CONTROLLED-SYSTEM MODEL ADAPTED TO THE CONTROL LAW SYNTHESIS

Sébastien Henry, Eric Zamaï, Mireille Jacomino

Laboratoire d'Automatique de Grenoble (LAG)
ENSIEG - Rue de la Houille Blanche, BP 46
38402 Saint Martin d'Hères Cedex, France
e-mail: henrys, zamai, jacomino (@lag.ensieg.inpg.fr)

Abstract: Usually, in an industrial context, the design of discrete control law to drive manufacturing system is assumed off line by several experts. This is mainly due to the lack of a generic method to model the controlled system abilities. Consequently, not only the design of all the control laws mobilizes many PLC program developers, but also, in case of unexpected resource failures, the reconfiguration process can be only considered from a manual point of view. So, to bring a solution in this field, a specific controlled system model based on the operation point of view is proposed. And from the formalization of the operation behavior, the operations properties are defined. *Copyright ©2005 IFAC*

Keywords: Discrete-Event Systems, Logic Controllers, Control System Synthesis, Controlled Systems, Modelling.

1. INTRODUCTION

This paper deals with the dependability of automated systems and more particularly manufacturing systems (see Fig. 1). These systems are made of a Supervision, Monitoring and Control (SM&C) system and a controlled system defined by the set of resources and the product flow. Depending on the customer's request and services offered by the set of the resources, the control system applies control laws to act on the product flow. Generally, the control law are implanted in a Programmable Logic Controller and they are specified in one of the IEC 61131-3 languages (IEC, 1993). Only the Sequential Functional Chart language is considered in this paper. In spite of SFC advantages recognized by the industrials, the development of control law with SFC is not supported by a formal method to insure a correct code initially. Indeed, a verification step before the control law design is still now obligatory to insure correctness and safety requirements on the one hand (Zaytoon, 2002). The paper proposes a controlled system model to develop a new formal control law synthesis method based on this model.

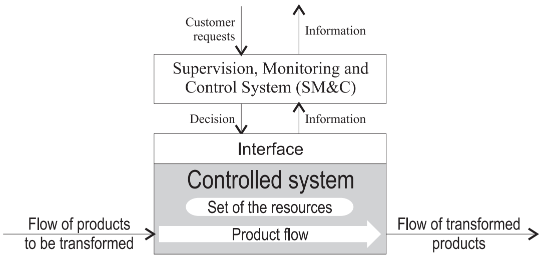

Fig. 1. A manufacturing system.

So, this paper is organized as follow: in section 2, main characteristics of the controlled system model are discussed. Section 3 presents to synthesize a control law a particular view of the controlled system. In section 4, the formalization of an operation and its properties are submitted. The paper then concludes and gives several research directions with section 5.

2. MAIN CHARACTERISTICS OF A CONTROLLED-SYSTEM MODEL

This paper is mainly directed towards the modelling of the controlled system to synthesize automatically a control law in SFC. The properties of a control law in SFC are to impose a deterministic controlled system behavior and to authorize the simultaneous events. In the context of the manufacturing system design, one of the objectives is to reduce the time required to create a control law and to minimize the debug time by creating correct code initially. Secondly, in the context of the dynamic reconfiguration, the resource failure reactivity and the customer's request variation must be definitely improved. The initial data of the synthesis problem are the objectives having to reach, an appropriate controlled-system model, and the initial controlled-system state. The objectives are made up of product specifications, the criteria to optimize the control law (i.e. time cycle or due date for instance) and sometimes the final controlled-system state (i.e. ready to start an other cycle for instance). A controlled-system model is appropriate when it represents the faithful picture of the controlled-system behaviors respecting constraints as security and environmental. In the next paragraphs, the results of two main synthesis methods are compared with the expected properties of a control law. The first synthesis method stemmed from the research field of the discrete-event system control is based on the RW theory (Ramadge and Wonham, 1987). And the second one is the general method used in the scheduling research field.

