A Hybrid Supervisory Control System for Flexible Manufacturing Workcells

R.A. Williams
Graduate

B. Benhabib
Associate Professor

K.C. Smith
Professor

Computer Integrated Manufacturing Laboratory, Department of Mechanical Engineering,
University of Toronto, 5 King's College Road, Toronto, Ontario, Canada, M5S 1A4
beno@me.utoronto.ca

Abstract

Because of their nondeterministic nature of behaviour, the supervisory control of manufacturing systems must be carried out in closed loop. This trait greatly increases the size and complexity of the Discrete-Event System (DES)-based supervisory-controllers of manufacturing systems.

In an attempt to cope with this complexity, a Hybrid Supervisory Controller (HSC) is proposed. It splits operations between its three elements: (i) a DES supervisory controller, which contains the nominal control strategy, (ii) a diagnostic system, which monitors the workcell, and (iii) an Alternate Strategy Driver (ASD), that asserts control by revising the nominal strategy of the DES supervisory controller, whenever events diverge from the states of the DES supervisory controller. This modified approach to DES control provides a more efficient controller than could reasonably be attained using solely a DES supervisory controller.

1 Introduction

The design of a supervisory controller entails the formulation of control laws, and the synthesis of supervisors. The laws specify how the supervisor is to react to the behaviour of the manufacturing system, the goal being to have some production specifications satisfied within the standing control enforcement constraints.

Petri nets [1, 2], knowledge engineering [3, 4], timed transition models, real-time temporal logic [5, 6], and controlled automata [7] have been used as supervisory controllers for manufacturing systems. This work will focus on the application and development of a DES theory based supervisor, which utilizes controlled-automata concepts.

A systematic investigation of DESs from the control theoretic perspective was initiated by Ramadge and Wonham [8]. A DES is modeled as an automata, hence it is usually described by a set of *states* joined by transitions called *events.* A DES is controlled as a generator of a formal language. The performance of this generator is regulated by stating that its generated language must belong to some specification language. This specification language defines a control strategy.

Supervisory controllers designed around Discrete-Event Systems (DES) theory have the desirable feature that their behaviour may be proved and verified correct before implementation using the tools developed within the theory. However, due to the nondeterministic nature of behaviour of a manufacturing system, its management and therefore supervisory control must be carried out in closed loop. These traits complicate and greatly increase the complexity of the supervisory-control implementation. Thus, control of even moderately complex systems can easily require an immensely large DES strategy[1].

To *some extent*, this problem of excessive states can be mitigated through modular synthesis[2], use of aggregation, decentralization, and hierarchies, and in certain instances by restricting attention to special structures [10, 11, 12]. However, these techniques are of limited use, since they draw on special characteristics of the numerous control objectives. In a complex environment with many interdependencies between objectives, these techniques will not be sufficient for reducing the number of states to a manageable number.

If it is not possible to construct for a given set of control objectives a DES supervisory controller having a manageable number of states, it might be possible to create a split approach that uses some alternate mechanism in addition to a DES supervisory controller. This second mechanism would relieve the DES supervisory controller of the need for so many states by (i) taking on the responsibility for some of the control objectives, and (ii) asserting control whenever events diverge from the (reduced number of) states of the DES supervisory controller. This paper describes such a hybrid supervisory controller that was developed.

There exists a limited number of research papers where a DES is reported to work as part of a mixed-control mechanism. Balemi's furnace controller [13] is one of them. His DES-type supervisory control has two parts: a "supervisor" and a "controller". The "supervisor" ensures that safety constraints are enforced. The "controller" steers the system towards the desired

[1]Ho [9] notes that, when solving basic control synthesis problems, although they have been shown to be of polynomial complexity in the number of states, the number of states in a practical system can be exponential in the number of constituent processes.

[2]Exploitation of modularity of specifications, and modularity of models.

1050-4729/94 $03.00

goal, which is to accomplish a sequence of tasks. In this system, a command chosen by the "controller" will always be compatible with the safety constraints, and therefore will be accepted by the "supervisor". This arrangement allows the "controller" to be bypassed, via manual override, while maintaining safety constraints. In order to limit the size and complexity of his hybrid supervisory controller, Balemi uses an incomplete "controller" that does not accept all possible responses.

Despite having been formulated independently, the distribution of tasks between the ASD and DES Supervisor in our work bears a resemblance to that of Balemi's furnace controller. In both, one unit (our DES Supervisor and his "supervisor") is responsible for maintaining safety constraints, and another (our ASD with enforcement by DES Supervisor, and his "controller") is responsible for directing operations towards some goal. The motivation for using a mixed control in both cases was to get sufficient control power without requiring too many states.

