
SUPERVISION AND CONTROL OF HETERARCHICAL DISCRETE EVENT SYSTEMS

Marcos R. Da Silveira*
dasilveira01@yahoo.fr

Michel Combacau*
combacau@laas.fr

*LAAS-CNRS - 7 Avenue du colonel Roche 31077 TOULOUSE - cedex 04 France

ABSTRACT

This paper presents a method for distributing centralized control models of discrete events systems (DES). The contribution of our approach is to offer a systematic way to decompose centralised models in order to obtain a modular representation of the production process. We observe as an advantage of this method that the main model properties are preserved and that the knowledge acquired during the modelling process is still valuable for the new structure. We emphasise that redundant information can be introduced to the system to increase local autonomy and to help the local fault detection and identification functions. Another important aspect is that the distributed models (sub-models) can be modified in order to increase their flexibility and to reach a more realistic behaviour representation. A set of terminologies and definitions related to the Control and Supervision domain are introduced in this paper to assist the understanding of our work.

KEYWORDS: Supervision, monitoring, distributed models, discrete event systems, architectures.

RESUMO

Este artigo apresenta uma metodologia de distribuição de modelos de controle e de supervisão de sistemas a eventos discretos (SED). A contribuição da nossa proposta é de proporcionar uma forma sistemática de decomposição de modelos

centralizados com a finalidade de obter uma representação modular da produção, preservando suas principais propriedades assim como o conhecimento adquirido durante a fase de concepção do modelo. Uma das características da nossa proposta é a possibilidade de introduzir informações redundantes nos sub-modelos afim de aumentar a autonomia local e auxiliar as funções de tratamento de falhas. Ressaltamos também que a distribuição do modelo centralizado permite adaptações locais que aumentam sua flexibilidade e precisam seu comportamento, aproximando-o do sistema físico considerado. Neste artigo, nos apresentamos também alguns termos e definições relativos à área de Controle e Supervisão de SED que serviram de base aos nossos trabalhos.

PALAVRAS-CHAVE: Supervisão, monitoramento, sistemas distribuídos, sistemas a eventos discretos, redes de Petri.

1 INTRODUCTION

Production systems concepts were firstly proposed to satisfy the increasing demand for ready-to-consume products. These concepts have been enriched to meet the new needs of clients (diversity, quality, etc.) and to reduce the production costs. Many researches contribute to improve specific criteria of the production process (for example, organizational strategies, technologies, etc.), but none has completely satisfied this process. The evolution of computer science and its applications to industry is a fundamental engine for improving their performance. On the other hand, systems have become more complex, modeling and integrating them efficiently is harder (sometimes impossible) to implement. The new challenge is to monitor, supervise and control the production process precisely in order to measure all a decision's

Artigo submetido em 13/12/02

1a. Revisão em 04/06/03

2a. Revisão em 31/05/04

Aceito sob recomendação do Ed. Assoc. Prof. Takashi Yoneyama

consequences before it drives the system to abnormal functioning. In practice, when model checking is not efficient, latent errors can lead the system to an uncontrollable state or breakdown state.

The most referenced research on this subject is the work of (Ramadge and Wonham, 1987) which introduces the supervisory theory with controllable (which can be turned on and off by an external agent) and uncontrollable events. Many approaches enriched this theory (Lin and Wonham, 1990; Cury and Krogh, 1999; Yoo and Lafortune, 2000). Similarly, some researchers centre their studies on the system decomposition (Dalesio and al., 1994; Du and Zhou, 1999; He et al, 2000) to obtain a modular system representation. In a sense, (Combacau, 1991) suggests separating the control model into two (*control* and *resource*) different models and the supervisory functions. Together they compose a hierarchical architecture module. However, recent researches consider diverse types of module organization, of which holonic, bionic, fractal and multi-agents architectures are the most common (Dilts and al., 1991; Leitão and Restivo, 1999).

Our research is in line with the decomposition problem, i.e. we propose a method for distributing centralized control models based on Petri nets' properties in order to obtain a modular representation of control models (sub-models) with a partial redundancy of data.

