Implementation of diagnosis approach for Discrete Event Systems

Philippot Alexandre, Pascale Marangé, Véronique Carré-Ménétrier, Bernard Riera

Implementation of diagnosis approach for Discrete Event Systems

A. Philippot*, P. Marangé, V. Carré-Ménétrier*, B. Riera***

**Centre de Recherche en STIC (CReSTIC), University of Reims Champagne-Ardenne (URCA) Reims, France, (Tel: 03326918616; e-mail: alexandre.philippot@univ-reims.fr).*
***Centre de Recherche en Automatique de Nancy (CRAN), UMR 7039 – Nancy University, CNRS, Vandœuvre-lès-Nancy, France, (pascale.marange@cran.uhp-nancy.fr)*

Abstract: This paper presents an approach of diagnosis for manufacturing system considered as Discrete Event Systems. It uses plant decomposition and a decentralized diagnosis structure to reduce the combinatory explosion found in centralized structures. The local behavior is extracted using decentralized plant modeling. It is from this behavior that possible faults are identified to construct abnormal behavior models. The approach is illustrated around a manufacturing benchmark.

Keywords: Discrete Events Systems, Modeling, Diagnosis, Manufacturing systems.

1. INTRODUCTION

Manufacturing systems have become more and more complex owing to technological evolution. Moreover, each maintenance intervention has an important cost, as false removal for example in case of uncertainty. This complexity and the desire for improved availability, reliability and dependability require the development of systematic approaches of diagnosis to detect and isolate a fault.

In literature, many approaches are developed and are based on the observability of the system (Sampath, 1995), (Qiu, 2005), (Cordier et al., 2007). However, manufacturing systems are often described as dynamic systems with discrete states which change only by discrete events. They are considered as Discrete Event Systems (DES) (Cassandras and Lafortune, 1999). Consequently, the observable information present is poor because of the discrete state space. Its observation alone does not often allow to detect a fault occurrence and to isolate the responsible element. Several approaches have been developed to solve the Fault Detection and Isolation (FDI) problem (Sampath, 1995), (Su and Wonham, 2000), (Cordier et al., 2007). Most of them are based on the use of a model in order to specify the normal and/or faulty behaviors of the system. This model defines how system states change due to event occurrences. The goal is to compare the system model state with the real one, based on the system observations. A failure is detected when the two states do not match.

Three structures of diagnosis decision are presented in literature. In centralized structure, the global diagnosis decision is taken by one centralized diagnoser based on one system model (Sampath, 1995). Decentralized structure comprises one system model associated with several local diagnosers (Wang et al., 2005). A very limited communication is permitted through a coordinator to solve the ambiguity or indecision problem between local diagnosers' decisions. Distributed structure uses several local models associated with several local diagnosers. A pure concurrent communication among local diagnosers is necessary to realize a global diagnosis decision (Qiu, 2005), (Cordier et al., 2007).

The main disadvantage of centralized approaches for DES is the state explosion. The approaches with decentralized structure constitute a solution to this drawback. But so far, designing a decentralized diagnoser requires the existence of centralized model which entails again the state explosion problem (Su and Wonham, 2000). In (Cordier et al., 2007), the authors propose an approach to realize the decentralized diagnosis without using a global model. This approach is based on the use of a merging procedure which exploits the independence property between sub-models. The major advantage of distributed diagnosis approaches is that there is no need for a centralized plant model. The communication between local diagnosers is used to solve the indecision problem. However, their main disadvantage is the need to establish a performing protocol of communication, which scales well with the system complexity.

In this paper a decentralized diagnosis approach is proposed to realize the diagnosis of DES, specifically manufacturing systems with discrete sensors and actuators. Manufacturing systems can be represented by several models of components called "Plant Elements" (PEs) which describe all possible evolutions. From each local component, an analysis is made to construct local models of abnormal behavior called diagnosers. Each local diagnoser returns a label to the user and the global decision is all local decisions merged.