The significant difference between our work and Balemi's is that Balemi's "controller" is implemented within the context of DES theory, whereas the ASD proposed in this paper operates by heuristic means. It is possible for Balemi's "controller" to be implemented within the context of DES theory, because the environment it controls (a furnace) is much simpler than typical manufacturing workcells.

2 Problem formulation

The Hybrid Supervisory Controller (HSC) consists of three main elements, Figure 1:

1. DES Supervisor, which contains the nominal supervisory-control strategy,
2. Diagnostic system, which monitors the cell, identifies errors and initiates recoveries, and
3. Alternate-Strategy Driver (ASD), which generates alternate part routes when needed.

The diagnostic system interprets sensory data. It feeds its interpretation to the DES Supervisor and the ASD. If the ASD determines that new (alternate) part routes are needed, it derives them and submits them to the DES Supervisor. The DES Supervisor reacts to the information received from the diagnostic system, accepts the alternate routes from the ASD, and proceeds to synthesize control commands.

In this approach the alternate mechanism takes on responsibility for only some control objectives. This requires the alternate mechanism to force its bidding upon the DES supervisory controller. One way to do this is to revise part of the nominal strategy of the DES supervisory controller. Essentially, this is asking the alternate mechanism to respond to an event that is not nominally handled by the DES supervisory controller, to find a course of action (i.e., reconfiguration) which the DES supervisory controller can be made to follow, and to force the DES supervisory controller into doing it.

Awareness of which actions are available for response to a particular event is a prerequisite to any solution generation. It is apparent that:

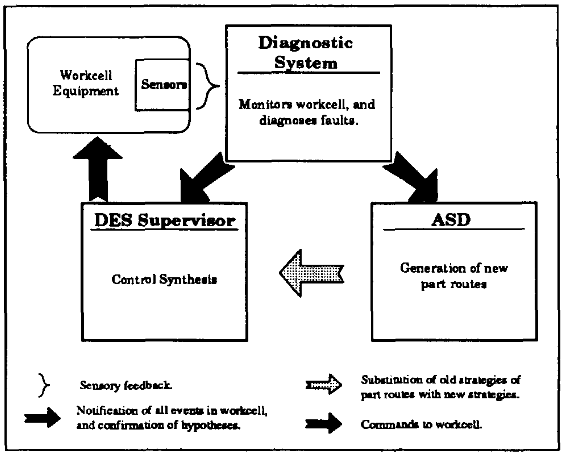

Figure 1: Overview of the interaction of the components of the HSC.

- Control enforcement constraints for maintaining the safe operation of the manufacturing system (safety contraints) are defined by the characteristics of the manufacturing system and would have to be obeyed under all conditions,
- All routes must obey safety constraints, and
- There could be a very large number of part routes possible that obey safety constraints.

In a possible solution, the safety specifications could form the kernel of the DES supervisory controller. A set of part routes would be included as well. The routes would be changed by the alternate mechanism as needed. The benefit of this division is that only one or two sets of part routes need to be kept in the DES Supervisor at any time, hence limiting the number of states.

3 Solution

3.1 DES Supervisor

For a given manufacturing system, the proposed architecture of the DES Supervisor requires:

1. Derivation of an appropriate set of plant models and specifications for maintaining safe operation of the equipment in the workcell, as well as specifications for encoding routes, Figure 2,
2. Creation of an event-classification scheme,
3. Design of the supervisor such that it does not need to be re-generated each time its specifications for part routes are modified,
4. Development of a method to check for and remove conflicts from the supervisor produced via the ASD, and
5. Design of a mechanism to correlate states in different specifications for transferring control from one specification to another.

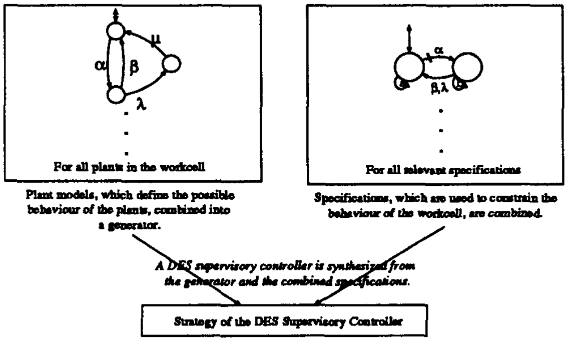

Figure 2: Constructing a DES-based supervisor.

