduction

The Cybernétix case study treats coordination aspects of a smart card *personalization* machine. This machine is one element in chain of relatively compact machines for smart card productions.

The particularity of the machine HPX4000 is the hardware for transporting cards through the machine, a patented combination of a conveyor belt (which results in a linear design of the machine) with mechanical elements to lift cards to positions with programming interfaces (programming stations). This allows to perform the time consuming programming step on several cards in parallel.

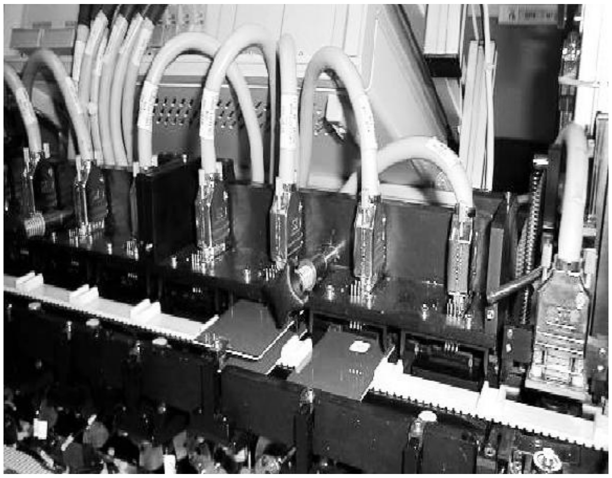

Figure 1: The conveyor belt and programming stations

The machine is designed to work on batches, i.e. it takes piles of *raw* cards and produces piles of programmed and printed cards. However, production need follow a rythm of one batch at a time. The principle design goal for this machine in a very competitive market was to build relatively small machines (fitting into normal rooms) with a high throughput. For personalization, one of the determining factors is the programming or personalization time. For an increased throughput, a parallel architecture with several programming stations is an obvious solution. Getting the cards from the input stations to the programming stations, from there to the printing stations and from there to the output, while removing cards failing the tests requires an efficient transport system. For this transport system, Cybernétix invented and patented a dedicated conveyor belt for moving a sequence of cards and a mechanism for lifting cards from this conveyor to a programming station above. In particular, the mechanism allows to move the conveyor while a card is lifted and other cards may pass on the conveyor below. The second challenge of the design is the development of a scheduling algorithm that routes the cards through the system on this given architecture in an efficient manner. Cybernétix found a particular algorithm for this baptized *SuperSingle Mode*, which is equally part of the patent for the conveyor system.

A second concern for the scheduling focuses on faulty cards. A certain fraction of the cards contains electronic defects not detected before actually programming the cards. A defective card must be replaced by a new card without modifying the order of output and at the lowest possible cost. For the SuperSingle Mode, Cybernétix invented a recovery method that actually required to modify the machine design (the length of the conveyor), which has been thoroughly tested in various failure scenarios and never failed. But the method was not strictly understood to be correct under any circumstances nor known to be optimal.

2 Activities in this case study in year one and two

The case study was presented to the AMETIST consortium by a detailed description with timing parameters [3] and a first formal model [2] was provided to the consortium for reference.

Automatic synthesis.

The consortium took up the challenge with a surprising amount of energy and several groups produced competing models for the machine and the related scheduling, of two of which are fully documented [5, 8]. The remarkable achievement of these works is that the patented "SuperSingleMode" was synthesised for small problem instances (few programming stations, few cards) by fully automatic means. However, it was quickly remarked that the fully automatic approaches do not scale up on the models, and little additional progress was made in this direction until year three.

Performance evaluation.

One of the original questions of the case study concerns the optimality of the SuperSingle Mode: Is it also optimal or, if that can not be answered, how far is it from optimal throughput.

To approach this question, in [4], an actual performance evaluation of the throughput of two modes, the "BatchMode" and the "SuperSingle Mode" was done and parametric formulae (with various delay parameters) were derived. This allowed to visualize the actual perfomance of these modes. While trivial upper bounds are easily established, an enigma of the SuperSingle Mode is its use of *holes* on the conveyor. By a sophisticated modeling technique, we managed to prove that the Syper Single mode is very close to optimal throughput, with a maximal error of 1% for typical parameter values.

While the latter evaluation was done manually, [6] proposed a modelling with timed automata using Uppaal for automatic performance evaluation: Rather than synthesizing a certain scheduler, it is possible to evaluate the throughput of a given scheduler with Uppaal efficiently.

A very interesting aspect of this work is the proposition of an alternative architecture of the Smartcard machine: Instead of adjacant programming stations, Mader proposes to have such stations only every second position. This architecture allows for a completely different, significantly simpler, yet well performing operation mode and was a welcome surprise to Cybernétix engineers.

Thus, a high degree of understanding of the HPX machine was achieved.

Simulation and Play-in play-out

In order to render the complex combinatorics of the HPX machine more accessible, an interactive simulator based on Java was developed [1] and continuously extended until year three. The simulator implements different configurations of the HPX machine, including error handling. Among other aspects, it allowed to discover certain aspects that are difficult to imagine theoretically, like a fundamental change in machine behaviour when passing from four to eight programming stations. The simulator was also very useful for debugging other models.