The paper is organized as follow. In section 2, the proposed approach is presented with the different steps of modeling. Section 3 illustrates the approach in several aspects. Conclusion and prospects close the paper in section 4.

2. DECENTRALIZED DIAGNOSIS

In industrial processes, a manufacturing system is a functional chain composed of a controller which sends

commands to a plant and receives sensor values (Fig. 1). Plant represents the mechanical part whereas controller is the logical part which describes the desired behavior. This exchange between controller and plant represents the only observable information available on line. As a diagnoser is defined as an observer of the system, it is necessary to use this information to rebuild behaviors through models.

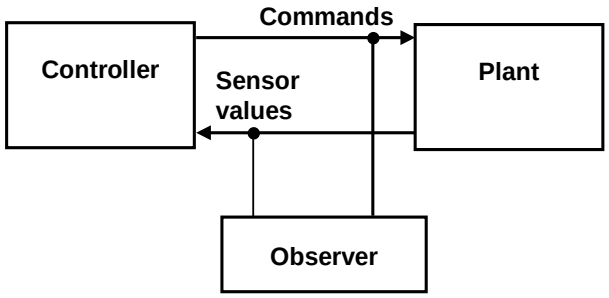

Fig. 1. Functional chain.

From the introduction, a centralized approach appears as unthinkable for complex systems. However, the difficulty of a decentralized approach is to determine the level of modular decomposition in a generic way. A manufacturing system is composed of mechanical components, called Plant Elements (PEs), which are actuators with associated sensors. Consequently, the proposed approach builds local diagnosers from each PE. Figure 2 shows the different steps to obtain local diagnosers. Firstly, from the real plant, PEs are defined from a library of models. Secondly, an expert analysis is made to identify the possible abnormal situations, and then to obtain local diagnosers.

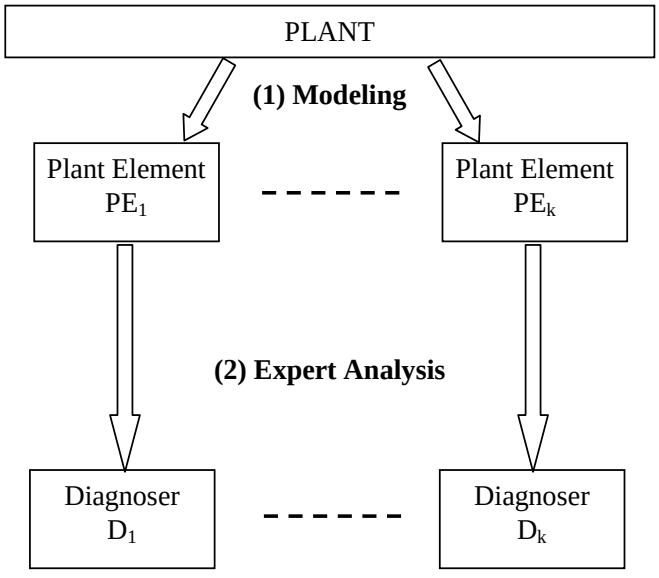

Fig. 2. Steps of the proposed approach.

2.1 Plant Elements

Plant is divided into several components (actuator with associated sensors) which can be modeled in Plant Element. Each model G^i and corresponding language L_i describes the logical and untimed behavior of the monitored system. It is a Moore automaton: M = (Σ, X, Y, δ, h) where Σ is the set of finite events, X is the set of states, Y is the output space, δ: Σ x $X \rightarrow X$ is the state transition function. $\delta(\alpha, x)$ gives the set of possible next states if α occurs at x. h: Σ x $X \rightarrow Y$ is the output function. $H(\alpha, x)$ is the observed output when α occurs at x.

In (Balemi et al., 1993), authors define controllable events $\Sigma_c \subseteq \Sigma$ as the control outputs (actuators) and uncontrollable events $\Sigma_u \subseteq \Sigma$ are defined as the control inputs (sensors). $\Sigma_o \subseteq \Sigma$ is the set of observable events where $\Sigma_c \subseteq \Sigma_o$. An automaton is used for each model. This automaton takes into account all the observable events. The detailed explanation of the construction of this model can be found in (Philippot et al., 2004).