These studies are based on the concepts of modularity and internal organization introduced previously by (Combacau, 1990), (Chaillet, 1995) and (Zamai, 1997).

The second section presents some terminology and the definitions used in this paper and in our research work. The main results from previous work developed at the LAAS are presented in section three. Section four details the systematic procedure for distributing the centralized model and gives the mathematical formalism to support this distribution. The communication and the data update problems are introduced in section five. A communication protocol is proposed and described in section six. Finally, an illustrative example is detailed in section seven.

2 TERMINOLOGIES OF REFERENCE

This section presents some usual definitions of important terminologies in the Supervision and Control domain. There is no consensus about the terminologies definition presented in this section. Hence, the objective of the paper not to propose new terminologies, instead, it presents definitions used to describe ours researches. Different authors contribute to this section, but a concentrate of definitions can be found in (Combacau et al., 2000) and (Laprie, 1992). Control, monitoring and supervision are first defined and then the definition

of other terms used in this paper is given.

Control: triggers the execution of a set of operations by giving orders to the process actuators, which may be:

- A set of operations corresponding to the manufacturing sequence of the product.
- A set of operations executed in order to restore the process functionality offered during normal execution.
- Actions with a high priority level applied in order to protect the shop workers and to prevent catastrophic developments.
- Some checking, tuning or cleaning operations executed in order to maintain the process in an operational state.

This means that our definition of control includes all the functions actually acting on the process.

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

Supervision: computes and sets the parameters of the control sequence to be executed according to the state of the control system and to the state of the process. This includes normal and abnormal operations. During normal operation, supervision takes the decisions to raise the indecision in the control system (real-time scheduling, optimization, control sets and switching from one control law to another). When a process failure occurs, supervision takes all the decisions necessary to allow the system to resume normal operation (rescheduling, recovery actions, emergency procedures, etc.). It should be noted that supervision takes place in a hierarchical structure (of at least two levels). At the lowest level only the control and monitoring functions are generally implemented - no real decisions have to be taken.

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 which does not exactly match the specifications of the physical system. Logically, an error is the 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 no longer corresponds to the nominal specifications.

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

Symptom: Event or data by which the detection system identifies an abnormal process operation. 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 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 the 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 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 the state space of the system, it traces the events observed in the control/supervision model to update the state of 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 (resource unavailable) or induced (faulty parts in the workshop).

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

Recovery (C, S): acts both 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.

3 THE CONTROL-MONITORING MODULE

Rapid technological advances in computer science have offered a wide range of possibilities to design control architectures. There are many basic architecture propositions, but centralized, hierarchical and "heterarchical" (as well as bionic, holonic, fractal, etc.) are the most commonly accepted. In this paper, we will use the term *heterarchical* to indicate an architecture that does not have the same characteristics as centralized or hierarchical architecture. The heterarchical architecture is adopted in order to pursue full local autonomy in which the global information is minimized or eliminated. This implies that: (1) external higher levels of control can change according to the activity to be coordinated. In this case one module can compose, for instance, two different hierarchical structures. (2) The communication between entities will not, necessarily, have a master/slave relationship, for example, they can co-operate, negotiate or dynamically change roles from master to slave and vice-versa. (3) We can introduce a new entity or modify the existing ones without significant structural changes.

These flexibilities introduced by the heterarchical architecture imply a more complex relationship between modules, risking failure situations hardly detected during the design process. We observe too that these flexibilities increase the difficulty of solving failure situations if their origins are not local, as the modules often do not know to where the faulty treatment functions must be propagated. This second problem does not exist either in hierarchical nor in centralized architecture, thus, no other reference, as far as the authors know, takes into account that new situation. Part of our contribution is to prevent that situation including local information about the modules' relationship.

