Supremica – An integrated environment for verification, synthesis and simulation of discrete event systems

Knut Åkesson[†], Martin Fabian, Hugo Flordal
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Göteborg, Sweden

Robi Malik
Department of Computer Science
University of Waikato
Hamilton, New Zealand

***Abstract*— An integrated environment, Supremica, for verification, synthesis and simulation of discrete event systems is presented. The basic model in Supremica is finite automata where the transitions have an associated event together with a guard condition and an action function that updates automata variables. Supremica uses two main approaches to handle large state-spaces. The first approach exploits modularity in order to divide the original problem into many smaller problems that together solve the original problem. The second approach uses an efficient data structure, a binary decision diagram, to symbolically represent the reachable states. Models in Supremica may be simulated in the environment. It is also possible to generate code that implements the behavior of the model using both the IEC 61131 and the IEC 61499 standard.**

I. INTRODUCTION

The supervisory control theory (SCT) [1], [2], [3] is a framework for verification and synthesis of discrete event supervisors. Since the theory is applicable to any system that may be modeled as discrete event system there are a large number of potential applications. However, user friendly tools that are able to solve large problems are critical in order for the theory to be accepted in industry. Supremica [4], [5], [6] is an attempt to build an integrated development environment that is able to solve large scale supervisor verification and synthesis problems. Previous releases of Supremica used finite automata without hierarchy and variables. Recent work has extended Supremica to include a more high-level modeling language that includes variables, guards, actions and hierarchy. A new user interface has been developed that is a complete integrated development environment (IDE) since it contains a graphical automata editor as well as an easy to use interface to analyze, synthesize and simulate discrete event supervisors. Supremica with the IDE, shown in Fig. 1. The editor is an extended version of, Waters, an automata editor developed at University of Waikato, New Zealand. Supremica is constantly evolving but the latest release can always be downloaded, free for education and research, from [4].

[†]Corresponding author: `knut@chalmers.se`

Beside the authors of this paper other people have also contributed to Supremica. From Chalmers the following people are the main contributors: Arash Vahidi (BDD based algorithms), Markus Sköldstam and Martin Byröd (extending the editor to handle variables, actions and guards), Goran Čengić (Fuber, a IEC-61499 runtime), Avenir Kobetski (optimization algorithms). Gian Perrone and Simon Ware at University of Waikato developed the original Waters editor.

II. SUPREMICA

To make it easier to develop large models an extended type of automata has been introduced in Supremica. This new automata type is called Extended Finite Automaton (EFA) and is an augmentation of the regular automaton with guard and action formulas associated to the transitions. State hierarchy is introduced by allowing the user to group states together. There is also support for parameterized automata that facilitate the development of models with a large number of automata with similar structure. An EFA, as defined in Supremica, is an augmentation of the regular automaton with guard and action formulas associated to the transitions. A transition in the EFA is enabled when its guard formula is true. Moreover, when a transition in an EFA is taken, updating actions of a set of variables may follow. Before any analysis on the EFA is done they are translated to standard finite automata without hierarchy, variables, guards and actions. Hence, the standard supervisory control algorithms may be used to solve the verification and synthesis problems even.

Analysis: Supremica implements monolithic and modular verification and synthesis algorithms for solving non-blocking, controllability, and combined non-blocking and controllability problems. The modular algorithms are described in [7], [8], [9], [10]. The algorithms based on binary decision diagrams are presented in [11], [12]. Supremica also has rudimentary support for associating a cost with states. Based on this cost it is possible to generate a supervisor with minimal total cost. Applications and algorithms is presented in [13], [14], [15].

Analysis Code generation and simulation: When a set of supervisors has been synthesized it is desirable to be able to generate code that implements these supervisors. Supremica can generate code in a number of formats including IEC 61131 – Instruction List, Sequential Function Charts, Structured Text; ANSI C and Java. It is also possible to generate IEC 61499 code from Supremica [16]. For some applications it is also possible to verify the behavior of existing IEC 61499 code. To make this tight integration possible a new IEC 61499 runtime environment, called Fuber, has been developed [17], [18].

Applications: Supremica has been used both a Chalmers and other universities to communicate the main ideas of supervisory control to students. Supremica has also been used in a number of applications including handling the