This paper is not focused on the PE modeling method. PE models have been validated in previous works and a library of common PE used in manufacturing systems has been established (Philippot et al., 2007). An example is given in section 3.

2.2 Temporal information

The majority of sensors and actuators in manufacturing systems produce constrained events since state changes are usually effected by a predictable flow of materials (Boufaïed, 2003), (Holloway and Chand, 1994). To enrich Boolean models, a timed model centred on the notion of expected event sequencing and timing relationships can be used. The temporal information about events minimal and maximal durations is represented by the actuator's minimal and maximal response times.

For each state of PE a temporal relationship between input and output events is obtained thanks to the learning phase which returns a Gaussian distribution. Each prediction is constructed for observable correlated events and it describes the next events that should occur and the relative time periods in which they are expected. These pre-defined time periods are determined by experts according to system dynamic and to the desired behavior. An extrapolation of the Gaussian distribution is made to choose minimal and maximal times. Consequently, when an event α_1 occurs at the state x_1, the event α_2 should happen at the state x_2 and within the interval $[t_{min}, t_{max}]$ (Fig. 3). The use of the Gaussian distribution instead of intervals will be a prospect.

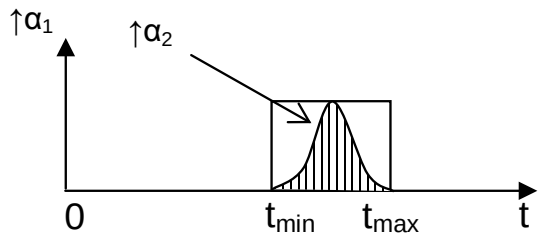

Fig. 3. Temporal information.

2.3 Diagnosers

In industrial processes, a manufacturing system is a functional chain composed of a controller which sends commands to a plant and receives sensor values. Plant represents the mechanical part whereas controller is the logical part which describes the desired behavior. This exchange between controller and plant represents the only observable information available on line. As a diagnoser is defined as an observer of the system, it is necessary to use this information to rebuild behaviors through models.

From the introduction, a centralized approach appears as unthinkable for complex systems. However, the difficulty of a decentralized approach is to determine the level of modular decomposition in a generic way. A manufacturing system is composed of mechanical components, called Plant Elements (PEs), which are actuators with associated sensors. Consequently, the proposed approach builds local diagnosers from each PE by an expert analysis to identify the possible abnormal situations from each normal state.

Consequently, a diagnoser is defined as $D_i = (X_i \cup XDF_i, \Sigma_{io}, \delta_i, x_{i0}, V_i, h_i, PF_i, l_i)$ with:

- X_i is the set of normal states of NBMi,
- XDF_i is the set of abnormal states,
- Σ_{io} is the set of observables events by the PEi,
- $\delta_i : X_i \times \Sigma_i^* \rightarrow X_i \cup XDF_i$ is the transition function,
- x_{i0} is the initial state,
- V_i is an input/output vector with $V_i(x)$ the vector of the state x,
- $h_i : X_i \cup XDF_i \rightarrow \Sigma_{io}$ is the output function where $h_i(x)$ is the observable event at the output of the state x,
- $PF_{ix} = \{PFx, \forall x \in X_i\}$ represents the set of prediction functions of the state x,
- l_x is the decision functions of the state x which can be the label N to indicate a Normal functioning or/and one fault label {Fj} or more.

If l_x = {N}, then the diagnoser can decide with certainty the non-presence of faults. If l_x = {Fj}, then the diagnoser indicates with certainty the occurrence of a fault of the type Fj. If l_x = {N, Fj}, then the diagnoser cannot decide whether a fault has occurred or not and the system is in ambiguity or indecision case. Labels are defined in subsets called partition Π_{Fj}. The knowledge of all faults comes from an expert analysis and/or documentation such as Failure Mode and Effects Analysis (FMEA), a tool used in safety, dependability and quality management (Ashley, 1993).