This research is based on the modular organization presented

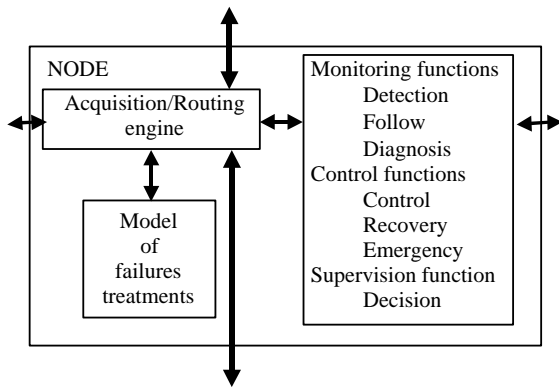


Figure 1: A generic module for control, supervision and monitoring

in (Combacau and al., 1998), called Control-Monitoring architecture. In their approach only resource failures are taken into account. The control and monitoring system is considered error free. When a resource failure occurs, the corrective actions to be executed are performed according to the activity state (function in execution, resource used, kind of manufactured products and production strategy specified by the user). Failure processing is not limited to the classical sequence (detection, diagnosis, decision and recovery).

The Acquisition/Routing block manages all these functions, as shown in Figure 1. This block is based on an algorithm, which directs incoming messages to the most suitable functions, according to the nature of the data and 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).

Two complementary tools are used to specify this approach. Petri nets with Objects are used for modeling control, recovery and emergency sequences and failure treatment. An extended entity relationship model provides a process representation (called Information System) in which data, which is not easy to model by means of Petri nets (time notions, histories, data flow, etc.) can be found.

In a sense, our research considers the topics pointed out previously to build a systematic procedure for distributing a centralized model of supervision and control. It is based on the Petri nets (PN), as described in (Valette and al., 1999), and shows the advantage that each part of the control process has, at least, the same properties as the whole model and at most all good properties (boundedness, liveness, home state). This procedure is founded on linear PN invariants theory and the results of applying it is a set of sub-models, each one describing the behavior of one resource (or a set of them) and its interactions with other resources. Some sub-models have re-

dundant information about the systems, because the relation between resources must be represented in all entities that use its services. The method is detailed in the following sections.

4 THE DISTRIBUTION PROCEDURE

This section describes the method used to distribute a centralized model. In this proposition, the (centralized) model is split into several sub-models. Each sub-model must represent one resource (or a finite group of resources): (i) all possible evolutions from one state to another; (ii) all pre-conditions and consequences of each related evolution; and (iii) all possible relations with other models (for example, co-operation, competitiveness, mutual exclusion, etc.). The main constraint is that all resources should have their activities represented in at least one sub-model.

In this work, the description of the production relies on two main models: the control model and the resource model.

The control model represents all the operational constraints related to the transformation of raw parts to finished ones.

The resource model represents the physical and functional characteristics of the process (mutual exclusions, sequence constraints, co-operation activity, etc.).

According to this information, an entity composed of a control sub-model, a process sub-model and all the functions of supervision and control, constitutes a module and is associated to a given resource.

A module must have all the necessary information to execute any modeled activities. Therefore, parts of the Petri net, modeling co-operation between resources, have to be represented in all the modules associated to these resources. In other words, there is a partial redundancy of global information because some activities are described in different modules. The scope of this redundancy can be enlarged to incorporate important information in order to increase the reliability of the system. In this paper, all sequences of activities that can drive the system to a catastrophic situation are considered as important information. The duplication of information follows strict mathematical rules as described in the next section.

4.1 The formal description of the distribution procedure

Some basic concepts of Petri nets are presented to notation used to describe the distribution method (Valette and al., 1999).

A Petri net is a four-tuple $PN=(P, T, I, O)$ such that: P is

a finite and non-empty set of places, T is a finite and non-empty set of transitions, I is the input function and O is the output function.

Definition 1 Let M be the marking vector of a PN , t be a transition of PN and K be a constant. The good properties are:

K-Bounded: $\forall M \in A(PN, M_0) M(p) \leq K$

where $A(PN, M_0)$ is the accessibility tree of PN with M_0

Liveness: $\forall t \in T$ and $\forall M \in A(PN, M_0)$, $\exists M' \in A(PN, M_0)$ and $\exists s | M \xrightarrow{st} M'$

Home state: $\forall M \in A(PN, M_0) \exists s M \xrightarrow{s} M_0$

The linear invariants are classed as place invariant (p-invariant) and transition invariant (t-invariant). We are interested on positive linear invariant because it represents a real evolution of the system.

