Cybernétix Case Study – First Year Report

LIF, CYR

June 7, 2003

AMETIST DELIVERABLE 3.1.2

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

Figure 1: The smart card personalization machine

1 Introduction

The Cybernétix group designs, manufactures and sells advanced automated and robot driven systems. Cybernétix offers technological solutions, which aim at increasing productivity of production lines and improving product quality in the field of the industrial automation and robotics. One of the domains of activities concerns the production of smartcards. Cybernétix is producing small machines for all steps of production of a smartcard.

For AMETIST, Cybernétix proposed as case study a smart card *personalization* machine, depicted in Figure 1. This machine realises some of the later steps in the production of smart cards. It takes cards with an already integrated chip, programs them with personalised data and does some tests, prints on them. 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 not be stopped at borders of batches.

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. A detail of this transport system is depicted in Figure 2: Two cards can be seen on the conveyor belt, while a third card, above the left card, is being held up against the electronic contacts of the programming station (actually a cable leading to it). The array like assembly of several such programming stations at adjacent positions above the conveyor is clearly visible.

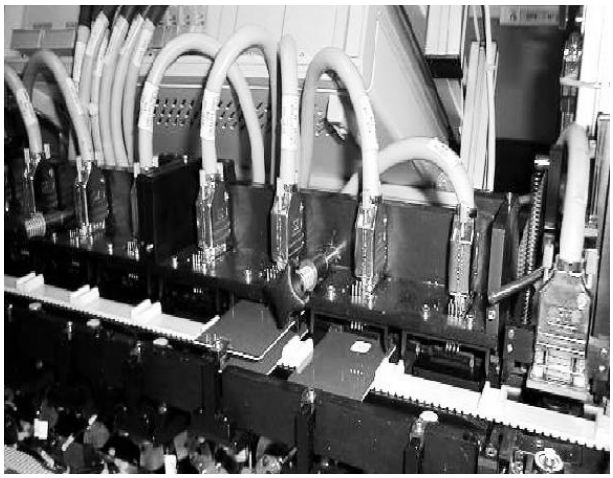

Figure 2: The conveyor belt and programming stations

The modular design of the machine allows up to 32 parallel programming stations. From the point of view of scheduling, the second challenge of the design is how to route the cards through the system on this given architecture. The following objectives are to be met:

- The throughput of the scheduled machine should be as high as possible. The throughput actually achieved by the machine may reach 6000 cards per hour for a programming time of several seconds per card.
- Cards are considered identical at the input (although some cards may be defective) and must leave the system programmed in a predefined order (e.g. with a serial number).

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 concerns *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: Only non defective cards should be discarded and the loss of throughput due to faulty cards should be as low as possible.

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 is not strictly understood to be correct under any circumstances nor known to be optimal.

Challenges to AMETIST

The scheduling algorithm used by Cybernétix, the SuperSingle Mode, is considered as far from obvious. It was found playing with a simulation model of the machine, after a much simpler, originally anticipated algorithm, the BatchMode, turned out to have a disappointing throughput.

Since the methods explored in AMETIST are oriented towards scheduling and control synthesis, the choice of the case study is first of all motivated by the question whether a method such as the SuperSingle Mode can be found automatically, or whether maybe an even better method exists. The interest of this possibility is less oriented towards the current machine but towards future machines using similar architectures.

Also of high interest is the question of faulty cards. A challenge to the consortium is whether error recovery can be synthesized or at least verified and whether the error recovery can be optimized.

A very different line of research of interest is to consider modifications of the architecture of the machine. As it seems, the SuperSingle Mode is tightly connected to the particular architecture chosen. It is conceivable however, that modified architectures may provide advantages and allow different scheduling methods.

2 Individual contributions

In the following, we summarize the work done on this case study in the first year.

Model and description

Deliverable 3.1.1 [2] gives a precise informal description of the case study and the problem. It was considered precise enough by the consortium so as to make a common unique formal model, as originally suggested in the proposal, unnecessary. Instead, several formal models were written for different formalisms, as outlined in the following.

Models and automatic analysis

SMV model and analysis

The first analysis oriented model of the conveyor and the programming stations was written in SMV [4]. To make the problem accessible to automatic analysis with BDDs, it was somewhat simplified and is more synchronous than the actual system.

The models descriptions in SMV are remarkably simple, two to three pages for scalable parametric models. This is important because it makes modelling errors less likely. The conclusion is that SMV has the right expressive means for this kind of problems.

Two experiments are conducted: (a) an optimal control mode for the machine without faulty cards is found using symbolic breadth first search. It turns out that the found mode is exactly the SuperSingle Mode! (b) limited error treatment is considered in terms of self stabilization: From correct functioning, as soon as an error occurs, a recovery strategy is searched for that goes back to the correct functioning as quickly as possible, while respecting some safety constraints.

The SuperSingle Mode was found on models in the order of magnitude of eight cards and four programming stations. Despite the often very successful approach of BDDs as symbolic representation, the complexity grows too quickly to scale up.

SPIN model and branch-and-bound solution

In a different approach, [6] modeled the machine with SPIN and applies a modelling technique on the level of linear time temporal logic specifications, developed in the same article, to guide the

model checker SPIN towards an optimal solution.

The aim of this work is to explore the feasability of scheduling using a standard model checker (without internal modifications) just by using auxiliary variables and temporal logic formulae for restricting the search space.