For an input/output vector of 3 variables (*e1 e2 e3*) with labels {N, F1, F2, F3} for a normal or an abnormal behavior respectively on *e1*, *e2* or *e3*:

From a normal state *x0*, h(*x0*) = ↑*e1* is the only one normal evolution possible to attempt normal state *x1* with label N. Consequently, possible faults which can be detected and isolated from *x0* are:

- h(*x0*) = ↑*e2*: observable fault on *e2* associated to label F1 (state *x2*),
- h(*x0*) = ↑*e3*: observable fault on *e3* associated to label F2 (state *x3*),
- *PFx0* = 1: non observable fault on *e1* corresponding to a non-satisfaction of the temporal constraint. Diagnoser goes to state *x4* with a label F3 by an internal event e*PFx0*.

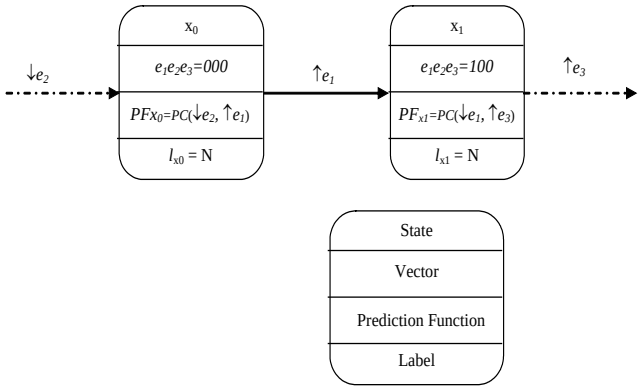

Fig. 4. Construction of a Diagnoser.

Remarks: non observable faults on *e2* and *e3* (stuck to 0) cannot be detected from *x0* and need new events. Normal behavior on *x0* cannot be guaranteed Label on this state is not only {N} but must be {N, F2, F3}.

Consequently, for a diagnoser with *p* variables, it is possible from each normal state to detect *p-1* observable faults and *1* non observable faults.

3. ILLUSTRATION ON A BENCHMATK

3.1 Presentation

All of the proposed methodology will be illustrated by means of a virtual system from the ITS PLC collection, proposed by the Portuguese company Real Games. ITS PLC collection is a set of simulation software dedicated to automation training (Riera *et al.* 2009). Demos and technical descriptions of the five virtual industrial systems are available and freely downloadable at web address www.realgames.pt. As part of the work presented in this paper, the “sorting system” is used. The objective of this system is to bring boxes of entry conveyor to exit conveyor by sorting them according to for instance their height (Fig. 5).

Fig. 5. Sorting system.

The system is instrumented using 11 sensors to determine the size of the boxes (small or large) and the entry or exit of a box in different conveyors (feeding, intermediate, evacuation) or turntable. The seven outputs of the Programmable Logic Controller (PLC) (IEC 61131-3, 1993) can activate the various conveyors and the turntable.

Inputs (Sensors): *c0*: Feeder belt exit detector, *c1*: Lower case detector, *c2*: Higher case detector, *c3* Exit detector of the entry conveyor, *c4*-*c5*: Detectors of the turntable position, *c6*: Turntable pallet detector, *c7*: Entry detector of the left exit conveyor, *c9*: Exit detector of the left exit conveyor, *c8*: Entry detector of the right exit conveyor, *c10*: Exit detector of the right exit conveyor

Outputs (Actuators): *A0*: Feeder belt, *A1*: Entry conveyor, *A2*: Turntable rollers (loading), *A3*: Turntable rollers, *A4*: Turntable, *A5*: Left exit conveyor, *A6*: Right exit conveyor

The specification used is as follows: after pressing the "start" button, the boxes are sent successively one to the left elevator and one to the right elevator. After pressing the "stop" button, boxes in transit are evacuated. In order to not overload the paper, the output conveyors are always on (*A5* = *A6* = 1) and the management of the buttons "start" and "stop" is not presented.