To drive the controlled system, the supervised control presented in (Charbonnier *et al.*, 1999) is based on the RW theory. In this approach, the supervised system is a set of resources with their local control law given by an expert. From the automata system model, the supervisor limits the system behavior to the most permissive, which respects the objective specifications. So, the supervisor does not impose a determinist behavior on the controlled system but it limits it even thought a control law imposes a behavior on the controlled system.

The result of scheduling methods (Chretienne *et al.*, 1997) is a precedence graph or with an initial date, a Gantt graph. It imposes a behavior on a system. Finally, it is a control law which has an upper level point of view. To synthesize a control law in SFC language, we propose to adapt the scheduling method to the lowest level. The first step to achieve the above proposition is to verify the controlled-system model adequacy with the lowest level constraints. And if it is necessary, it must be adapted. First, the classical considered constraints are not sufficient for the scheduling problems with operations which are not only manufacturing or transport operations, like the problems with setup operations (Gupta, 1982). A solution of the previous problem is to have a more detailed operation model with constraints on the resource state. But only considering constraints on the resource to run the operation, it is impossible to model all the controlled-system constraints respected by a control law. For instance, a cylinder can extend only if the other resources and the product flow are in a state for which there will be not any collisions between resources and any product falls. Then from the same principle proposed in (Gupta, 1982), we propose to extend the constraints on the other resource states and on the product flow states. As the scheduling problems with constraints on the resource state, the operation constraints must be verified in comparison to the updated controlled-system state. During the synthesis step, to update the controlled-system state, an operation model must also define the operation effects on the controlled-system state. The proposed model of a controlled system is near to the action model in automated planning field (Ghallab *et al.*, 2004). The next section submitted a controlled-system model adapted to the control law synthesis presented in (Henry *et al.*, 2004).

3. VIEW OF A CONTROLLED SYSTEM

The basic function defined by the norm NF X50-100, and in (ValueMethodology, 2005) of a manufacturing system is to increase the value of the product flow. The control system does not drive directly the product flow evolutions, it drives the resource evolutions. Thus, we propose to represent the controlled-system model by the set of the product flow evolutions, the resource evolutions and the links between the product flow and resource evolutions. With these links and from product flow evolutions respecting the product specifications, the required resource evolutions can be deduced, and in this way a control law can be automatically synthesized. If the product flow and resource evolutions are modelled in a same operation, the operation model defines this link.

3.1 SAPHIR

To perform the understanding of the proposed approach, this section presents an application example based on the loading system (see Fig. 2) of the research platform Saphir of the Laboratoire d'Automatique de Grenoble, in France. This platform is dedicated to the assembly of camshafts. A rotating storage been made up of a tray with four places is used to receive until six different kinds of products. These products are identified by a

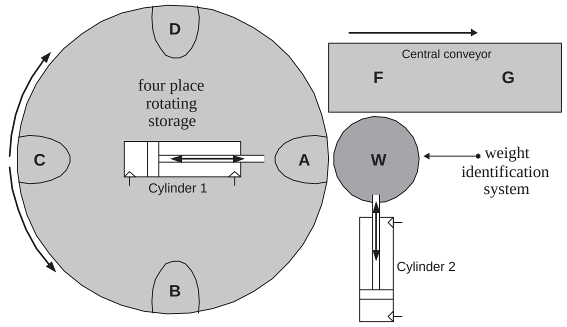

Fig. 2. Loading system of SAPHIR.

weight identification system. Once a product has been identified, a central conveyor drives it to an output device. So, a robot takes the different products to assembly them. A shopworker is charged to fill the rotating storage and to empty the assembly station.

3.2 Concept of operation

To reach the final product flow state given by the product specifications, a set of decisions must be made to define the operation having to be run. Then, it is required to have information on the operation effects on the product flow. To have these effects, the operation modifies the state of the resource which runs the operation. Thus a resource evolution defined by initial, intermediate and final resource states is characteristic of an operation. Depending on the product flow state before the operation running, an operation can have none or several different effects on the product flow. For instance, the cylinder 1 (C1) state from the retracted position to the extend position is characteristic of the "Extend Cylinder 1" (EC1) operation. If there is a product in A, this operation will then have an effect on a product. But also if there is a product between A and W. Thus, an operation model is made up of a basic sub-behavior describing the effect on the resource, and none or several extra sub-behaviors describing the possible effects on the product flow, as shown in Fig. 3.

