Prognosis and recovery evaluation in flexible manufacturing systems supervision

M. Da Silveira[1][2][3], M. Combacau[1], A. Boufaied[1]
{combacau, dasilveira, aboufaie} @ laas.fr
[1]LAAS-CNRS - 7 Avenue du colonelRoche - 31077 TOULOUSE - cedex 04 FRANCE
Université Paul Sabatier - 118 Route de Narbonne - 31044 TOULOUSE - cedex 04 France

[2]Pontifícia Universidade Católica do Paraná PUCPR
Laboratory of Automation and Systems - R. Imaculada Conceição, 1155 - 80215-901 CURITIBA – BRAZIL
[3]CNPq – Centro Nacional de Desenvolvimento Científico e Tecnológico

Abstract: This paper presents some solutions of decision-making aspects integrated in an approach of process failure monitoring suitable for complex systems like flexible manufacturing systems. A hierarchical and modular structure corresponding to the monitoring and control requirements is described. Each control and monitoring module is able to accept any signal from the lower levels of the hierarchy and must be able to elaborate an acceptable solution for each failure. When a failure is detected, the elaboration of a curative solution is furthered by the knowledge of the details of the current situation. The prognosis function has to foresee the consequences of the failure on the future operation of the system in order to give relevant data to the decision function. Recovery evaluation is necessary to foresee what will be the consequences of the introduction of a corrective sequence on the current schedule.

Keywords: reliability, supervision, monitoring, fault diagnosis, process failures

INTRODUCTION

Design of control systems of automated manufacturing processes, using formal models (Petri nets, automata), normalized implementation techniques (GRAFCET, LADDER) and specific hardware (PLC) well suited to hard conditions of workshops, have now reached a satisfactory level of dependability. However, the confidence placed by the user in the control system does not elude the problem of unforeseen and uncontrollable process malfunctions. The research on production system control focuses on this failure aspect with respect to industrial preoccupations.

The main difficulty in this subject concerns the various situations and data that are encountered. A supervision and monitoring system must be able to recognise abnormal situations. It means that it has to deal with time and behavioural aspects of the evolutions. Then, the origin of this situation must be found. The corresponding research is typically a diagnosis problem. So, partial information, imprecise data, heuristic knowledge, etc. has to be taken into account. At last, when the problem is well identified, a curative solution must be elaborated. Actions on the control system and on the controlled process have to be performed. Naturally, the interaction with human operators and the processing of emergency situations must be guaranteed. Moreover, all correct treatments of erroneous situations have to be compatible with the scheduling of the factory: corrective treatment must not take too long time and must not mobilize resources allocated to other tasks.

Two classes of solutions have been developed to ensure process operation: failure avoidance and failure processing.

Failure avoidance: it consists in foreseeing the unusual evolutions of a process and in minimizing the probability of their occurrence. This can be done by introducing material redundancy or by giving flexibility (degrees of freedom) to the control system. Such an approach cannot ensure that no failure will occur.

Failure processing: some failures can be foreseen and their processing can be integrated into the controller. On the other hand, a failure without a dedicated processing cannot be treated by the controller. So, it must be taken into account by some other functions constituting what is called a monitoring and supervision system.

This paper presents a contribution to the domain of error recovery.
Precise definitions of some terms used in the paper are given first. The second section briefly describes the hierarchical and modular architecture in which these works are integrated. Part three deals with the prognosis function and part four describes the mechanisms used to evaluate a recovery sequence. Part five describes an illustrative example.

1 Basics concepts and definitions

This section presents some definitions taken from (Combacau *et al.*, 2000) a collaborative paper of French researchers of the production systems community. Control, monitoring and supervision are defined first, then the definition of terms used in this paper are given.

Control: it triggers the execution of a set of operations by giving orders to the process actuators. It may be:

A set of operations corresponding to the manufacturing sequence of the product.

A set of operation executed to restore the process functionality offered during normal execution.

Actions with a high priority level engaged in order to protect the shop workers and to prevent catastrophic evolutions.

Some checking, tuning or cleaning operations executed in order to maintain the process in its operational state.

It means that our definition of control includes all the functions actully acting on the process.