3.2 Turntable diagnoser

All PEs models are not shown in the paper. In this paragraph, only turntable is described, from PE model to diagnoser model.

The turntable PE is a monostable actuator and then is driven by only one output (Fig. 6). At the initial state x_0, the turntable is at the loading position c4. When output *A4* is activated by the controller, then the turntable is in movement and sensor *c4* is deactivated (state x_1 to x_2). From here, if the command is always activated, the turntable goes to unloading position *c5* (state x_3) and awaits the deactivation of *A4* to go back at the initial position (states x_4, x_5 and x_0). From state x_2, during the movement, if controller sends the deactivation of *A4* (state x_5), then the turntable comes back to state x_0 without having achieved sensor *c5*. And from state x_5, then during the movement back, it is possible to reactivate *A4*. This automaton with 6 states and 10 transitions represents all possible evolutions of the turntable PE. The others PEs have been modeled in the same way and are present in a library (Philippot, 2006).

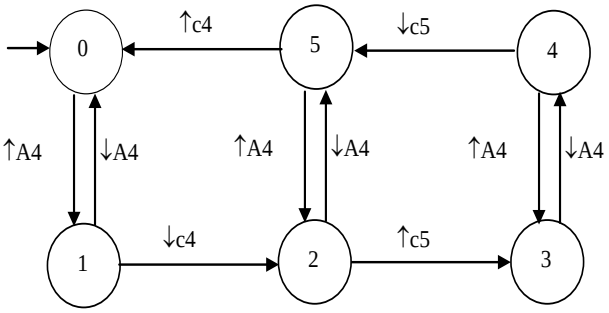

Fig. 6. Turntable Plant Element Model.

Turntable diagnoser is obtained after identification of all possible faults on the PE. It is an expert analysis which allows to define the following faults associated to a label:

- F1: Sensor *c4* stuck-off to 0
- F2: Sensor *c4* stuck-on to 1
- F3: Sensor *c5* stuck-off to 0
- F4: Sensor *c5* stuck-on to 1
- F5: Cylinder stuck to *c4* position
- F6: Cylinder stuck to *c5* position
- F7: Unexpected *c4* from 0 to 1
- F8: Unexpected *c4* from 1 to 0
- F9: Unexpected *c5* from 0 to 1
- F10: Unexpected *c5* from 1 to 0
- F11: Unexpected movement from *c4* to *c5*
- F12: Unexpected movement from *c5* to *c4*
- F13: Cylinder blocked from *c4* to *c5*
- F14: Cylinder blocked from *c5* to *c4*

Three fault partitions can be defined:

- To sensor *c4* : Πc4 = {F1, F2, F7, F8}
- To sensor *c5* : Πc5 = {F3, F4, F9, F10}
- To actuator *A4* : ΠA4 = {F5, F6, F11, F12, F13, F14}

The assumption that only one fault can occur at the same time in a PE is kept. For each state of PE, an expert must analyse the possibility of fault occurrence and especially the possibility to detect and isolate a fault. It returns a model with faulty states labelled (Fig. 7). « ? » symbol corresponds to a floating time which depends to previous state. The diagnoser is initialized from a normal state according to the input/output vector.

From the analysis, a first conclusion is that one fault can occur from any normal state without being directly detected. Consequently, all normal states (from x_0 to x_5) cannot be considered as dependable with certainty and then cannot have only label {N}. These states must have multi labels {N, F1, F2, …}. These labels are not marked to improve legibility.