But the effect modelling does not take the security and environmental constraints into account. For instance, this model does not guarantee that collision between the cylinders (1 and 2) will be avoided. Thus, the EC1 running ends accurately only if constraints on the other resources and the product flow are satisfied before (pre-constraints), during (constraints), and after (post-constraints) of the EC1 running, as in the associated constraints in Fig. 3. Finally to optimize the control law, the quantifiable criteria like the time cycle are required. To assess these criteria, the operation modelling must give the operation features like the duration.

Finally, the loading system model is made up of thirteen operations which can have between none and four extra sub-behaviors.

4. FORMAL OPERATION MODEL

To define the behavior and the properties of an operation, it is required to define a notation of the automaton used and of an operation. The operation behavior is then characterized with this notation. Finally, operation properties are submitted.

4.1 The automaton notation.

The controlled system is made up of all the resources and the product flow. The state of a resource or a product is defined by state variables (sv). A state q of the controlled system is defined by the value of each sv. The set of the Controlled-System Model states is denoted Q_{CSM}. We assume the following partition: $Q_{CSM} = Q_A \cup Q_F$, where Q_A and Q_F are the set of Authorized and Forbidden states, respectively. From the control law synthesis point of view in the SM&C context (Combacau *et al.*, 2000), the controlled system is a system that evolves in accordance with the occurrence of events: "start operation i" ,"end operation i". The state evolution of the resources and the product flow may be seen as a four-tuple deterministic automaton: $CS = (Q_{CSM}, \Sigma, \delta, q_0)$ where: Σ is the alphabet of events defined above; $\delta : Q_{CSM} \times \Sigma \rightarrow Q_{CSM}$ is a partial transition function; and q_0 is the initial state. When the designer begins the modelling of the controlled system, it is not exist any operation. Then, Q_A and Q_F are empty, and there is not any event, and any δ.

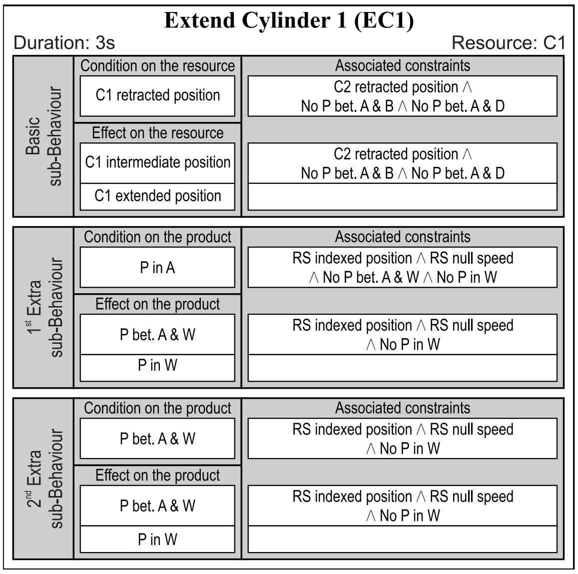

Fig. 3. "Extend Cylinder 1" operation (EC1).

4.2 The operation notation.

First, the notation of the different sub-behaviors is submitted. Then, the notation of an operation behavior is presented. And this section finishes with the definition of the sub-behavior structure.

An operation i will be denoted O_i. An operation is made up of two kinds of sub-behaviors (see Fig. 4). The basic sub-behavior, denoted bb_i, defines the effect on the resource with the associated constraints. The extra sub-behaviors, denoted $eb_{i,j}$, define effects on the product flow with the associated constraints. The $eb_{i,j}$ number is not limited, so $j \in [0, N]$. When an operation is run from a state, the effects defined by bb_i and $eb_{i,j}$ are obtained simultaneously.