Monitoring: it collects data from the process and from the controller, it determines the actual state of the controlled system and it makes the inferences needed to produce additional data (historic, diagnosis, etc.). Monitoring is limited to data processing and has no direct actions on the models or on the process.

Supervision: computes and set the parameters of the control sequence to be executed according to the state of the control system and to the state of the process. It includes normal and abnormal operations.
During normal operation, the supervision takes the decisions to raise the indecision in the control system (real-time scheduling, optimisation, control sets and switching from a control law to another one).
When a process failure occurs, supervision takes all the decisions necessary to allow the system to resume its normal operation (rescheduling, recovery actions, emergency procedures, etc.).

It must be noticed that supervision takes place in a hierarchical structure (with at least two levels). At the lowest level of the structure only the control and monitoring functions are generally implemented - no real decision have to be taken.

Some generic terms need to be defined. Some of them can be found in (Laprie, 1992).

Fault: action, voluntary or not, that does not take all the specifications into account.

Defect: difference between the actual value of a parameter and its nominal value.

Error: part of a model not exactly matching the specifications of the physical system. Logically, an error is a consequence of a fault.

Latent error: the error is qualified as latent as long as the erroneous part of the model has not been used. After using the erroneous part of the model, the error becomes effective.

Failure: event characterizing a situation in which an operation is not executed by a resource because its state does not correspond to the nominal specifications any longer.

Breakdown state: state of a resource from which the system cannot provide the specified service. This state is a consequence of a failure.

Symptom: event or data through which the detection system identifies an abnormal operation of the process. The symptom is the only information the monitoring system knows at the detection step.

Recovery point: state reachable from the breakdown state in which the system must be driven to resume the normal operation.

Recovery sequence: set of ordered actions executed to bring the process back from the breakdown state to the recovery point.

According to these basic concepts, we can define the elementary functions of a supervision and monitoring system. Between brackets the letter M, S or C indicates to which previously discussed group (Monitoring, Supervision, Control) the function belongs.

Detection (M): determines the normality or abnormality of the functioning system. Two classes of abnormal operations are considered :

The first one includes situations in which basic operating constraints of the process are violated (collisions for instance).

The second one groups together situations in which the part routing (control law) is not respected (fabrication delays for instance).

Follow (M): maintains a history of treatments executed and a trace of events observed by the control/supervision system.

Diagnosis (M): looks for a causality link between the observed symptom, the failure and its origin. Classically, three sub-functions are distinguished:

Localization determines the subsystem responsible for the failure,

Identification identifies the causes of the failure,

Explanation justifies the conclusions.

Prognosis (M): foresees the consequences of a failure on the future operation of the system. The consequences can be immediate ones (resource unavailable) or induced ones (faulty parts in the workshop).

Decision (S): determines the state that must be reached to resume to the normal operation, then determines the sequence of corrective actions to be performed to reach this state.

Recovery (C, S): acts on the process by changing the states of the resource or equipment and on the control system by changing the control laws, the part routing, etc. Three classes can be defined:

Minor, only the control laws are adapted,

Significant, other resources are reallocated,

Major, reallocated resources need to be prepared to execute the recovery.

2 Hierarchical monitoring and supervision

Previous works have defined a hierarchical and modular approach of monitoring and supervision (Combacau, 1991; Chaillet, 1995). It must be noticed that only process failures are taken into account. The control and monitoring system is considered as error free. When a process failure occurs, the corrective actions to be executed are performed according to the process resources used, to the kind of manufactured products and to the production strategy specified by the user.

The failure processing is not limited to the classical sequence (detection, diagnosis, decision and recovery) (Zamaï *et al.*, 1998).