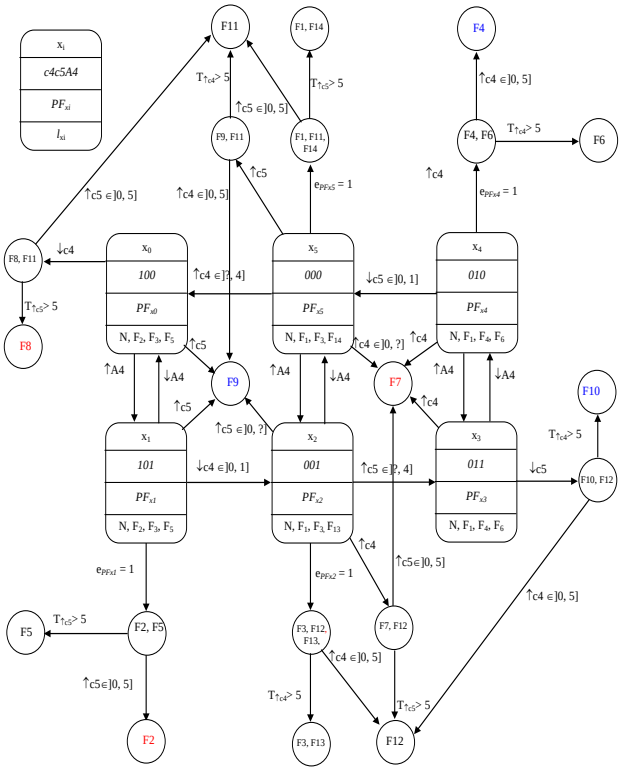

Fig. 7. Turntable Diagnoser.

From state x_1, if $\downarrow c4$ does not appear in the defined interval]0, 1], an event is generated ($e_{PFx1} = 1$). The turntable is then considered in an abnormal situation and a fault is detected. From the expert analysis, 2 candidates can be responsible for this fault, F2 or F5. To isolate the fault, diagnoser waits for a new observation or not to help it in its decision. If sensor *c5* is activated in an interval $]t_{min\downarrow c4} + t_{min\uparrow c5}, t_{max\downarrow c4} + t_{max\uparrow c5}]$, consequently]0, 5], then a movement has been realized by the turntable and the faulty event is F2, sensor *c4* stuck-on to 1. By contrast, if $\uparrow c5$ does not appear after ($t_{max\downarrow c4} + t_{max\uparrow c5}$), consequently 5 units of time, the turntable is considered as stuck on the position *c4* and the label is F5. This deduction can be realized thanks to the single fault assumption. All abnormal states are defined in the same way and finally the turntable diagnoser is composed of 6 normal states and 12 final abnormal states (8 intermediate states). All turntable faults can be detected but it is not possible to isolate with certainty states {F1, F14} and {F3, F13}.

3.3 Discussion

A first discussion from the turntable diagnoser can be quickly made. As each label is detected in a bounded delay and with certainty in one state, the turntable is then detectable according to the partition of faults defined. However, the turntable is not diagnosable because all faults cannot be isolated. Consequently, another rule must be present to guarantee diagnosability notion (Lin, 1994).

Concerning the delay of detection and isolation, it depends on the numbers of states of each local diagnoser. Here, delays are counted in terms of events due to the controller. For the turntable diagnoser, maximum delay to detect and isolate a fault is 3 events. For example, *c5* can be stuck-off to 0 from state x_5, as soon as it was deactivated, and it will be isolated from state x_2, after events ($\uparrow c4$, $\uparrow A4$, $\downarrow c4$).

Thanks to failures simulation mode of ITS PLC, it is interesting to see that if the diagnosis information is returned to the controller, it is possible to establish a fault tolerant controller and avoid mechanical and/or products failures (F2, F4, F7, F9). In contrast, some failures (F11 and F12) are unavoidable.

4. CONCLUSIONS

The paper presents an approach for diagnosis of discrete manufacturing systems based on the plant decomposition. A decentralized structure is used to diminish combinatory explosion found in centralized structure. From Plant Elements Models, all possible faults are identified to construct abnormal behavior models called diagnosers. The approach is illustrated using a benchmark.

However, only the faults related to components (actuators and sensors) are considered. To take into account the product faults, a product model is necessary. This model depends on the product nature and on the production objective. Thus, a future work is to extend this approach to include the faults related to product.

