Intelligent Error Recovery in Flexible Production Systems

Emmanuel D. Adamides, Ekaterini C. Yamalidou and Dominique Bonvin

Institut d'Automatique
Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne
Switzerland

Abstract

Real time failure recovery planning in flexible production systems is considered at the process level. Initially, the problem is structured and formulated within a systemic framework, and then, a generic roll-forward recovery approach that uses the process's dynamic model, as well as general production knowledge, is proposed. The Colored Timed Petri Net formalism is used to represent, in a modular and compact way, the process discrete-event dynamics. In connection with this, an intelligent opportunistic recovery technique for finding, in real time, the recovery trajectory that conforms best within a specific operational strategy, is proposed. The application of the approach is demonstrated by considering cases taken from the chemicals batch-processing industry.

I. Introduction

So far, in sequential mass-production lines, the recovery of a plant from a failure has been considered principally as a situation assessment problem with a localized view and a roll-back perspective. That is, after an alarm signals the existence of an abnormal situation, the monitoring system's primary task is to assess (from diagnosis) the cause of perturbation so that, one or more *reactive pre-defined* actions are triggered locally. The production has to be halted, locally or globally (depending on the size of the failure and the intermediate workstation buffers), until the cause of failure is repaired. To conform with the operating production schedule, partially processed products have to either be thrown away or stored for later consideration and probable repair.

In this framework, the recovery-problem has been addressed by many authors, e.g.,[Moore *et al.*, 1984], [Donald, 1989], [Teng and Black, 1989], [Zhou and DiCesare, 1989], [DiCesare *et al.*, 1993], [Adlemo *et al.*, 1990]. They are mostly concerned with single production resources (machines, robots, etc.) and/or inflexible production technologies with roll-back decisions and actions made and applied locally in space and time without, in most cases, considering consequences at the production-system level.

In multi-product flexible production systems where alternative routing and resource- allocation choices exist, a dynamic adjustment of the process, after a disturbance, is feasible. In other words, the production process need not be halted completely and the repair time can be used, for instance, to operate the system with a different schedule, which, however, tries to meet the main production objectives and priorities (importance of a specific product, importance of a specific customer if produce-to-order, etc.) as close as possible. Alternatively, the repair time can be used to re-process/repair the affected by the disturbance items, or to divert a batch to a slightly different type of product of high demand and/or low inventory costs, etc.

In this line, the error-recovery problem, at the level of production line, can be reduced to the general problem of finding, in an opportunistic way, as imposed by the tight time constraints, the closest to an *ideal recovery state* and/or the best *recovery path* to reach this state from the *error state* within a pre-specified repair time interval. This recovery-state/recovery-path framework can embed easily within a dynamic modeling formalism various operational strategies or short-term operational heuristics. For example, an ideal recovery state can be considered to be the state that the system would reach, under normal operating conditions, after all the in-process products (WIP) exit the line. Complementary, within the recovery interval, apolicy which implements a strategic choice according to which the recovery is driven by the most cost effective path that the system can follow, could have been chosen.

For relatively simple and independent production processes with low flexibility and minimal on-line information availability, where the behavior of the system can be represented fairly well with quantitative abstractions, simple forward dynamic optimizations of the recovery trajectory, to a known or a searched state, seem to be sufficient (Bean *et al*, 1991], [Paul *et al.*, 1992). With the availability, however, of enterprise-wide information systems, the failure recovery problem has to be placed in a wider framework, as a part of the *reactiveness* requirement/need of the modern enterprise, and thus, treated in a *systemic* multi-technology way. A first attempt to address the issue of failure recovery in such a holistic

approach, in a Distributed Artificial Intelligence framework, has been presented in [Adamides and Bonvin, 1993].

In this paper, we propose a general dynamic recovery scheme for flexible production systems which is based on the above view of the problem. More specifically, the *Colored Petri net* (CPN) formalism is used to represent the system's dynamic behavior under both normal and abnormal operating conditions. To be able to exercise control externally, in synchrony with a global clock, an external time scale is introduced. In the CPN context, the recovery problem becomes a kind of optimal *marking control* problem, in which, it is required to determine the transition-firing sequence that will bring the net into a desired final marking at minimum cost. This implies that the behavior of the CPN is modified in a controlled way.