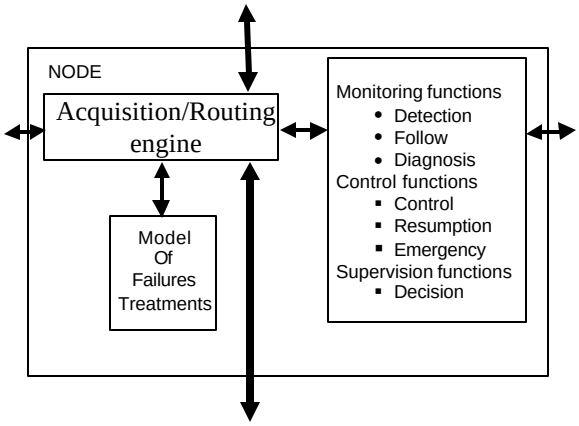

Figure 1: A generic module for control, supervision and monitoring

The Acquisition/Routing block manages all these functions. This block is based on an algorithm directing incoming messages to the most suitable functions according to the nature of the data and to the state of the monitoring system. Moreover, this algorithm maintains the state of this model and triggers the suitable monitoring, control and/or supervision functions (Combacau *et al.*, 1998).

From a specification point of view, this approach is based on complementary tools. Petri nets with Objects are used to model control, recovery and emergency sequences and failures treatments. An extended entity relationship model provides a process representation (called Information System) in which data that are not easy to model by means of Petri nets (time notions like dates, duration, history to keep a track of data evolution, etc.) can be found.

Most of monitoring and supervision functions have already been described by other publications (Berruet *et al.*, 1998; Dangoumau *et al.*, 1999; Combacau *et al*, 2000). Our recent works focus on the integration in this structure of the prognosis function and to some aspects of the decision function : the recovery evaluation. The sequel of the paper is dedicated to these recent results.

3 The prognosis function

The role of the prognosis function is to determine the consequences of a failure on the future functioning of the workshop. Two aspects have to be considered.

- Whatever the diagnosed failure, what are its unavoidable consequences? **(**Preliminary prognosis)
- If a latent error coming from a failure that has not be immediately detected and if the diagnosis has pointed out the cause of the symptom occurrence (this is what we call “fault

propagation"), how many latent errors have been introduced in the workshop by the use of the faulty resource? (Preventive prognosis)

3.1 Preliminary prognosis

At this step of the corrective treatment, only the failure and its causes are known. To foresee the effect of the fault, three kinds of data can be taken into account:

The scheduling of the tasks to be executed by the faulty resource.

The off–line analysis of the failure consequences (FMEA for Failure Mode and Effect Analysis) (Villemur, 1998).

Statistical data about the breakdown state of the resource (MTTR for Mean Time To Repair).

In a first step, the FMEA of the failure is used to propagate the foreseeable consequences of the failure within the pool of resources. An FMEA consists of a set of causal links between failure modes of resources. This analysis is made off-line by taking into account an inventory of the components, of their failure modes and of their effects on the system. This inventory is mainly based on the physical links between components to show the tasks that cannot be correctly executed when a failure occurs. An original extension of the basic FMEA can be found in (Berruet *et al*., 1999).

When the set of resources related to the fault treatment is identified, an unavailability period is allocated to each resource. This period is the MTTR of the diagnosed failure. By doing this, we suppose that an induced effect of the failure will not take longer to be repaired than the failure itself. The consequence of this assumption is that the result reached by the prognosis function is the minimal set of constraints to be taken into account. The completeness of the set of constraints to be respected is a property that the decision function deals with.

In a second step, without any consideration on the recovery sequence (yet, it has not been determined), the unavailability period of the resources pointed out at the previous step is inserted as soon as possible in the scheduling of each resource: immediately, if the resource is not currently working, and at the end of the current task if the resource is working. The ready dates (r_i - date when the task "i" is ready to start) of the next tasks to be executed by the resource are delayed to ensure the consistency of the sequence (the effective ready date is used to avoid starting a task before its original scheduling, equation 3). The execution delay of a task is computed by its sequential margin defined by the difference between the earliest finishing time in the modified schedule and the due date (d_i) in the original one. For a product, the minimal delay that can be foreseen is the maximum of the sequential margins associated to the tasks constituting the sequence of fabrication.
The procedures to compute the sequential margin are: Compute the new ready dates (r_i'), the effective ready dates (r_i'') and the sequential margin.

New ready dates:

$r_0' = MTTR + T$, (1)

$r_{i+1}' = r_i' + p_i$., for $i = 0 \ldots n-1$ (2)

T is the failure date.
p_i is the processing time of the task "i".
n is the number of tasks not yet executed.