Another prospect is to use diagnosers with filter approach to be dependable regardless the controller. This filter is a set of constraints which must be verify the safety and liveness properties of the system (Marangé et al., 2009). Finally, the approach must be implemented into a Programmable Logic Controller for a real, and not simulated, system such as the flexible manufacturing system platform Cellflex of university of Reims (http://meserp.free.fr/) or the CISPI installation of university of Nancy.

ACKNOLEDGEMENT

These works are integrated in incentive action of university of Nancy. Authors wish say thanks all staff of this project.

REFERENCES

Ashley, S. (1993). Failure analysis beats Murphy's law, *Mechanical engineering*, ISSN 0025-6501, vol. 115, n°9, pp. 70-72.

Balemi, S., Hoffmann, G.J., Gyugyi, P., Wong-Toi, H. and Franklin G.F. (1993). Supervisory control of a rapid thermal multiprocessor. *IEEE Transactions on Automatic Control*, vol. 38, N°7, pp.1040-1059.

Boufaïed, A. (2003). Contribution à la Surveillance

Distribuée Des Systèmes à Evénements Discrets Complexes. *Thesis*, University Paul Sabatier, Toulouse.

Cassandra, C.G. and Lafortune, S. (1999). Introduction to Discrete Event Systems. *Kluwer Academic Publisher*, ISBN 0792386094.

Cordier, M.O., Le Guillou, X. Robin, S., Rozé, L. and Vidal, T. (2007). Distributed Chronicles for On-line Diagnosis of Web Services. 18th International Worshop On Principles of Diagnosis (DX'07), Nashville, USA.

Holloway, L.E. and Chand, S. (1994). Time templates for discrete event fault monitoring in manufacturing systems. American Control Conference, Baltimore, USA.

IEC 61131-3 (1993). *International Electrotechnical Commission, PLCs – Part 3: programming languages*. Publication 611131-3.

Lin, F. (1994). Diagnosability of Discrete Event Systems and its Applications. *Discrete Event Dynamic Systems*, Kluwer Academic Publishers, USA.

Marangé, P., Gouyon, D., Pétin, J.F. and Riera, B. (2009). Verification of functional constraints for safe product driven control. 2nd IFAC Workshop on Dependable Control of Discrete-event Systems (DCDS'07), Bari, Italy.

Philippot, A., Tajer, A., Gellot, F. and Carré-Ménétrier, V. (2004). On-line synthesis approach based on a structured plant modeling. IFAC 7th Workshop on Discrete Event Systems (WODES'04), Reims, France.

Philippot, A. (2006). Contribution au Diagnostic Décentralisé des Systèmes à Evénements Discrets : Application aux Systèmes Manufacturiers. *Thesis*, University of Reims Champagne-Ardenne, France.

Philippot, A., Sayed-Mouchaweh, M. and Carré-Ménétrier, V. (2007). Unconditional Decentralized Structure for the fault diagnosis of Discrete Event Systems. 1st IFAC Workshop on Dependable Control of Discrete-event Systems (DCDS'07), Cachan, France.

Qiu, W. (2005). Decentralized/distributed failure diagnosis and supervisory control of discrete event systems. *Thesis*, Iowa State University, USA.

Riera B., Marangé P., Gellot F., Nocent O., Magalhaes A., Vigario B. (2009). Complementary usage of real and virtual manufacturing systems for safe PLC training, *8th IFAC Symposium on Advances in Control Education (ACE'09)*. Kumamoto, Japan.

Sampath, M. (1995). A Discrete Event Systems Approach to Failure Diagnosis. *Thesis*, University of Michigan, Michigan, USA.

Su, R. and Wonham, W.M. (2000). Wonham. Decentralized fault diagnosis for discrete-event systems. *CISS, Princeton*, New Jersey, USA.

Wang, Y., Yoo, T.S. and Lafortune S. (2005). Decentralized diagnosis of discrete event systems using conditional and unconditional decisions. 44th IEEE Conference on Decision and Control (CDC'05), Seville, Spain.