The set of the extra sub-behaviors of O_i is

$$EB_i = \{eb_{i,j}\}_{j \in [0,N]}$$

Depending on the controlled-system state from which O_i is run, all the extra sub-behaviors are not inevitably obtained. Or two extra sub-behaviors cannot be incompatible, e.g. they can never be obtained simultaneously for all controlled-system states. For the above reasons, it is necessary to consider the sub-sets of EB_i, denoted $EB_{i,k}$. The number of $EB_{i,k}$ with p $eb_{i,j}$ elements is the combination of N elements taken p at a time $({}^{N}C_p)$. If all the extra sub-behaviors are compatible, the maximum number of $EB_{i,k}$ is equal to 2^N. $EB_{i,k}$ is defined by:

$$EB_{i,k} = \{eb_{i,j}/j \in J_k \wedge k \in [0, 2^N - 1]\}$$

$$J_k = \{x/(k = \sum_{y=0}^{N-1} c_y 2^y / c_y \in [0,1] \wedge y \in \mathbb{N}) \wedge c_y = 1 \Rightarrow x = y + 1\}$$

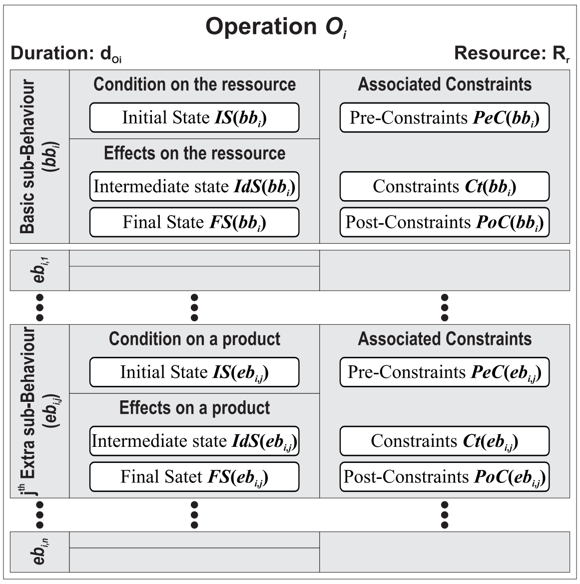

Fig. 4. The notations of the operation elements.

When an operation is run, the basic sub-behavior is always obtained. Finally, a behavior of a O_i operation is made up of the basic sub-behavior and a sub-set of EB_i. It is denoted $B_{i,k}$ and it is defined by:

$$B_{i,k} = \{bb_i, EB_{i,k}\}$$

The set of the O_i behaviors is denoted $B_i = \{B_{i,k}\}$. If all the extra sub-behaviors are compatible, the number of $B_{i,k}$ is equal at the cardinality of $EB_{i,k}$: 2^N. For instance, the extra sub-behaviors of the EC1 operation (O_{EC1}) are:

$$EB_{EC1} = \{eb_{EC1,1}, eb_{EC1,2}\}$$

Then $k \in [0, 2^2 - 1]$ and the set of J_k is:

$$J_0 = \{\emptyset\},\ J_1 = \{1\},\ J_2 = \{2\},\ J_3 = \{1, 2\}$$

Finally, the set of the $EC1$ behaviors is: $B_{EC1,0} = \{bb_{EC1}\}$, none extra sub-behavior is simultaneously obtained with the basic sub-behavior. $B_{EC1,1} = \{bb_{EC1}, EB_{EC1,1}\}$, extra sub-behavior 1 is simultaneously obtained with the basic sub-behavior. $B_{EC1,2} = \{bb_{EC1}, EB_{EC1,2}\}$, extra sub-behavior 2 is simultaneously obtained with the basic sub-behavior. $B_{EC1,3}$=$\{bb_{EC1}$, $EB_{EC1,1}$, $EB_{EC1,2}\}$, the extra sub-behaviors 1 and 2 are incompatible, then the $B_{EC1,3}$ behavior does not exist.

