Control Synthesis for a Smart Card Personalization System using Symbolic Model Checking

B. Gebremichael, F.W. Vaandrager

Nijmegen Institute for Computing and Information Sciences/

NIII-R0312 May 2003

Nijmegen Institute for Computing and Information Sciences
Faculty of Science
Catholic University of Nijmegen
Toernooiveld 1
6525 ED Nijmegen
The Netherlands

Control Synthesis for a Smart Card Personalization System using Symbolic Model Checking*

Biniam Gebremichael and Frits Vaandrager

Nijmeegs Instituut voor Informatica en Informatiekunde
University of Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
[biniam,fvaan]@cs.kun.nl

Abstract. Using the Berkeley SMV symbolic model checker we synthesize, under certain error assumptions, a controller for the smart card personalization system, a case study that has been proposed by Cybernetix Recherche in the context of the EU IST project AMETIST. The controller that we synthesize, and of which we prove optimality, has been previously patented. Due to the large number of states (which is beyond 10^{11}), this control synthesis problem appears to be out of the scope of existing tools for controller synthesis, which typically use some form of explicit state enumeration. Our result provides new evidence that model checkers can be useful to tackle industrial sized problems in the area of scheduling and control synthesis.

1 Introduction

Background

Model checking involves analyzing a given model of a system and verifying that this model satisfies some desired properties. System models are typically described as finite transition systems, while properties are described in terms of temporal logic. Once the definition of the system, $\mathbf{S}$, and its property, ψ, are fixed, the model checking problem is easily described as $\mathbf{S} \models \psi$? (does $\mathbf{S}$ satisfy ψ?). Thanks to the symbolic representation of transition systems, state-of-the-art model checking tools are now capable of solving such problems for models with more than 10^{20} states [BC90].

Control synthesis, on the contrary, does not assume the existence of a model of the full system. Instead, it considers the uncontrolled plant and tries to synthesize a controller by finding a possible instance of a

* This work was supported by the European Community Project IST-2001-35304 AMETIST, http://ametist.cs.utwente.nl.

model that satisfies a desired property. Control synthesis for Discrete Event Systems (DES) has been extensively studied over the past two to three decades, and a well-established theory has been developed by Ramadge and Wonham [RW89]. The Ramadge and Wonham framework (RW) is based on the formal (regular) language generated by a finite state machine. The RW plant model P (*generator*) is obtained by describing the plant processes in terms of a formal language which is generated by a finite automaton. A *means of control* is adjoined to this *generator* by identifying the events that can be enabled or disabled by the controlling agent. The specifications S_p are described in terms of formal language generated by P. The controller is then constructed from a recognizer for the specified language given by S_p. An alternative approach is the timed transition model of Ostroff [OS89], where the specification is given as a temporal logic formula instead of a formal regular language.

Control synthesis problems for Discrete Event Systems like the Cybernetix smart card personalization system [Al02] are covered by the Ramadge and Wonham supervisory control theory. In the present paper, however, we solve the problem using a model checker, namely SMV [McM93].[1] This approach allows us to benefit from the (BDD-based) symbolic representation technique of SMV and to solve the problem which, because of its size, would be intractable otherwise. Our results demonstrate that model checkers can be useful to solve problems in the area of scheduling and control synthesis.

Outline

Using SMV we synthesize a controller for a smart card personalization system, which has previously been patented by Cybernetix Recherche. We also show that this controller or scheduler, known as the "super single mode" [Al02] is optimal in the absence of errors. Finally, we synthesize a defective cards treatment that stabilizes the system to the super single mode.

The paper is structured as follows: Section 2 provides a formal definition of the uncontrolled plant of the smart card personalization system, and defines the correctness and optimality criteria. Section 3 explains the super single mode, and how it was generated using SMV. Section 4 deals with systems with faulty cards. We list the errors that may occur during the operations of the machine, show how to deal with such errors, and give

[1] We use the version of SMV developed at Cadence Berkeley Laboratories, see `http://www-cad.eecs.berkeley.edu/~kenmcmil/smv/`.

an overview of the synthesized error treatment methods. We conclude the paper by pointing out some observations and directions for future work in Section 5. The complete SMV code for the super single mode and defective card treatment is provided in Appendices A and B, respectively. An electronic copy of this code and also of the trace simulator that we developed to visualize the schedules are available via the URL

`http://www.cs.kun.nl/ita/publications/papers/biniam/cyber`.