Effective ready dates:

$r_i'' = Sup(r_i', r_i)$, for $i = 0 \ldots n$ (3)

Sequential margin:

$Sequential\_Margin_i = d_i - r_i'' - p_i$ (4)

The delay of a product "i" is the delay of the last task executed on the resources related to the fault. The knowledge of a delay induced by the failure is of main interest for decision purpose because, if a delay exists, the error recovery cannot be performed by a local failure processing. Finally, the control and monitoring module must propagate the failure processing to the upper level of the hierarchy by indicating that the schedule will not be respected any longer.

3.2 Preventive prognosis

We only consider the failures which effects are not immediately detected. In this case, a machining defect affects all products processed since the failure has occurred. An error propagation can happen between activities according to the routing of this product.

When the defect on the product induces the detection of a symptom, the diagnosis determines the origin of this symptom. At this step, the preventive prognosis determines whether there are other products affected by a similar defect that will induce the same symptom in the future.

For this purpose, the prognosis function uses the history of the tasks executed by the resource. We assume that all products machined after the failure occurrence are affected by the same defect. The main problem is to determine the first activity that has produced the considered defect. Very often, in a flexible manufacturing system, a resource executes sequentially different tasks related to different fabrications. Each product having its own routing in the workshop, the first one leading to the detection of the defect is not necessarily the first one produced with this defect.

The identification of the first product with the defect is technically speaking a complex problem, because the defect cannot be automatically identified. The technical characteristic of each suspected part has to be checked manually. If an activity of the same type as the one that has led to the symptom detection has been previously executed without any problem, then we can consider that the product implicated in this activity was not affected by the defect. This piece of information gives a bound to the backward research of the first faulty part. Otherwise, the heuristic approach has to be used or, at the last resort, the prognosis can be forgiven.

4 Evaluation of a recovery sequence

Whatever the situation, a local failure processing must be performed. This section focuses on the elaboration of the recovery sequence by the decision function and more specifically on the evaluation of the recovery sequence timing.

The decision function elaborates a corrective treatment called the recovery sequence. Globally, the recovery sequence is defined by the following date.

- Its initial state: it is the current actual state of the process determined by the diagnosis and prognosis functions.
- The tasks necessary to put the process back in its nominal state: it constitutes the technical recovery sequence (TRS). This is the main decision-making aspect of this function.

The sequence of already scheduled tasks on the resources related to the failure treatment and whose execution dates have to be modified to insert the TRS at the current date. The set of such tasks constitutes the real recovery sequence (RRS), because of it, the original schedule can only be respected again when all of these tasks have been executed that.

The evaluation consists in determining the set of tasks that must be delayed to allow the TRS insertion and their new execution dates. In other words, the evaluation computes the new execution dates of the tasks belonging to the RRS. The global algorithm of the decision function is the following one:

Begin
Compute the TRS (not described in this paper)
Insert *the TRS at the current date*
Evaluate *the RRS*
While (RRS is not admissible) do
Begin
Negotiate with the upper level a delay on some products
Delete these products from the previous RRS
Evaluate *the RRS*
End
Send the new schedule to the recovery function
End

If the solution (the real recovery sequence) can be locally executed without inducing any disturbance on the remainder of the workshop, the loop **`while`** is not executed. It is often the case when the recovery sequence is very simple (for instance, the task must be forgiven, continued or re-executed).

The aim of this section is to give an algorithm suitable for the RRS evaluation. This problem is related to the one described in (Artigues and Roubellat, 2000). The RRS evaluation consists in computing new starting dates for scheduled tasks. In our case, the computing time is optimized by taking into account two specific constraints: first, the TRS must be inserted at the current date and second the execution order of the tasks is not modified. The modifications of the dates are computed by constructing a graph in which the new constraints imposed by the insertion of the recovery sequence are propagated. At the end of this process, the graph contains all the tasks which dates have to be modified if this solution is accepted by the decision function.

The construction of this graph uses a vertex to represent a task, each input arc of a vertex represents a sequencing constraint between two tasks, each task T_i is associated with its latest processing time interval $[startT_i, finishT_i]$ of possible execution.