Definition 2 Let a finite PN system with $P = \{p_1, p_2, \dots, p_m\}$ and $T = \{t_1, t_2, \dots, t_n\}$ be given. A vector $v \in Z^m$ is called a positive p-invariant of a given PN, iff:

$$v^T \bullet C = 0 \wedge v \neq 0 \wedge v_i \geq 0 \quad \forall i = 1, 2, \dots, m$$

where C is the incidence matrix.

A vector $s \in Z^n$ is called a positive t-invariant of the given PN iff:

$$C \bullet s = 0 \wedge s \neq 0 \wedge s_i \geq 0 \quad \forall i = 1, 2, \dots, n$$

The systematic distribution proposed is founded on the positive p-invariant (pp-invariant) theory and generators, referenced in (Kruckenberg, 1987). In this paper we will focus only on the third level of generators, where each positive invariant is a positive linear combination of generators, which are minimal standardized invariants.

The places that compose a generator and the input/output transitions associated with them establish the minimal set of places/transitions of a distributed sub-model. The distributed process does not explicit the arc (input/output functions) components from the PN, because we suppose that a place and a transition copied into a module implies that every arc connecting them is also copied into it. Considering the resource model, the places of a generator represent some (or all) possible states of one resource. Other components can be included to give more autonomy and/or reliability to the module. They belong to two different classes:

- The ones directly connected (by one arc) to the transitions of the minimal set.
- The result of a reduced PN.

This systematic distribution is ordered in four steps:

- Step one, the places that compose a generator are identified.

Definition 3 A set of places of one generator v_k .

$$\left\langle \begin{array}{l} v_k | v_k^T \bullet C = 0 \\ Q_k = \{p \in P | v_k(p) > 0\} \end{array} \right\rangle \quad (1)$$

$$Q = \bigcup_{i=1}^N Q_i, N = \text{number of generators} \quad (2)$$

- Second, the input and/or output transitions of these places are grouped as a set of local transitions.

Definition 4 A set of input/output transitions of one generator

$$S_k = \{t_j \in T | I(t_j, p) + O(t_j, p) \neq 0 \forall p \in Q_k\} \quad (3)$$

$$S = \bigcup_{i=1}^N S_i, N = \text{number of generators} \quad (4)$$

- Third, all input and/or output places of these transitions, except the ones identified in the first step, are grouped as a set of external places.

Definition 5 The external places.

$$\varphi_k = \{p \in P \setminus Q_k | \exists t_j \in S_k I(t_j, p) + O(t_j, p) \neq 0\} \quad (5)$$

$$\varphi = \bigcup_{i=1}^N \varphi_i, N = \text{number of generators} \quad (6)$$

- Finally, all other places and transitions external to the considered net are reduced (structural reduction) to compose the *reduced PN* (PN^-). The structural reduction uses classical reduction algorithms. The PN^- is coupled with the sub-model as an abstraction of other modules' behaviors. Some constraints imposed by the user (for instance, duplication of critical information) can change the final PN^- .

Other tools are used to optimize the model distribution, as well as the predicate/transition PN. In this case, all activities

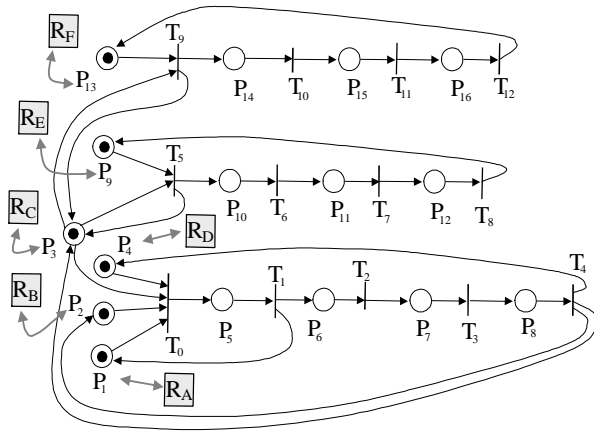


Figure 3: The centralized model

native for keeping data coherence. The pre-reservation becomes a real reservation when all the resources associated to the activity are available (Ready), in any other case the activity is refused.