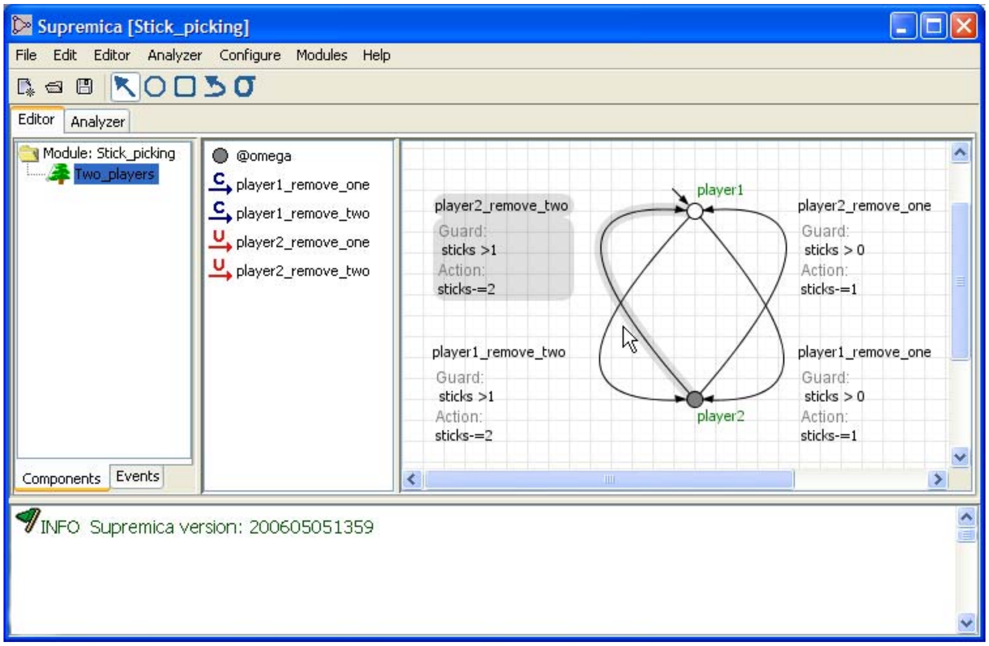

Fig. 1. Screenshot of Supremica. An extended finite automaton is shown that contains an integer variable named `sticks` and four events `player_1_remove_one`, `player_1_remove_two`, `player_2_remove_one`, `player_2_remove_two`. Each transition has an associated event together with a guard condition and an action function that update the variable.

resource allocations in a commercial chemical batch control system [19], multiple industrial robot coordination [20], human-computer supervision [21] and for manufacturing systems [22], [23].

References

[1] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. of IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[2] W.M.Wonham, "Supervisory control of discrete event systems," Department of Electrical and Computer Engineering, University of Toronto, Tech. Rep., 2005.

[3] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, Sept. 1999.

[4] "Supremica." [Online]. Available: http://www.supremica.org

[5] K. Åkesson, M. Fabian, H. Flordal, and A. Vahidi, "Supremica—a tool for verification and synthesis of discrete event supervisors," in *Proc. of the 11th Mediterranean Conference on Control and Automation*, Rhodos, Greece, 2003.

[6] K. Åkesson, "Methods and tools in supervisory control theory: Operator aspects, computation efficiency and applications," Ph.D. dissertation, Chalmers University of Technology, Göteborg, Sweden, 2002.

[7] K. Åkesson, H. Flordal, and M. Fabian, "Exploiting modularity for synthesis and verification of supervisors," in *Proc. of the 15th Triennial World Congress of the International Federation of Automatic Control*, Barcelona, Spain, July 2002.

[8] H. Flordal, M. Fabian, and K. Åkesson, "Heuristics for verification and synthesis of mutually nonblocking discrete event systems," Department of Signals and Systems, Chalmers University of Technology, Tech. Rep. R012/2004, 2004.

[9] H. Flordal and R. Malik, "Modular nonblocking verification using conflict equivalence," in *Proc. 8th Workshop on Discrete Event Systems (WODES'06)*, Ann Arbor, Michigan, USA, 2006.

[10] ——, "Supervision equivalence," in *Proc. 8th Workshop on Discrete Event Systems (WODES'06)*, Ann Arbor, Michigan, USA, 2006.

[11] A. Vahidi, "Efficient analysis of discrete event systems," Ph.D. dissertation, Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, 2004.

[12] M. Byröd, B. Lennartson, A. Vahidi, and K. Åkesson, "Efficient reachability analysis on modular discrete-event systems using binary decision diagrams," in *Proc. 8th Workshop on Discrete Event Systems (WODES'06)*, Ann Arbor, Michigan, USA, 2006.

[13] H. Flordal, D. Spensieri, K. Åkesson, and M. Fabian, "Supervision of multiple industrial robots—optimal and collision free work cycles," in *Proceedings of the IEEE Conference on Control Applications*, Taipei, Taiwan, sep 2004.

[14] A. Kobetski and M. Fabian, "Scheduling of discrete event systems using mixed integer linear programming," in *Proc. 8th Workshop on Discrete Event Systems (WODES'06)*, Ann Arbor, Michigan, USA, 2006.

[15] A. Kobetski, D. Spensieri, and M. Fabian, "Scheduling algorithms for optimal robot cell coordination – a comparison," submitted to the 2006 IEEE Conference on Automation Science and Engineering, Shanghai, China.

[16] G. Cengic, K. Åkesson, B. Lennartson, C. Yuan, and P. Ferreira, "Implementation of full synchronous composition using IEC 61499 function blocks," in *Proc. of 2005 IEEE Conference on Automation Science and Engineering*, Edmonton, Canada, 2005, pp. 267–272.

[17] "Fuber – IEC 61499 Function Block Execution Runtime." [Online]. Available: http://sourceforge.net/projects/fuber

[18] G. Čengić, O. Ljungkrantz, and K. Åkesson, "Formal modeling of function block applications running in IEC 61499 execution runtime," submitted to the 2006 IEEE Conference on Emerging Technologies and Factory Automation , Prague, Czech Republic.

[19] K. Åkesson and M. Fabian, "Implementing supervisory control for chemical batch processes," in *Proc. of the 1999 IEEE Int. Conf. on Control Applications*, Hawai'i, USA, 1999, pp. 1272–1277.

[20] H. Flordal, M. Fabian, and K. Åkesson, "Automatic implementation and verification of coordinating PLC-code for robot cells," in *Proceedings of the 11th IFAC Symposium of Information Control Problems in Manufacturing*, Salvador, Brazil, apr 2004.

[21] K. Åkesson, S. Jain, and P. Ferreira, "Hybrid computer-human supervision of discrete event systems," in *Proc. of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, 2002, pp. 2321–2326.

[22] K. Danielsson, J. Richardsson, B. Lennartson, and M. Fabian, "Automatic scheduling and verification of the control function of flexible assembly cells in an information reuse environment," in *6th IEEE International Symp. on Assembly and Task Planning*, Montreal, Canada, july 2005, pp. 80–85.

[23] K. Andersson, "Hierarchical control and restart of flexible manufacturing systems," Licentiate thesis, Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, 2006.