The propagation consists in inserting the recovery task (TRS) as the first one executed by the faulty resource. Then, we compute for each node T_i, following the increasing order of starting date of activities, the new starting ($startT_i'$) and finishing date ($finishT_i'$) by:

$$[startT_i', finishT_i'] = [\underset{T_j \in T}{Max}(finishT_j'), startT_i' + (finishT_i - startT_i)]$$

(T is the set of immediate predecessor of vertex T_i)

The procedure stops when we have, for a task T_i, $startT_i' <= startT_i$ (the execution date of this task need not be modified)

Then, this new graph is sent back to the decision function, the evaluation of the consequences of the recovery sequence introduction at the current date being complete.

5 Illustrative example

5.1 System Description

In this section, we propose an example of a prognosis method and a recovery sequence evaluation of a FMS subsystem providing Assembling, Soldering, Painting and Packaging operations {A,S,P,K}.

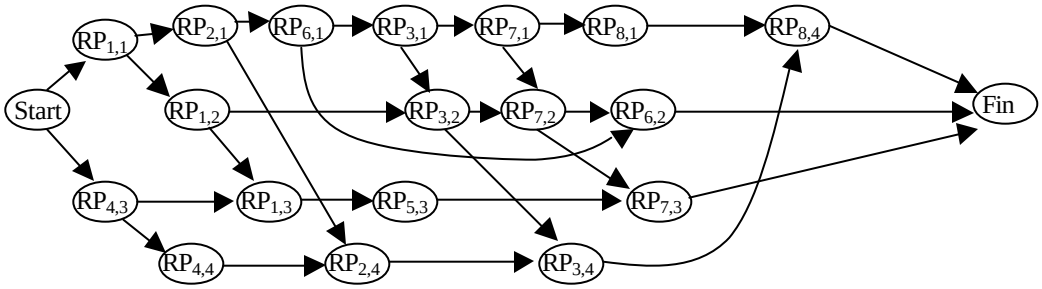

Figure 2: Sequence Constraints Graph

The operations are executed by four resources {R_1, R_2, R_3, R_4}. The transportation aspects are not considered in this example. Raw parts (RP) are manufactured in six different products by the activities (associations of operation tasks): {{A,S,P},{A,T},{A,S,T},{P,T},{P},{A,S}}.

The proposed example considers a predefined set of activities during a well-defined time interval, where a raw part "i" manufacturing by a resource "j" is represented by $RP_{i,j}$ in the text.
The resource schedule is shown by Table 1 and a particular case is illustrated by Figure 3.

R_1: Assembling								
$RP_{i,1}$	1	2	3	4	5	6	7	8
r_i	0	0	1			1	4	5
di	2	5	4			4	7	9
R3: Painting								
$RP_{i,3}$	1	2	3	4	5	6	7	8
r_i	1			0	3		6	
di	4			2	5		11	

R_2: Soldering								
$RP_{i,2}$	1	2	3	4	5	6	7	8
r_i	1		3			5	4	
Di	3		5			8	8	
R4: Packaging								
$RP_{i,4}$	1	2	3	4	5	6	7	8
r_i		2	4	0				7
Di		5	8	3				8

Table 1: Scheduling of normal processing.

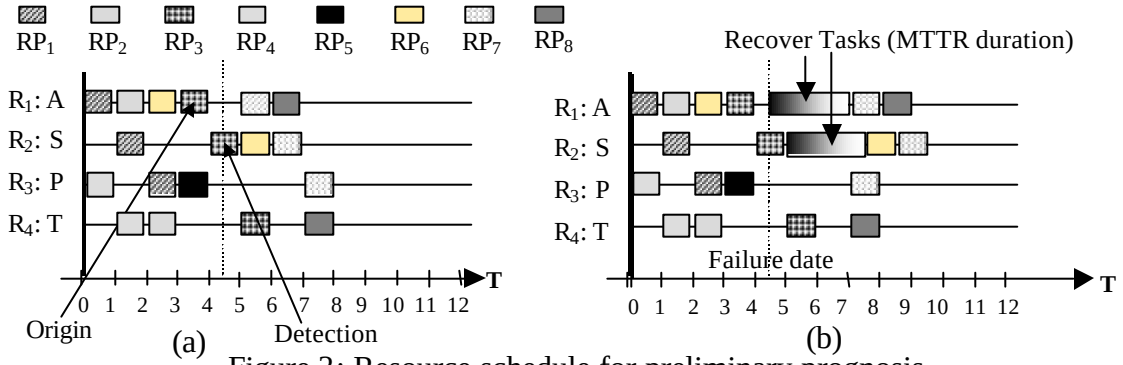

Figure 3: Resource schedule for preliminary prognosis.