Related Work

The Ramadge and Wonham framework has been implemented by several research groups and industries. One of the tools developed by Wonham and his research team is CTCT (C based Toy Control Theory)[2], a tool that was basically built for research purposes only, and uses an exhaustive list to represent the model. Its capacity, as the name indicates, has never extended beyond toy examples. A new approach, Vector Discrete Event Systems, was studied in [LW93,LW94] to alleviate the shortcoming of CTCT by exploiting the structural properties of DES. Although this approach resulted in better performance, its structural analysis approach cannot be generalized [CL99].

The UMDES-LIB library [SSLST95] developed by the DES group at the University of Michigan is another implementation of a control synthesis tool, which is very similar to RW supervisory theory. UMDES-LIB is a library of C routines written for the study of discrete event systems modeled by finite-state machines (FSM). There are several routines for the manipulation of FSM, including routines that implement many of the operations of supervisory control theory, and routines that implement part of the methodology developed at University of Michigan for failure diagnosis of discrete event systems.

Bertil Brandin at Siemens, Muenchen, also developed a tool for DES control synthesis, which incorporates heuristics to deal with large systems composed of multiple FSMs [Br96]. Bruce Krogh and his group at Carnegie Mellon University developed a tool for Condition/Event Systems [SK91] which is similar to the supervisory control theory of Ramadge and Wonham. Martine Fabian and Knut Åkesson [AF99] at Chalmers University in Gothenburg, Sreenivas at the University of Illinois at Urbana Champaign [SK91] and several other researchers have also developed similar software.

[2] See `http://odin.control.toronto.edu/people/profs/wonham`.

All the above tools lack symbolic representation of state transitions, and suffer from state space explosion problems. A Binary Decision Diagram (BDD) like data structure called Integer Decision Diagram (IDD) has been used to represent sets of states symbolically. For example, Gunnarsson in [JG97] and Zhang and Wonham in [ZW01] have used IDDs in their implementation. This approach is quite promising for dealing with large systems, but it is still in the laboratory stage, and not available to the public.

Our main motivation for using SMV is thus to overcome this deficiency and benefit from symbolic representation of SMV. The smart card personalization system is quite a large system and cannot be handled with a tool that does not use symbolic representation. Our paper shows how the control synthesis can be solved using a model checker and presents new evidence that model checkers can be useful in solving problems in the area of scheduling and synthesis.

We were the first to model the smart card personalization system and to synthesize a controller for it. However, the same case study has also been addressed by other members of the AMETIST consortium. T. Krilavicius and Y. Usenko [KU03] constructed models using UPPAAL and μCRL, and used these to synthesize controllers. Whereas in our model production of cards is essentially an infinite process, Krilavicius and Usenko only consider scheduling of a finite number of cards. As a consequence, they do not synthesize the super single mode. Inspired by [KU03], T. Ruys used SPIN to synthesize a controller for the smart card personalization machine [Ru03]. Also this model only considers scheduling of a finite number of cards (the largest parameter values considered are 5 cards and 4 stations). In order to handle the state space explosion, Ruys encodes branch & bound search strategies in SPIN. In addition, he has to instruct SPIN to use a number of heuristics, which in our view are both complex (the code for the heuristics is longer than the code of our entire model!) and debatable (Ruys assumes that cards cannot overtake each other; in the real machine this is possible with the help of the personalization stations). A. Mader in [Ma03] applied decomposition and mixed strategies to model and synthesis a controller for the extended smart card personalization machine that include printers and flippers. G. Weiss employed Life Sequence Charts (LSC) to synthesize a scheduler with smart play-in/play-out approach [We03]. None of the mentioned approaches deals with error handling.