The modeling and guiding technique is described in detail on the case study and several heuristics are compared experimentally. Although the technique is completely different from the SMV approach (fundamentally different data structures and algorithms), the size of the feasable problem instances is very similar.

Petri net model with Design/CPN

Sarah Albert of Cybernétix worked out a colored Petri net model in Design/CPN [1] (this model requires a free license of Design/CPN `http://www.daimi.au.dk/designCPN/`). The idea of that model is to represent the cards of the smartcard machine by colored tokens (carrying, e.g., the serial number) and the formalism seems adapted from the point of view of modelling.

However, from the point of view of provided analysis methods and efficiency, Design/CPN turned out to be an insufficient tool for the problem and it was decided to finish this line of work when Sarah Albert left Cybernétix.

Analytic approaches

Where the fully automatic approaches reach their limits, human insight is required. Of the following to contributions, one is fully manually, while the other uses the automatic tool support only to analyse the throughput of predefined methods. Both approaches have in common to manually derive theoretical upper bounds of the throughput, but the modelling assumptions and accordingly the results differ slightly.

Manual performance evaluation of the SuperSingle Mode

One of the original questions of the case study concerns the optimality of the SuperSingle Mode: Is it really optimal or how far is it from upper bounds of the throughput. To approach this question, in [3], 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. In fact, the output of this mode is not fully continous, but from time to time, a little slowdown corresponding to a hole on the conveyor is unavoidable. A challenge solved in this work is thus to derive an upper bound where the time corresponding to this hole appears. This was achieved using an abstract *measure*, which distributes this non-continuous slowdown continuously over the flow.

The result shows that the throughput of the SuperSingle Mode is very close to the theoretical optimum. Essentially, one parameter in its performance remains unexplained, for typical parameter values in the order of magnitude of 1%.

Modelling and analysis with cost optimizing UPPAAL

In [5], modelling with timed automata using UPPAAL is proposed. Since initial experiences exposed similar results with respect to scalability as the other approaches, Mader then set up a framework for evaluating automatically the throughput of different strategies – which are added to the model rather than synthesized.

The evaluation of a given strategy (mode) is performed using a cost optimizing version of UPPAAL. On the other hand, a manual analysis similar to [3] is used to establish upper bounds and the

results computed by UPPAAL are compared with the measured results.

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 other position. This architecture allows for a completely different, significantly simpler, yet well performing operation mode and was received with surprise by Cybernétix engineers.

Moreover, this work is the first to integrate other aspects of the machine into the performance analysis, a printing and "flip-over" station that is downstream on the same conveyor. This part of the machine superimposes additional constraints on the throughput that are not independent of the programming stations and very difficult to analyse manually. This shows that automatic analysis using a tool like UPPAAL for this kind of task – more modest than control synthesis – is feasable and potentially of use for everyday engineering.

Alternative Modelling Approaches

In [7], an alternative modelling approach using scenarios is presented. Life Sequence Charts (LSCs) are used to do partial specifications of what should happen with the cards in the machine, by and by the specifications can be refined to obtain a controller for the machine.

LSCs constitute a visual formalism for specifying sequences of events and message passing activity between objects. The language allows to specify scenarios of behavior that cut across object boundaries and exhibit a variety of modalities, such as scenarios that can occur, ones that must occur, ones that may not occur (called anti-scenarios), ones that must follow others, ones that overlap with others, and more. Together with LSCs comes an intuitive development process called Play-In/Play-Out, with which one can conveniently capture inter-object scenario-based behavior, execute it, and simulate the modeled system in full.

The contribution represents a case study, application of LSCs and Play-In/Play-Out to the smart card machine. On the one hand, it will help to evaluate the use of this development approach on a practical example, on the other hand, a continuation of this work is aimed to eventually obtain full controllers from partial specifications with the help of underlying analysis tools.

3 Conclusions and outlook

The presentation of this case study at the kickoff meeting immediately triggered the interest of the consortium in this challenge, some groups even started to work on it based on slides before the description (deliverable 3.1.1) was available.

Several research groups independently modelled the case study with different formalisms and tools that actually rediscovered the SuperSingle Mode on small instances (the machine is parametric in the number of programming stations and also the batch size can be considered as a parameter). On the negative side, the overall conclusion is that current automatic methods do not scale up to the practically used parameters of the machine. It is a challenge to learn from the problems encountered here to improve these methods or find useful abstractions for the parameters of the machine. However, with respect to the discovery of algorithms, the automatic approach might still be useful to generate an optimal schedule automatically for small instances and let human thinking derive a generally applicable method from it.

Another remarkable observation is that, despite differences in the models, coherently the SuperSingle Mode was synthesized. This indicates, that this mode is rather robust and not very dependent on actual parameters.

On the other hand, more specific investigations done manually gave more insights on the SuperSingle Mode (concerning its optimality) and a first step in the direction of alternative architectures was actually done with surprising results.

At the meeting May 5-7/2003, it was observed that the focus for this case study should now turn to error recovery, so far almost neglected from investigations.

Concluding, this case study has seen a very inspiring and productive first project year and will hopefully continue to provide a similar inspiration and progress in the following years.

References

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

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

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

[4] 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`. Submitted.

[5] A. Mader. Deriving schedules for the cybernetix case study, 2003. Available from World Wide Web: `http://ametist.cs.utwente.nl/INTERNAL/PUBLICATIONS/UTPublications/mader-axxom2003.ps.gz`.

[6] 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`.

[7] 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`.