We suppose that (1) a resource cannot simultaneously execute two tasks, but that task may need two resources simultaneously. (2) A Soldering operation is always preceded by an Assembling operation. This constraint is also described in FEMA (Failure Mode and Effect Analysis). (3) The processing time of each task is equal to 1. (4) The MTTR of each resource is 2.5. (5) A faulty part is taken off of manufacturing system.

5.2 Prognosis function

When a symptom is detected and diagnosed, the prognosis function initiates a preliminary prognosis. Based on MTTR, an unavailable period (to execute the recovery task) is immediately inserted in the R_1 schedule. The FEMA of R_1 shows that the resource R_2 will be affected by the consequences of the failure, so the same unavailability period is inserted in its scheduling.

The new ready dates of RP_7 and RP_8 are:
$r(RP_{7,1}) = 4$; $r'(RP_{7,1}) = 2.5 + 4.5 = 7$; $r''(RP_{7,1}) = Sup(7, 4) = 7$
Sequential_Margin $(RP_{7,1}) = 7 - 7 - 1 = -1$
Similarly, we compute:
$r''(RP_{8,1}) = 8$, $r''(RP_{7,2}) = 8.5$.
Sequential_Margin $(RP_{8,1}) = 0$, Sequential_Margin $(RP_{7,2}) = -1.5$.

The delay (Delay$(RP_7) = 1.5$ and Delay$(RP_8) = 1$) will be transmitted to the decision function.

In our example, the detection function pointed out the RP_3 as a faulty part and the diagnosis function pointed out R_1 as the origin of the failure (the tool used was broken). On R_1, no parts were assembled after the RP_3 processing.. The history of the treatments shows that RP_1 does not lead to the same symptom during the soldering operation. The fault happened between the tasks $RP_{1,1}$ and $RP_{3,1}$. A binary search algorithm used to minimize the number of checked parts, identifies the list of faulty products (RP_2, RP_3, RP_6), which will be taken off the manufacturing system.

At this step of the failure processing, the actual state of the workshop is known (the failure and its consequences have been identified). A recovery solution must be determined. We do not have knowledge of generic works on this aspect. In fact, the elaboration of a recovery sequence is strongly related to the workshop, so only case studies can be done. In our works, we consider that this recovery sequence is given by the decision function. We call it the "Technical Recovery Sequence" (TRS). The following section deals with the evaluation of the consequence of the TRS on the schedule of the resources.

5.3 Recovery evaluation

The evaluation procedures consist in finding out the system perturbations related to the TRS insertion in the production schedule. The Figure 4(a) shows a graph and the links between the tasks not executed yet. The new schedule (tasks $RP_{6,2}$ and $RP_{3,4}$ have been deleted according to the prognosis results) computed by the proposed method is given in Figure 4 (b).