The special requirements of the DES Supervisor required its division into two modular supervisors, Figure 3. One enforced the safety specifications. The other enforced the specifications for the part routes. The latter was an aggregation of several modular supervisors, each a specification designed as a proper supervisor for routing a single part. This allowed them to be revised without having to derive a new supervisor for each.

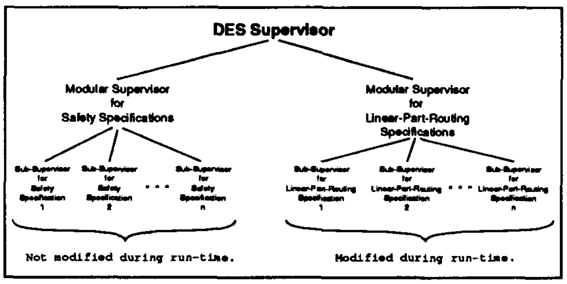

Figure 3: Design of DES Supervisor.

3.2 Diagnostic system

Several different types of diagnostic approaches were examined [14, 15]. An artificial-intelligence type model-based approach was selected and developed into a diagnostic system [15]. It utilizes several different models rather than just one universal model. Models are chosen according to the state of the equipment, what operations they are performing, and the location of parts. Models are customized to their range of application. This permits them to be significantly simplified. As well, the model-based technique automatically defines a strategy for optimal sensor usage, see [16].

3.3 Alternate Strategy Driver

The ASD required the development of:

1. A mixed on-line/off-line approach to re-routing [17], that allows quick re-routing while attempting to maintain (preset) preferred routes. The off-line strategy selects preferred routes. The on-line strategy selects routes that are as similar to the preset routes as possible,
2. A route selection mechanism that is capable of optimally choosing among:
   - The types of manufacturing operations,
   - The order of manufacturing operations,
   - The equipment used to perform the manufacturing operations, and
   - The paths to get the part to the equipment that performs the manufacturing operations.
3. A strategy for actively resolving deadlocks,
4. A special strategy for the re-routing of parts, which are being operated on, and
5. A method of producing look-up tables for generating alternate supervisory strategies that correspond to the new routes.

Deadlocks are a derivative of the unusual design of the DES Supervisor. Normally they would be removed as part of the checking for correctness and conflict removal of the supervisor, and therefore would never be allowed to occur. Since this is not possible with the DES Supervisor at hand, the ASD uses an active approach. It lets them occur, then resolves them by re-routing parts.

4 Enforcing routes: implementation

Three types of strategies are utilized in the HSC for enforcing routes:

1. The nominal strategy,
2. The alternate strategy for producing a part from its start of production, and
3. The alternate strategy for a partially manufactured part.

Normally the nominal strategy is utilized. In the instance of a plant failure or a part deadlock, it might happen that the nominal strategy may no longer be utilized, in which case alternate strategies are generated. An alternate strategy is generated for each part that has been partially manufactured, as well as for future parts that will start at the beginning of production.

To allow for the alteration of the strategy enforced on a part, two special features are incorporated into the two computer algorithms, Figure 4 for checking whether an event is permitted and Figure 5 for processing an event transition in the strategy, that perform control synthesis for the DES supervisor:

1. Flags are used to indicate which is the enforced strategy, and
2. The current state in the look-up table of the enforced strategy is determined from descriptive information about the part, rather than a specific state idenfication number.

The former feature permits the selection of the strategy to enforce. There are two flags: Flags 1 and 2. They determine which of the three columns to descend in the algorithms in Figures 4 and 5. Each column corresponds to one of the three types of strategies for enforcing routes. The latter feature simplifies transfer of control from one strategy to the next.

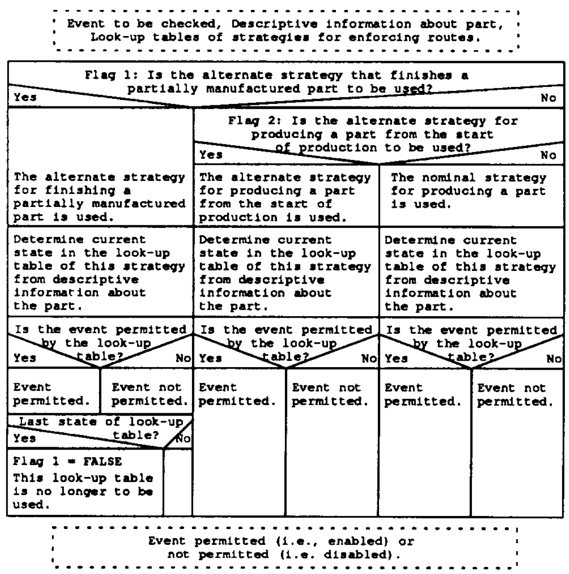

Figure 4: Enforcing routes: checking events.