A heuristic-search algorithm evaluates every state (marking) as a potential goal state by considering its relative "goodness" and/or its reaching-path cost. Every state is examined in accordance with a "weighted-resemblance" criterion which, indirectly, implements priority heuristics associated to the various system objects (products, resources, etc.) The search operators (occurrence elements of the CPN) are ordered according to proximity measures and heuristics. Hence, on-line process and product information can be used to adjust dynamically the heuristic parameters. Figure 1 shows the functional architecture of the approach presented in this paper.

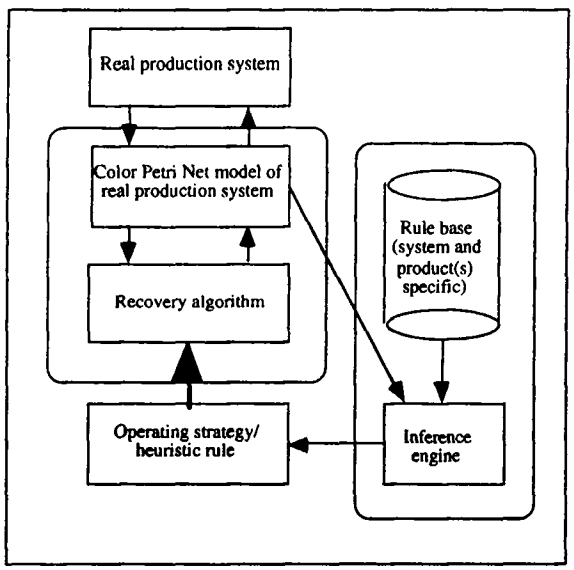

Figure 1: The recovery system's architecture

The *inference engine* module is a rule interpreter whose outcome, after forward chaining, is the best recovery strategy/heuristic for the particular situation. The rules are taken from the rule base and they relate the system's state, as given by the CPN marking, to the causes of failures and to the general production objectives and requirements/ constraints as given in the nominal schedule, the product database, etc.

The application of the proposed error-recovery approach is demonstrated in the recovery of a batch-processing flexible production line. Various recovery strategies are considered.

II. Externally controlled CPN

The Petri net, in its various types and forms, has been a popular functional and dynamic modeling formalism for manufacturing control applications at various levels. The merging of Petri net modeling and heuristic search has been proposed for the scheduling of simple flexible manufacturing systems (Lee and DiCesare, 1992). A simple timed Place-Transition net (P/T-net) has been used to generate the states in the search for a final "all-scheduled" state. Some very simple heuristics functions to guide the search efficiently have also been tried. The optimal schedule is obtained as the minimum-cost sequence of transition firings.

Heuristic search algorithms guide the search according to heuristic metrics that either evaluate states in relation to a target state to provide (order) dynamically the "next-state" operators. The Petri net is a modeling formalism that can represent the dynamic behavior of the production system in relation to the discrete events that take place in the shop floor. Therefore, compared to quantitative abstractions in the form of flow equations or to pure knowledge-based techniques where temporal behavior of parallel processes is difficult to be managed, it can represent better the behavior of a dynamic process. Furthermore, the Colored Petri net (CPN) (Jensen, 1992) offers a compact modeling framework which has an inherit hierarchical representation of process detail and, thus, it can be "unfolded" selectively to reveal behavioral details and properties of interest without needing to be considered it in its whole. This inherit hierarchy, together with its structure, offer a very attractive basis for the development of an intelligent recovery approach which essentially uses the same (normal operation) process model.

The CPN has been defined formally by Jensen (Jensen, 1992). In an informal way, a CPN is a PN with tokens of different *colors* (types) and functions (*expressions*) associated with its arcs. The arc expressions translate the relation between the color associated with and chosen to fire this transition (*firing color*), and the colors associated with the tokens in the corresponding place (*token color*). A *guard* of a transition is a Boolean expression which must be evaluated to true before the transition can *occur/fire*. In addition, every transition has a number of variables associated with it (variables from the incoming arc expressions). Before the transition fires, these variables have to be *bound* to colors of the corresponding types. This means that a transition of a CPN can fire in many different

ways according to its *binding*, its guard and the specific net marking. A pair, where the first element is a transition and the second a binding of that transition, is called an *occurrence element*. A transition may be enabled for many different bindings at a particular time instance and many occurrence elements may be enabled at the same time. A *step* includes all the enabled occurrence elements i.e. it is a non-empty and finite multi-set over the set off all net bindings.