- The protocol immediately refuses any incoming request while the resource is Busy.
- Different control laws may require the same resource simultaneously. Each resource (except those associated with a CDM) has its own local rule to treat this kind of conflict (for example, the requests are queued).

The distribution method and the communication specification can be visualized in a practical example in the next section.

7 AN ILLUSTRATIVE EXAMPLE

In this section, we use an illustrative example to explain how the distribution method and the communication protocol can be applied to the manufacturing systems modeling.

We will use the centralized model presented in Figure 3 as our study case. The physical system description is out of the scope of this paper; we are mainly interested in the model's characteristics.

Observing the centralized model (Figure 3), six resources can be identified $\{R_A, R_B, R_C, R_D, R_E, R_F\}$, their behavior (reachable places and transitions) is described by the Petri nets model, where the places (states) $\{p_1, p_2, p_3, p_4, p_9, p_{13}\}$ correspond respectively to the *Ready* state of each resource.

Thus, the four steps necessary to apply distribution method are:

Generators	Transitions
$R_A = Q_1 = \{p_1, p_5\}$	$S_1 = \{t_0, t_1\}$
$R_E = Q_2 = \{p_2, p_5, p_6, p_7, p_8\}$	$S_2 = \{t_0, t_1, t_2, t_3, t_4\}$
$R_C = Q_3 = \{p_3, p_5, p_6, p_7, p_8\}$	$S_3 = \{t_0, t_1, t_2, t_3, t_4, t_5, t_9\}$
$R_D = Q_4 = \{p_4, p_5, p_6, p_7, p_8\}$	$S_4 = \{t_0, t_1, t_2, t_3, t_4\}$
$R_E = Q_5 = \{p_9, p_{10}, p_{11}, p_{12}\}$	$S_5 = \{t_5, t_6, t_7, t_8\}$
$R_F = Q_6 = \{p_{13}, p_{14}, p_{15}, p_{16}\}$	$S_6 = \{t_9, t_{10}, t_{11}, t_{12}\}$
External places	
$\Phi_1 = \{p_2, p_3, p_4\}$	
$\phi_2 = \{p_2, p_3, p_4\}$	
$\phi_3 = \{p_1, p_2, p_4, p_9, p_{10}, p_{13}, p_{14}\}$	
$\phi_4 = \{p_1, p_2, p_3\}$	
$\phi_5 = \{p_3\}$	
$\phi_6 = \{p_3\}$	

Figure 4: The distribution methodology

- The generators set Q_N is calculated and presented in Figure 4-A.
- All input/output transitions from each place of Q_N are identified, as shown in Figure 4-B.
- The external places ϕ_N are grouped, Figure 4-C shows the set of external places obtained without force the introduction of redundant information.
- The reduced Petri net is computed for each sub-model composed by $(Q_N \times S_N \times \phi_N)$.

We will consider resource R_C as an instance to detail the distribution methodology. This resource is described by the generator Q_3 , composed of $\{p_3, p_5, p_6, p_7, p_8\}$. Some of these states appear in other generator state groups like in Q_1 , Q_2 and Q_4 , this implies that a co-operation exists between them. The input/output transitions of Q_3 are $S_3 = \{t_0, t_1, t_2, t_3, t_4, t_5, t_9\}$. The external places are grouped in $\phi_3 = \{p_1, p_2, p_4, p_9, p_{10}, p_{13}, p_{14}\}$ and finally, the reduced Petri net, represented by $\{PN_1^-, PN_2^-\}$, the dashed elements of Figure 5.

Figure 6 presents a request sequence composed of 3 conflicting requests $\{Rqt_1, Rqt_2, Rqt_3\}$. The Rqt_1 requires the resources of modules 3 and 5 (R_C, R_E); the Rqt_2 requires the resources of modules 1, 2, 3 and 4 (R_A, R_B, R_C, R_D); and the Rqt_3 requires the resources of modules 3 and 6 (R_C, R_F).