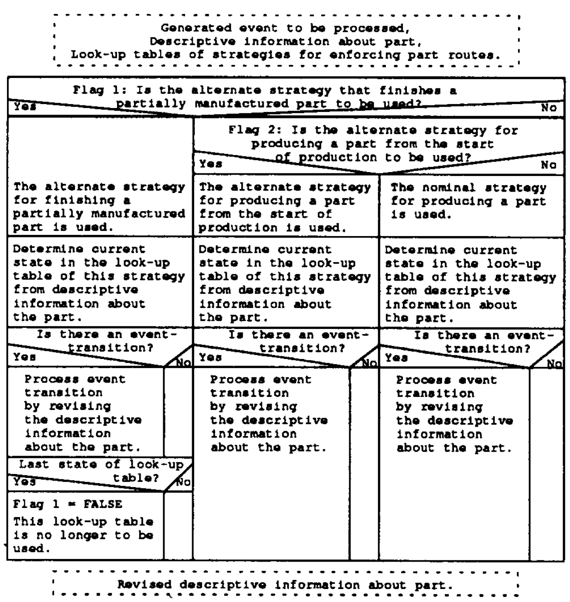

Figure 5: Enforcing routes: processing events.

5 A simulated example

The HSC was implemented in the C language and tested via simulation on a Sun Sparc Workstation. The example workcell considered consisted of two robots, two machining-centers, two buffers, a part-entry device, and a part-departure device. Two different types of parts were to be produced in the workcell. Each had its own production plan listing production goals and alternatives.

The nominal route for the production of the part was defined as follows (numbers in the list correspond to numbers on Figure 6):

1: Part arrives at Part-Entry Device.
2: Robot 1 moves the part to Machining-Center 2.
3: Machining-Center 2 performs drill operation #1.
4: Robot 1 moves the part to Machining-Center 1.
5: Machining-Center 1 performs a milling operation.
6: Robot 2 moves the part to the Part-Exit Device.
7: The part departs the workcell.

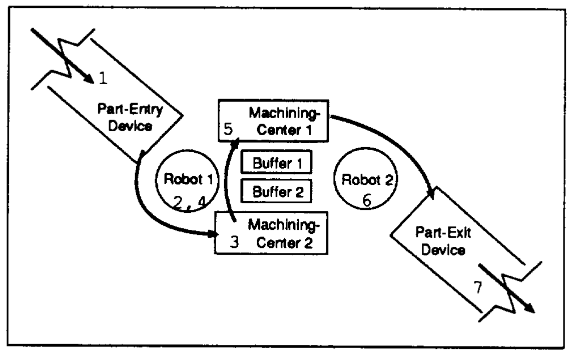

Figure 6: The nominal production route for the part.

System's response to a failure at Machining-Center 2:

1.1 Diagnostic system reads sensors.
1.2 Compares actual with expected behaviour as defined by the diagnostic models.
1.3 Detects discrepancy; (i.e., failure in Machining-Center 2).
1.4 Builds list of failed plants; list contains Machining-Center 2.
1.5 Sends a λ-type (i.e., failure) event corresponding to the failed Machining-Center 2 to the DES Supervisor and to the ASD.
2.1 ASD disables all α-type (i.e., start of operation) events of the DES Supervisor.
2.2 Generates a new route, one route is produced for the part at the start of production.
2.3 Produces look-up table for the alternate supervisory strategy.
2.4 Checks with the diagnostic system that additional plants have not failed nor been repaired.

2.5 Repeats steps 2.2 through 2.4 if the response is negative. Otherwise, transfers to the DES Supervisor the alternate look-up table, and flags for indicating the use of that look-up table.
3.1 Control is returned to the DES Supervisor, and production continues.

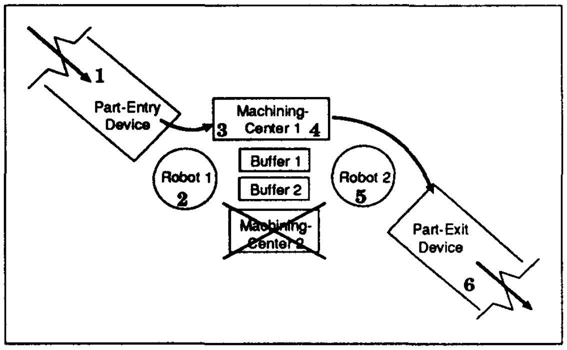

Figure 7: Alternate route for the part.