In order to control the firing of the CPN, one has to control/restrict the values of the bindings of the net's transitions and/or to adjust the guards associated with the transition. The control can be in a hierarchical manner, i.e., if a transition, for some reason (e.g. time), is not enabled at all, it is not considered, whereas if it is, the various firing possibilities can be examined and graded according to static or dynamic operational criteria imposed by strategic choices. Concurrently enabled transitions have to be checked for possible conflicts. In line with this, the search for the best recovery state (CPN marking) takes place by manipulating transition bindings and guards using static, and dynamically generated knowledge by the problem and its solution process, respectively.

In addition, to control the system in a deterministic way, an external time scale has to be introduced. The external time scale synchronizes the evolution of the system to discrete time steps by restricting the transitions of the Petri net to fire, if enabled, at specific discrete time instances. The inclusion of time information in the model helps decision making in the cases where the production processes are structured and the timing almost deterministic. A *global system clock* is introduced and, in addition to its color attribute, every token has a *time stamp* (Jensen 1992) attached to it. For our modeling framework, the time stamp is a pair (α,β), where α is the earliest time that the token can be removed by a binding element and β is the maximum time allowed for a token to stay at a place. Both elements refer directly to the global time scale whose values increase continuously without re-initialization. To calculate the time stamps of the tokens from the production recipe, *time clauses* are assigned to the output arcs of every transition. A time clause specifies for every color the minimum and maximum delay values that are added to the current reading of the global clock to produce the pair (α,β).

III. Intelligent search with Colored Petri nets

In order to accomplish the heuristic search on the marking space of CPN, in a hierarchical manner, the operator ordering function O is defined. In its generic form, it is defined as a map of CPN markings onto an ordered set of occurrence elements, i.e. into a list of ordered CPN *steps* (Jensen 1992)

$$O : M_X \times M_{id} \times S_X \longrightarrow S'_X$$

where S_X is the order of the step before marking M_X, and S'_X is the order after marking M_X has been reached. M_{id} is the target ideal recovery marking.

Within the context of the domain considered in this paper, O can be applied in connection with the proximity measure discussed below, i.e., operators are ordered according to the results of the evaluation of the (weighted) proximity measure. In addition to ordering, the elimination of some of the step's elements is possible. Occurrence elements are always ordered with primary key the heaviest-weighted attribute.

For the system to be able to search for the "best" state and/or the best path to this state, a relative evaluation measure for each state and every path has to be defined. The two metrics (resemblance and energy/cost) can be used to evaluate the states/markings reached. In the recovery algorithm given in the next section, states are evaluated only in relation to the proximity measure.

A. The proximity/goal attainment metric

In evaluating the resemblance of a state in relation to the ideal state, different colors in the same place have different degrees of importance as imposed by the recovery strategy employed. Hence, the proximity or deviation of a state to/from the goal state is evaluated by a function of the weighted difference of the two states. The weights are specific to the recovery strategy.

Since the marking m of a CPN is the set of markings of the individual places

$\mathbf{Pr}[\mu(i)-\mu(y)]$ = {weighted difference in number of tokens for every net place}

$$= [W_{Pl1}\,(w_{c1}*\Delta(c1) + w_{c2}*\Delta(c2) \ldots) + W_{Pl2}(w_{c1}*\Delta(c1) + w_{c2}*\Delta(c2) \ldots) \ldots\ldots W_{Ply}\,(w_{c1}*\Delta(c1) + w_{c2}*\Delta(c2) \ldots)]$$

where W_{Pli} is the weight of place *Pli*, w_{ci} is the weight of color *ci*, and $\Delta(c1)$ is the difference in tokens of color *ci* between the state being examined and the goal state.

By quantifying the difference between the ideal state/marking and an arbitrary marking of the CPN, a degree of relaxation of optimality (100% resemblance), e, can be defined as

$$|e| = |\ \mathbf{Pr}[\mu(i)-\mu(y)]\ |$$

It must be noted that recovery algorithms that guide the search using the above cost measure *only*, can be developed and used in cases where the recovery cost per se is the primary minimization factor, a recovery target state is known and its reachability is guaranteed. In this case, the problem can be formulated as a dynamic program whose solution for large systems can be obtained using hierarchical decomposition techniques. In reality, however, the existence of such conditions is a rare case.

IV. A proximity function-based recovery algorithm

Taking into account the discussion in the previous section, the proposed generic algorithm to be used for the recovery of failed processes is presented below in a hierarchical and modular way:

ALGORITHM Recovery
Until a marking that satisfies the termination condition is found **do**
1. Fire enabled transitions and obtain new marking;
2. Examine marking and sort operators;
end.