This example is a special case where co-operation and competitiveness are taken into account. Therefore, for each activity, the control system points out a CDM that will solve any conflicts between requests. A group can have the same CDM (like T_0, T_5, T_9 has module 3 as their CDM). We note that the inter-module's protocol role is to maintain the coherence between distributed data. Data coherence consists in verifying,

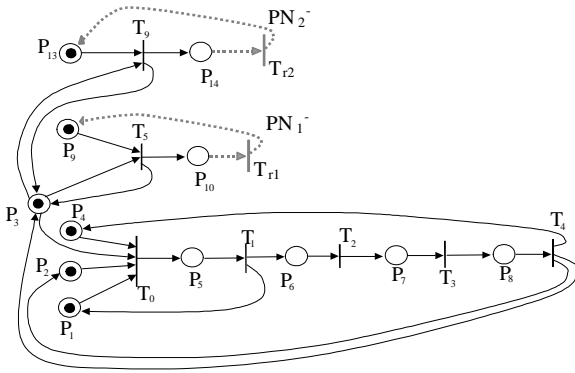


Figure 5: The model of module 3

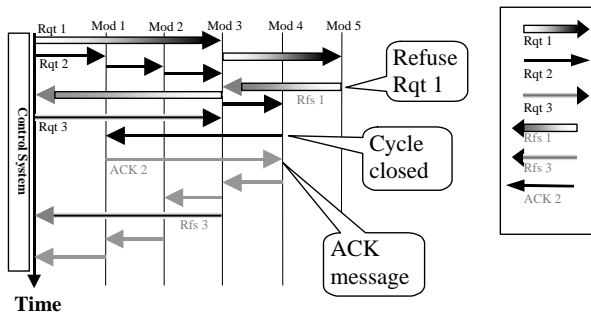


Figure 6: The request sequence treatment

in every related module, if the state of all resources associated with the activity corresponds to their state indicated in the CDM model. (In this example, it verifies if they are in the *Ready* state).

According to Figure 6 three activities are required at different dates. The first request "Rqt₁" arrives in module 3 (the CDM of this activity); it pre-reserves R_C. Suppose that the model of module 3 had not yet been updated and the resource of module 5 (R_E) is *Busy*, when Rqt₁ propagates up to module 5 R_E is not *Ready* and the request is refused, so the REFUSE signal (Rfs₁) is sent back to module 3. Based on this information the model is updated and the control system is notified. In this example, Rqt₂ and Rqt₃ require R_C while it is pre-reserved, then the requests go to the *Wait* state. They change state to *Prsv* or to *Abort* state when the resource required state changes to *Ready* or to *Busy* state, respectively.

The refused requests return to the control model, which decides if the request will be deleted or modified and sent again (back) to the resource model

8 CONCLUSIONS

The control and supervision models considered in this paper have at least two hierarchical levels: control model and supervision model. The control model is based on the physical system characteristics and on the product and production properties. In this work, we consider that the description of these properties is separated into two different models, the resource model and the control model. The resource model is established by the bottom-up method and is composed of all the possibilities of using the system resources. The control model takes production constraints, product specification, etc., into account in order to determine a sequence of actions that can use the resources efficiently to obtain the final product. We note that the control model can be applied only in normal functioning, any anomaly is solved by the upper level (supervision). The supervision and monitoring functions are elaborated based on the resource model; their objective is to identify and solve all eventual system failures. Thus, resource model distribution can affect all other models (or functions) based on it. In this sense, we proposed a formal technique based on the Petri net Theory in order to systematically distribute the centralized resource model and associate these sub-models with modules structured in a heterarchical way.

This procedure is based on some properties of Petri nets as well as the linear invariant theory. The distribution of a whole resource model results in a set of sub-models that describes the behavior of a resource (or a set of them) and the interactions with other resources. The distributed models maintain the same properties as the centralized one (good properties). The synchronization of actions executed by different modules is guaranteed by the intra-modules protocol that supports the communication between them and maintains the coherence of distributed data. The perspective of this work concerns the quantification of the redundancy inserted by the distribution methodology, the evaluation of consequence for applying this method on other models and the consideration of time on the PN model.