The alternate route shown in Figure 7 replaces the nominal route shown in Figure 6.

1: Part arrives at Part-Entry Device.
2: Robot 1 moves part to Machining-Center 1.
3: Machining-Center 1 performs drill operation #1.
4: Machining-Center 1 performs a milling operation.
5: Robot 2 moves the part to the Part-Exit Device.
6: The part departs the workcell.

6 Conclusion

During simulations the production of the parts was exposed to various failure types of equipment at different points along the production of the parts. Extensive testing showed that: (i) the ASD can re-route parts as prescribed, (ii) deadlocks can be actively resolved, (iii) the DES Supervisor can successfully enable and disable operations within the workcell, and (iv) the DES Supervisor can continue to function when modified to accommodate changes in the workcell environment.

The work presented in this paper is unique in its use of a supervisory strategy that is regenerated during run-time. It allows the supervisory control strategy to remain within the DES formalism, and it requires only the quick and simple manipulation of the DES supervisory control strategy. Given that the supervisory strategy for the case studied required the utilization of only 1082 states by the HSC, whereas a purely DES-based supervisory controller would have required in excess of 10^{28} states, it is clear that this work allows as well the control moderately complex systems using DES theory, where it was not previously possible. Lastly, the work is original in its application of a model-based technique to the run-time diagnosis of a workcell.

Appendix A

A.1 Explanation of event types and their subscripts

There are four types of events generated by the plants. They are labelled α, β, λ, and μ. Their meanings and controllability are as follows:

α: Plant commences operation (controllable),
β: Plant completes operation (uncontrollable),
λ: Plant fails (uncontrollable), and
μ: Plant reparation completed (controllable).

Due to the complexity of the supervisor developed in our work, information had to be attached to event labels in the form of 5-digit subscripts. The format is given in Figure 8, where stationary plants are buffers, machining-centers, etc., and transport plants are robots, conveyors, etc.

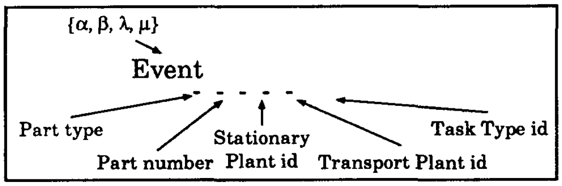

Figure 8: Clarification of events.

Two generic entries appear in many of the subscripts: (−) and X. The (−) means that any event of a class may be generated or permitted, in the case of a plant and specification, respectively. The X indicates that no event of a class may be generated or permitted.

A.2 The plants

The DES Supervisor requires a model of the equipment in the workcell. This central model is constructed from a set of component models — one model for each piece of equipment in the cell. An example of a stationary plant model is shown in Figure 9.

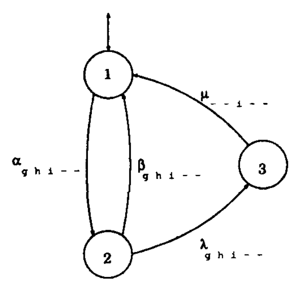

Figure 9: Specification for a stationary plant.

A.3 Specifications

An example of a control specification is shown in Figure 10. The part-activity specification, limits the number of plants operating on a part to a maximum of one stationary plant and one transport plant.

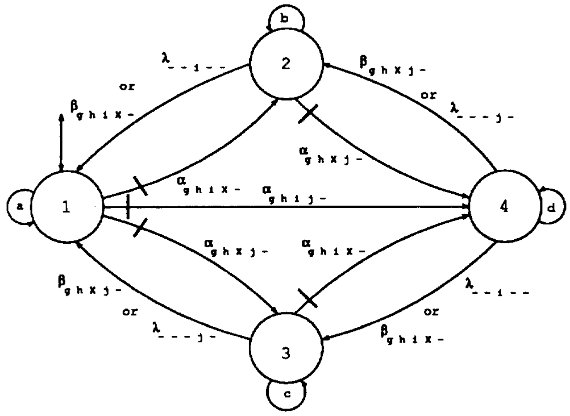

Figure 10: Part-activity specification.