ALGORITHM Fire enabled transitions and obtain new marking
1. **For** every CPN transition in the current marking **do**
1.1 Deduce from high-level recovery plan and markings of input places if transition has to be considered at this step;
1.2 **If** it need **not** be considered **then** exit
else
1.2.1 Use sorted occurrence elements list to bind transitions with the most promising bindings
1.2.2 Use external heuristics to resolve any firing conflicts;
1.2.3 Fire transitions and compose new marking;
end if; end for;

ALGORITHM Examine marking and sort operators
1. **With** the current marking **do**
1.1 Check if the marking satisfies the termination condition;
1.2 Calculate cost of arriving at the marking from the initial marking;
1.3 Calculate measure of proximity to the ideal target marking;
1.4 (Re)order occurrence elements in accordance with the results of the measure of proximity;
end;

The execution of the above algorithm(s) requires the specification of the occurrence-elements ordering function, which is either pre-specified or is given interactively by an operator. Recovery strategies and the information structures required to accomplish them are discussed in detail in [Adamides and Bonvin, 1993]. The termination condition of the *Recovery* level may have various forms, including step count, relaxation of optimality measure, etc.

V. Example of application of the proposed algorithm

To demonstrate the development and use of the proposed recovery method, a multi-purpose chemical batch plant with five units and three products is considered. Batch plants consist of a group of identical or different processing units that can process several products at the same time. The reference system consists of five reactors sized at 1000lt., 2000lt., 3000lt., 4000lt. and 5000lt. A single buffer having a capacity of 5000lt. also exists. Three products are produced and they are identified by A, B and C. Each of the units is equally capable of performing any of the tasks. Merging of batches of different products in any resource of the system is not allowed. The size of the batches entering the system is always a multiple of 500 lt. Since the recipe for A and B requires a doubling of the batch size in the second stage, then the only batches of A and B that may enter the system from the supply feed are the 500, 1000, 1500, 2000 and 2500 lt. The batches of C require only one processing stage, therefore, they are sized at 1000, 2000, 3000, 4000 and 5000 lt. Product A requires 3 time units in its first stage and 5 time units in the second. Product B requires 4 and 3 time units, respectively. Finally, the single processing step of product C takes 6 time units.

The structure of the CPN which models the "uncontrolled" process for the reference system is shown in figure 2 Arc expressions and quards are not shown, but they are defined so that the net models the above operational policy. Places p1 and p2 act as resource pools supplying resources to the process. p3, p4, p5 and p6 form the general set of states of the processing sequences for all the products. p7 and p8 signal errors in the system, whereas p9 is a repository for the tokens that represent failed reactors. Tokens remain there whilst the corresponding reactors are being repaired. The *guards* of the transitions ensure that the size of the unit assigned to a batch should be greater or at least equal to the size of the batch.

The occurrence of errors in the system is manifested by the presence of a tokens in places p7 and/or p8. Transitions t5 and t9 are fired by the physical cause of the errors. A failed batch from the first production stage can be recycled (t7) or it can be taken out of the system (t6). The colored token representing the failed reactor enters p9. Failure during the second stage implies the immediate dismissal of the failed batch. The buffer is assumed to form the hard core of the system, i.e. it is a completely reliable device. However, its contents can be discharged.

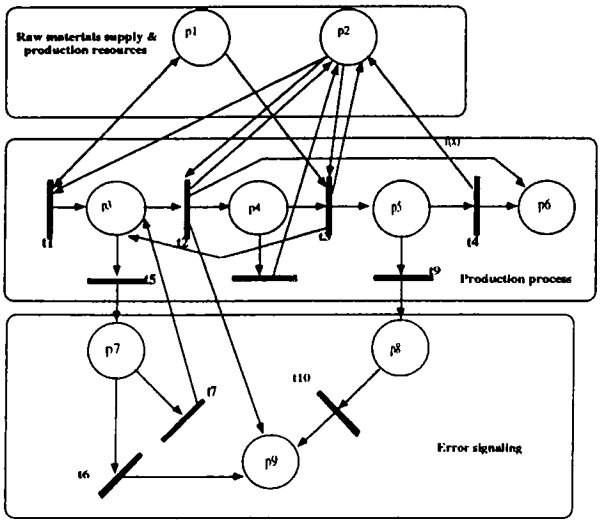

Figure 2: CPN model of the production process

We assume that the failure happens at a system state:

(R1, 1B, 3, a), (R2, 2A, 2, a), (R3, Ø, 0, -), (R4, 2A, 0, b), (R5, 5C, 0. a), (B, 5A)

where *Rl* represents unit (reactor) *I*, the second element in the parenthesis denotes the batch, 0,1,2,3.. denotes the time the batch (type and quantity) has already spent in the specific reactor and a and b are the processing stages. In the first recovery case presented below, we assume that the failed batch can move to the next processing stage, whereas in the second, it needs re-processing. The buffer is assumed to consist of 10b (b = 500lt) units and, in all cases, the decision to re-process the failed batch is assumed to have already been made.

CASE 1: Product B has very tight due dates and very high WIP costs. Product in failed reactor has been processed enough as to be considered as having completed the corresponding processing stage.

Target state: Place p6 must contain token 2B.

Constraint: Recovery interval is 15 time units.

Basic operator ordering rule: B has higher priority in any case over other products

The recovery trace for this case is shown in figure 3. The application of the priority rule can seen in time step t=7, where emptying the buffer to advance processing of product B is preferred from pouring a type A product into the buffer (transition t3 fires to empty the buffer and then t2 is bound with the color B).

CASE 2: Failed batch needs re-processing at stage 1.

Target state: Place p6 must contain as much as possible from all products. There is no product-importance weighting.

Constraint: Recovery interval is 20 time units. If it not possible to recover within this time interval failed batch will have to be thrown.

Basic operator ordering rule: Failed batch has highest priority.

Figure 4 shows the recovery trace for this case. At time step t=7, the error batch is poured into the buffer since it requires two stages of processing. In the CPN model, t3 fires to empty the buffer, then t2 is bound to color A and fires. Immediately after, t3 is bound to the same color, as far as product is concerned, but the token returns to place p3 to signal that stage 1 processing again.

The application of the proposed roll-forward recovery approach to the cases described above has been made in our prototyping set-up consisting of the CPN development and simulation environment SPECS, provided by Landis & Gyr Betriebs AG, and the hybrid AI workbench BABYLON [Cristaller *et al.*, 1992]. In this set-up, different to the above presented strategies can be very easily implemented without many changes.

p1	p2	p3	p4	p5	p6	time
	R3	1BR1	10bA	2AR4		
		2AR2(e)rror				
		5CR5				
			(t3)			(t=1)
	3b	1BR1(T)	7bA	2AR4		
		2AR2e		3AR3		
		5CR5				
						(t=2)
<<<<	<<<<	<<<<	<<<<	<<<<	<<<<	<<<<
			(t4)			(t=5)
	3b	1BR1(T)	7bA	3AR3(T)	2A	
	R4	2AR2e				
		5CR5(T)				
			(t2,t3,t4)			(t=6)
	7b	1BR1(T)	3bA	4AR4	5A	
	R3	2AR2			5C	
	R5					
			(t3)			(t=7)
	10b	1BR1(T)	Ø	4AR4	5A	
	R5	2AR2e		3AR3	5C	
			(t2)			(t=8)
	8b	2AR2e	2bB	4AR4	5A	
	R5			3AR3	5C	
	R1					
			(t3)			(t=9)
	10b	2AR2e	Ø	4AR4	5A	
	R1			3AR3	5C	
				2BR5		
			(t2)			(t=10)
	6b	Ø	4bA	4AR4(T)	5A	
	R1			3AR3	5C	
				2BR5		
			(t4)			(t=11)
	6b	Ø	4bA	3AR3(T)	9A	
	R1			2BR5(T)	5C	
	R4					
			(t3,t4)			(t=12)
	10b	Ø	Ø	4AR4	12A	
	R1				2B	
	R3				5C	
	R5					
			All WIP exits at t=15			