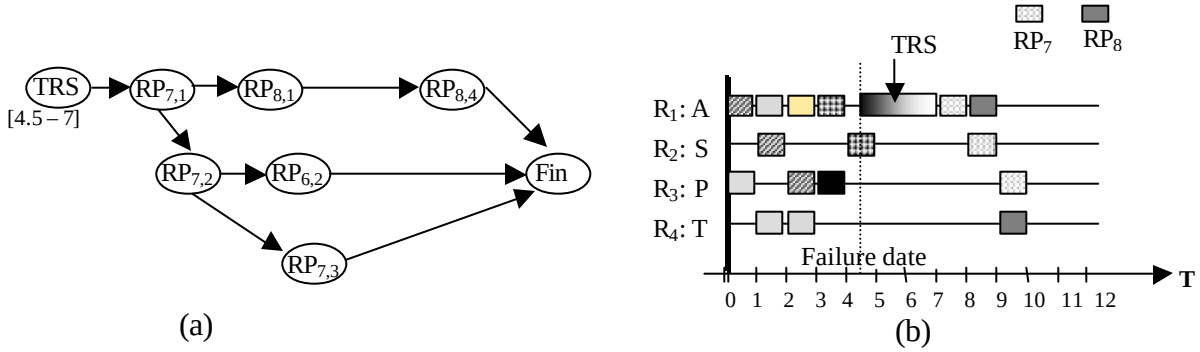

Figure 4: Constraints graph and new schedule after insertion of TRS

The evaluation procedures return the new scheduling and the delay of each task (part RP_8 will have a delay of 2). With this information, the decision function has to negotiate (if necessary) with the upper level a delay on some activities and/or the products that could be taken off the manufacturing process. This negotiation will finish when an acceptable RRS is founded.

CONCLUSIONS - PERSPECTIVES

Two monitoring/supervision functions have been presented. These two functions use well-known concepts and efficient algorithms in order to compute data used to determine an admissible corrective sequence when a process failure disturbs the operation of the workshop.
The prognosis function relies on the FMEA of failures and on the concepts of sequential margin. The result constitutes a set of constraints on tasks scheduling and allocation that have to be modified because of the failure occurrence. The recovery evaluation quantifies the effect of the recovery-sequence introduction within the forward-looking schedule. The new schedule (starting and finishing time of tasks) constitutes a flow of information that sent back to the decision function that can accept the solution or has to compute another recovery sequence.

This paper presents two basics aspects of supervision related to the decision-making process needed in order to elaborate a recovery sequence including timing considerations. Future works in this domain have now to focus on the decision function itself. This is a complex function because the human operator must be considered as an unavoidable actor to determine a recovery sequence and to accept the consequence of its introduction. In fact, in most of cases, the supervision system must be a decision support system rather than an automated recovery system. In this way, a complete supervision system would be specified and could be implemented on real applications.

REFERENCES

Artigues C., Roubellat F. (2000) A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes, European journal of operational research 127 pp.297-316

Berruet P., Toguyeni A., Elkhattabi S., Craye E. (1998) "Toward an implementation of Recovery procedures for FMS Supervision", IFAC INCOM'98, Nancy, June, organised session, Vol. 3, pp. 371-376.

Berruet P., Toguyeni A., Elkhattabi S., Craye E. (1999) Tolerance evaluation of flexible manufacturing architectures, Journal of Intelligent Manufacturing, Vol. 10, N°6, December, pp. 1-14.

Chaillet A. (1995) Multi-model approach for real-time control and monitoring of complex discrete events systems, PhD thesis, University of TOULOUSE III, December.

Combacau M., Berruet P., Charbonnaud, Khatab A. and Zamai E. (2000) Supervision and monitoring of production systems In: Proc. MCPL'2000, Grenoble, 4-6 juillet.

Combacau M., Zamaï E., and Chaillet-Subias A. (1998), Monitoring Strategies as Control Structure of Monitoring Architectures Based on Discrete Event Systems. Computational Engineering in System Applications, Nabeul-Hammamet, Tunisia, 1-4 April.

Combacau M. (1991) Control and monitoring of complex discrete events systems: Application to flexible manufacturing systems, PhD thesis, University of TOULOUSE III, December.

Dangoumau N. Elkhattabi S. Craye (1999), Design and Management of Flexible Manufacturing System's modes, ACS'99, Szczecin (POLAND), November 18-19, pp. 417-422.

Laprie J.C. (1992), Dependability basic concepts and terminologies, Springer Verlag edition, ISBN: 3-211-82296-8

Villemur A. (1998) Sûreté de fonctionnement des systèmes industriels, Ed Eyrolles, ISSN 0399-4198.

Zamaï E, Combacau M and Chaillet-Subias A (1998), Models and Strategies for Monitoring of Flexible Manufacturing Systems, 9th Symposium on Information Control in Manufacturing, Nancy-Metz, France, June.