On the other hand [9] showed how to obtain a controller for the HPX machine with the Play-In/Play-Out approach for Live Sequence Charts (LSCs). LSCs are a formal graphical inter-object scenario-based language. In comparison to other models, LSCs models are easier to read and maintain due to its scenario based nature. The PlayEngin moreover allows to add visualisation plugins. Thus, in [9], a simplified representation of the HPX state is visible while "teaching" the machine by (test) "cases".

3 Work in the third year

The third year of the case study turned out to be difficult, as the economic trouble hitting Cybernétix group resulted in the selloff of the microelectronics branch that had developed the HPX machine. This had two consequences : On the one hand, the AMETIST consortium no longer had access to the engineers developing the machine, on the other hand, the management of the remainder of the company struggling for survival showed very little interest in pursuing the case study, which put the PhD student assigned to this task in a very difficult situation. Nevertheless, we decided to pursue the work on the basis of previous understanding.

A second problem arose when the thesis project was finally aborted: An important part of the results actually achieved in the third year go without documentation. We nevertheless try to summarize the work done in [7] and document what turned out to be the the most important discovery in the last year, i.e. a very good abstraction for the smart card machine.

In [7], we report two contributions to the case study:

- A new abstraction technique allowing to break down the state space of the HPX model to a significantly reduced set of states : basically, the abstraction is based on *passing from absolute to relative card identities* and it allows, different from previous models, the verification of unbounded production (no limit to the batch size). Nevertheless, combinatory explosion remains and the problem is exponential in the number of programming stations.
- A model of the error handling as used by Cybernétix and our verification efforts using Verimag's IF tool.

It is obvious, that the first contribution significantly improves the prospects of the second. With a model consisting of three parts, (1) the raw machine and cards, (2) the controller (a model of Cybernétix control program) and as a third part, (3) the error model, we managed to verify the error handling mode as used by Cybernétix. In simulation, this model behaved as expected and the IF-verifier gave the expected yes/no answers to verification questions depending on the error model for cases with a bounded number of defective cards that could show up nondeterministically in any order.

Unfortunately, the IF-verifier was not capable of extracting error traces, due to the structure of the μ-calculus queries (the required feature was not fully implemented at the time of our experiments).

The simulator [1] was extended by a switch to choose between absolute and relative card identities from the new abstraction. This extension thus represents an implementation of the abstraction reported above. Moreover, the error handling mode as implemented by Cybernétix was added to the simulator.

4 Conclusions

The Cybernétix case study was very inspiring to the project, in particular during the first half of the project. It exposed at the same time the strength – modeling and automatic resolution of small problem instances – and weaknesses – most notably concerning complexity issues when scaling to bigger problem sizes –of the consortium's technology. The results were considered with a lot of interest by the industrial partner.

With exception of the new abstraction reported here, one can safely state that a full comprehension of the case study was achieved after about 18 months and that, consequently, the interest of the consortium shifted to harder and unresolved challenges of other case studies. One factor that certainly played a role in this was also the restructuring of the company, so that the results of the consortium were no longer fed back to the department developing the machine.

The abstraction reported in [7] was communicated to the consortium early in the third year, but since the focus was mostly on other case studies, a reevaluation of the experimental results obtained in the first and second year was not considered. Nevertheless, it is obvious that with this abstraction, all experimental results achieved could be seriously strengthened.

Concerning the original goals of the case study, important parts were achieved already in the first half of the project. The impression that we did not have the right abstraction technique for conquering the complexity of the machine, was relativized by the new abstraction found in the last year, which falls into the class of symmetry reductions. While symmetry reduction was introduced into Uppaal, the kind of symmetry in question here would not be handled automatically.

In summary, the Cybernétix case study can be considered a success, a proof that the use of AMETIST technology can lead to new insights into a complex production problem.

References

[1] M. Agopian. A simulation tool for the SuperSingle mode, 2003. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/SuperSingleSimulator.zip`. Not a paper, a tool.

[2] S. Albert. Design/CPN model of Cybernetix Case Study. Technical report, Cybern'etix - LIF, 2002. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/cybernetix-cpn.tgz`.

[3] Sarah Albert. Cybernetix case study – informal description. Technical report, Cybern'etix - LIF, 2002. Available from World Wide Web: `http;//www.cmi.univ-mrs.fr/~niebert/docs/cyx.pdf`.

[4] Sarah Albert and Peter Niebert. Cybern'etix case study – performance analysis – optimality of the supersingle mode. Technical report, Cybern'etix -LIF, 2002. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/cybernetix-optimality.pdf`.

[5] 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`.

[6] A. Mader. Deriving schedules for the cybernetix case study, 2003. Available from World Wide Web: `http://ametist.cs.utwente.nl/Docs/INTERNAL/PUBLICATIONS/UTPublications/mader-cybernetix2003.ps`.

[7] Peter Niebert and Mathieu Agopian. Notes on the verification of the hpx error handling mode, May 2005. Available from World Wide Web: `http://www.cmi.univ-mrs.fr/~niebert/docs/errorhandling.pdf`. manuscript.

[8] Theo Ruys. Optimal Scheduling Using Branch and Bound with SPIN 4.0. In Thomas Ball and Sriram K. Rajamani, editors, *Model Checking Software – Proceedings of the 10th International SPIN Workshop (SPIN 2003)*, volume 2648 of *Lecture Notes in Computer Science*, pages 1–17, Portland, OR, USA, May 2003. Springer-Verlag, Berlin. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/UTPublications/ruys-spin2003.pdf`.

[9] 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`.