REFERENCES

- Chaillet A. (1995). *Approche multi modèles pour la commande et la surveillance en temps réel des systèmes à événements discrets*. Ph.D. Thesis of Université Paul Sabatier de Toulouse, Rapport LAAS No. 95521.
- Combacau M. (1991). *Commande et surveillance de systèmes à événements discrets complexes: application aux ateliers flexibles*. Ph.D. Thesis of Université Paul Sabatier de Toulouse, Rapport LAAS No. 91470.
- Combacau M., Berruet P., Charbonnaud, Khatab A. Zamai E. (2000) Supervision and monitoring of production sys-

tems In: Proc. MCPL'2000, Grenoble, 4-6 July.

- Combacau M., Zamaï E., 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.
- Cury J.E.R., Krogh B. H. (1999). Robustness of supervisors for discrete-event systems. *IEEE Transactions on Automatic Control*, Vol. 44, No. 2, pp. 376 – 379.
- D. W. He, B. Strege, H. Tolle, A. Kusiak (2000). “Decomposition in automatic generation of Petri nets for manufacturing system control and scheduling”. *International Journal of Production Research*, vol. 38, No. 6, pp. 1437 – 1457.
- Da Silveira M., Combacau M., Subias A. (2002). From centralized to distributed models: A systematic procedure based on Petri nets. SMC - IEEE International Conference on Systems, Man and Cybernetics, octobre 2002, Hammamet, Tunisia.
- Dalesio L. R., Hill J. O., Kraimer M., Lewis S., Murray D., Hunt S., Watson W., Clausen M. and Dalesio J. (1994). The experimental physics and industrial control system architecture: past, present, and future. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 352, Issues 1-2, 15 December 1994, pp. 179-184.
- Dilts D. M., Boyd N. P., Whorms H. H. (1991). The evolution of Control Architectures for automated Manufacturing Systems. *Journal of Manufacturing Systems*, vol 10, pp. 79 - 93.
- Du X. H., Zhou C. (1999). Message-oriented decomposition for supervisory control in manufacturing systems. *Robotics and Computer Integrated Manufacturing*, No. 15, pp. 441 - 452.
- Ghosh S. (2001). Understanding complex, real-world systems through asynchronous, distributed decision-making algorithms. *Journal of Systems and Software*, No. 58, pp. 153-167.
- Kruckenbergh F., Jaxy M. (1987). Mathematical Methods for Calculating Invariants in Petri Nets. Advances in Petri Nets, Springer, LNCS 266, ed. G. Rozenberg, pp. 104-131.
- Laprie J. C. (1992). Dependability basic concepts and terminologies, Springer Verlag edition, ISBN: 3-211-82296-8
- Leitão P., Restivo F. (2000). A Framework for Distributed Manufacturing Applications. Advanced Summer Institute International Conference, pp.75-80.
- Lin F., Wonham W. M. (1990). Decentralized supervisory control of discrete event systems with partial observation. *IEEE transaction in Automatic Control*, No. 35, pp. 1330 - 1337.
- Ramadge P. J. G., Wonham W. M. (1987). “Supervisory control of a class of discrete event processes”. *SIAM Journal of Control Optimisation*, No. 25, pp. 206 - 230
- Valette R., Pradin-Chézalviel B., Girault F. (1999). “An introduction to Petri net theory”. Fuzziness in Petri nets Eds. J. Cardoso, H. Camargo, Studies in Fuzziness and soft computing. Vol. 22, Physica Verlag, ISBN 3-7908-1158-0, pp.3-24
- Yoo T., Lafortune S. (2000). A general architecture for decentralized supervisory control of discrete-event systems. Workshop on Discrete Event Systems, pp. 111 – 118, Belgium.
- Zamaï E. (1997). *Architecture de surveillance-commande pour les systèmes à événements discrets complexes*. Ph.D. Thesis of Université Paul Sabatier de Toulouse, Rapport LAAS No. 97361.