Figure 3: Recovery trace for case 1

p1	p2	p3	p4	p5	p6	Time
		(Steps t=1 to 6 as in case 1)				
						(t=6)
	7b	1BR1(T)	3bA	4AR4	5A	
	R3	2AR2(e)			5C	
	R5					
			(t3)			(t=7)
	6b	1BR1(T)	Ø	4AR4	5A	
	R5	2AR2(e)		3AR3	5C	
			(t2)			(t=8)
	6b	1BR1(T)	4bA	4AR4	5A	
	R5			3AR3	5C	
			(t3)			(t=9)
	10b	1BR1(T)	4bA	4AR4	5A	
		4AR5		3AR3	5C	
			(t2)			(t2)
	8b	4AR5	2bB	4AR4(T)	5A	
	R1			3AR3	5C	
			(t4)			(t=11)
	8b	4AR5(T)	2bB	3AR3(T)	9A	
	R1				5C	
	R4					
			(t3,t4)			(t=12)
	10b	4AR5(T)	Ø	2BR4	12A	
	R1				5C	
	R3					
			(t2)			(t=13)
	2b	Ø	8bA	2BR4	12A	
	R1				5C	
	R3					
	R5					
			(t3)			(t=14)
	10b	Ø	Ø	2BR4(T)	12A	
	R1			3AR3	5C	
				5AR5		
	<<<<	<<<<	<<<<	<<<<	<<<<	<<<<
						(t=19)
	10b	Ø	Ø	Ø	20A	
	R1				2B	
	R3				5C	
	R4					
	R5					

Figure 4: Recovery trace for case 2

VI. Conclusions

We have presented a systems approach to failure recovery in flexible production systems. We have shown how the problem of roll-forward error recovery at the production-system level can be rationalized and how the Colored Petri Net formalism can be used to represent, in a compact and modular way, the system's dynamic behavior, during the recovery phase, as in normal operation. In addition, we have shown how the decision making process can be simplified by structuring the decision space by using process knowledge. Finally, we have shown how the controlled firing of the CPN, in connection with various effectiveness measures, can implement different recovery policies The separation of the recovery strategy from the systems dynamics, implies that various policies deduced from the system's state, the global production strategy and the available information can be applied without any need of changing the system model.

References

[Adamides and Bonvin, 1993], E.D. Adamides, and D. Bonvin, "Error Recovery of Flexible Manufacturing Systems Through Cooperation of Distributed Intelligent Agents", *Proc. of KNOWHSEM '93* pages 227-238, Budapest, April 1993.

[Adlemo *et al.*, 1990], A. Adlemo, S.-A. Andréasson, T. Andréasson, and C. Carlsson, "Achieving Fault Tolerance in Factory Automation Systems by Dynamic Configuration", *Proceedings of the First International Conference on Systems Integration*, Morristown, NJ, pages 396-402.

[Bean *et al.*, 1991], J.C. Bean, J.R. Birgle, J. Mittenthall, and C.E. Noon,"Matchup Scheduling with Multiple Resources, Release Dates and Disruptions", *Operations Research*, 39(3):470-483, 1991.

[Cristaller *et al.*, 1992], T. Christaller, F. di Premio, U. Schnepf and A. Voss (edited by), "The AI Workbench BABYLON", Academic Press, 1992.

[DiCesare *et al.*, 1993], F. DiCesare, G. Goldbogen, D. Feicht, and D.Y. Lee, "Extending Error Recovery Capability in Manufacturing by Machine Reasoning", *IEEE Trans. Systems, Man and Cybernetics*, 23(1): 221-228, 1993.

[Donald 1989], B.R. Donald, "Error Detection and Recovery in Robotics, *Lecture Notes in Computer Science*, No. 336, Springer-Verlag, 1989.

[Jensen, 1992], K. Jensen, "Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1", *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, 1992.

[Lee and DiCesare, 1992], D.Y. Lee and F. DiCesare, "FMS Scheduling Using Petri Nets and Heuristic Search",, *Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation*, Nice, France, pages 1057-1062, 1992.

[Moore *et al.*, 1984], R.L. Moore, L.B. Hawkinson, C.G. Knickerbocker and L.M. Churchman, "A Real-Time Expert System for Process Control", *Proceedings of the First Conference on Artificial Intelligence Applications*, pages 569-574, 1984.

[Paul *et al.*, 1992], C.J. Paul, L.E. Holloway, D. Yan, J.K. Stronsnider, and B.H. Krogh, "An Intelligent Reactive Monitoring and Scheduling System", *IEEE Control Systems Magazine*, pages 78-85, June 1992.

[Teng and Black, 1989], S.-H. Teng and J.T. Black, "An Expert System for Manufacturing Cell Control", *Computers and Industrial Engineering*, (17)1-4:, 18-23, 1989.

[Zhou and DiCesare, 1989], M.C. Zhou and F. DiCesare, "Adaptive Design of Petri Net Controllers for Error Recovery in Automated Manufacturing Systems", *IEEE Trans. on Systems, Man and Cybernetics*, (19)5:963-973, 1989.