State 1: No plant is operating on the part.
State 2: A stationary plant is operating on the part.
State 3: A transport plant is operating on the part.
State 4: A stationary plant and a transport plant are co-operating on the part.

The selflooping events are:

a {all α_{kl---} where $k \neq g$ and $l \neq h$, all β, λ, and μ}

b {all α_{kl---} where $k \neq g$ and $l \neq h$, all β except β_{ghiX}, all λ except λ_{--i--}, and all μ}

c {all α_{kl---} where $k \neq g$ and $l \neq h$, all β except β_{ghXj}, all λ except λ_{---j-}, and all μ}

d {all α_{kl---} where $k \neq g$ and $l \neq h$, all β except β_{ghXj} or β_{ghiX-}, all λ except λ_{---j-} or λ_{--i--}, and all μ}

References

[1] J. Long, B. Descotes-Genon, and P. Ladet. "Hierarchical and Intelligent Control of Flexible Manufacturing Systems". In *Preprints, 7th IFAC Symposium on Information Control Problems in Manufacturing Technology*, pages 243–248, Toronto, Canada, 1992.

[2] R. David. "Modeling of Dynamic Systems by Petri Nets". In *ECC91 European Control Conference*, pages 136–147, Grenoble, France, 1991.

[3] B. Benhabib, C.Y. Chen, and W.R. Johnson. "An Integrated Manufacturing Workcell Management System". *Manufacturing Review*, vol. 2, no. 4:266–276, 1989.

[4] L.M. Camarinha-Matos and A. Steiger-Garcao. "Robotic Cell Programming: A Knowledge-Based Approach". In *Robotics and Artificial Intelligence 86*, pages 533–551, 1986.

[5] J.S. Ostroff. "Deciding Properties of Timed Transition Models". *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 2:170–183, 1990.

[6] J.S. Ostroff and W.M. Wonham. "A Framework for Real-Time Discrete-Event Control". *IEEE Transactions on Automatic Control*, vol. 35, no. 4:386–397, 1990.

[7] B.A. Brandin, W.M. Wonham, and B. Benhabib. "Discrete Event System Supervisory Control Applied to the Management of Manufacturing Workcells". In V.C. Venkatesh and J.A. McGeough, editors, *Computer-Aided Production Engineering*, pages 527–536. Elsevier, 1991.

[8] P. Ramadge and W.M. Wonham. "The Control of Discrete Event Systems". In *Proc. of the IEEE*, volume 77, no. 1, pages 81–98, 1989.

[9] Y-C. Ho. "Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems". *IEEE Transactions on Automatic Control*, vol. AC-32, no. 7:563–572, 1987.

[10] W.M. Wonham and P.J. Ramadge. "Modular Supervisory Control of Discrete-Event Systems". *Math. Control Signals Systems*, vol. 1:13–30, 1988.

[11] G. Hoffmann and H. Wong-Toi. "Symbolic Synthesis of Supervisory Controllers". In *American Control Conference*, volume 3, pages 2789–2794, Chicago, IL, 1992.

[12] B.A. Brandin, W.M. Wonham, and B. Benhabib. "Manufacturing Cell Supervisory Control - A Modular Timed Discrete-Event System Approach". In *Proceedings of 1993 IEEE International Conference on Robots and Automation*, volume 1, pages 931–936, Atlanta, Georgia, 1992.

[13] S. Balemi. "Discrete Event Systems Control of a Rapid Thermal Multiprocessor". In *Preprints, 7th IFAC Symposium on Information Control Problems in Manufacturing Technology*, pages 53–58, Toronto, Canada, 1992.

[14] W. Hamscher, L. Console, and J. de Kleer. "Model-Based Reasoning: Troubleshooting". In W. Hamscher, L. Console, and J. de Kleer, editors, *Model-Based Diagnosis*, pages 1–3. Morgan Kaufman Publishers, San Mateo, CA, 1992.

[15] Johan de Kleer and Brian C. Williams. "Diagnosing Multiple Faults". *Artificial Intelligence*, vol. 32:97–130, 1987.

[16] R.A. Williams, B. Benhabib, and K.C. Smith. "Model-Based Diagnostics for the Supervisory Control of Manufacturing Systems". In *1994 AAAI Spring Symposium Series*, Menlo Park, California, March 1994. In print.

[17] K. Hadavi, M.S. Shahraray, and K. Voigt. "ReDS - A Dynamic Planning, Scheduling, and Control System for Manufacturing". *Journal of Manufacturing Systems*, vol. 9, no. 4:332–344, 1990.