It is still impossible to define the operation behavior without giving details on the sub-behaviors (the basic sub-behavior (bb_i) and the extra sub-behaviors ($eb_{i,j}$)). The structure of the sub-behaviors is generic (see Fig. 4). They are made up of:

The sub-behavior effect on the controlled system. The intermediate state (IdS) defines the effect that results from the "O_i start" event occurrence. After the "O_i start" event occurrence, $IdS()$ gives the value of each one of sv on which the sub-behavior have an effect. And the final state (FS) defines the effect that results from the "O_i end" event occurrence. After this event occurrence, $IF()$ gives the value of each one of sv on which the sub-behavior have an effect. The effects ($IdS(bb_i)$, $FS(bb_i)$) of the basic sub-behavior (bb_i) can be only on the sv of the Rr resource running the operation. And the effects ($IdS(eb_{i,j})$ and $FS(eb_{i,j})$) of an extra sub-behavior ($eb_{i,j}$) can be only on the sv of the product flow.

The condition on the initial state. For a state q, if the condition (IS) of a sub-behavior is true, then the "start O_i" event occurrence will cause the effects defined by the sub-behavior. The condition is specified by a propositional formula with an atomic proposition is defined by the value of a state variable. The atomic proposition can be expressed with the sv of the resources or the product flow. And the sv used to specify the sub-behavior effects must be use in the condition specification. The conditions of the basic sub-behavior (bb_i) and an extra sub-behavior ($eb_{i,j}$) are denoted $IS(bb_i)$ and ($eb_{i,j}$), respectively.

The associated constraints. For the controlled system evolution resulting from the effects of a sub-behavior, they insure the respect of the security and environmental constraints to avoid the harmful consequences. The pre-constraints (PeC), the constraints (Ct) and the post-constraints (PoC) are specified by a propositional formula with an atomic proposition is defined by the value of a state variable. The sv used here have not to exist in IS, IdS and FS of the sub-behavior.

4.3 Method of Operation Modelling

The modelling of an operation begins by the definition of the effect on the Rr resource and next of the possible effects on the product flow resulting from the resource evolution. The sub-behaviors (bb_i and $eb_{i,j}$) having the same structure, the modelling method is the same for the effect on the resource with the associated constraints and for an effect on the product flow with the associated constraints. To model an effect, the designer defines the effect with the required sv and the possible harmful consequences which infringe the security and environmental constraints. These consequences can be the collisions between the Rr resource and the other resources, or the harmful consequences on the product flow (product falls from the weight system). The harmful consequences are formalized by forbidden states of the sub controlled system. For instance to define the collision between C1 and C2, the sub controlled system is constituted with these cylinders only with the state variables of position. Then, the four forbidden states are defined by the bellow proposition:

$$[\text{C1 extend} \vee \text{C1 inter}] \wedge [\text{C2 extend} \vee \text{C2 inter}]$$

From the forbidden states of the sub controlled system, the forbidden states of the controlled system are the states for which the proposition is true. When an operation is completely specified, the new states are added to the Q_{CSM} from the new required sv to specify the operation (the effects with their associated constraints). And the new forbidden states are added to Q_F. The events associated with the start and the end of this operation is added to Σ. And the behaviors of O_i define a partial transition function.

4.4 The behavior of an operation

The operation behavior $B_{i,k}$ is defined when the operation is the only one to be run. For a state $q \in Q_{CSM}$, the effects of the operation behavior is defined by the effect of the basic sub-behavior $\{bb_i\}$ and the simultaneously obtained extra sub-behaviors $\{eb_{i,j}/j \in J_k\}$. From a state q after the ”O_i start” event occurrence, the effect of a sub-behavior is obtained with the respect of the security and environmental constraints, iff the condition are true and the associated pre-constraints are satisfied. And from a state q after the ”O_i start” event occurrence, the effect of an extra sub-behavior is unobtained iff the condition are false.

Definition 1. The set $Q_{I(B_{i,k})}$ of the initial $B_{i,k}$ states is the set of the states from which the ”start of $B_{i,k}$” event occurrence is authorized. Formally, this set is represented by:

$$Q_{I(B_{i,k})} = \{q \in Q_{CSM}/IS(bb_i) \wedge PeC(bb_i) \bigwedge_{j\in J_k} [IS(eb_{i,j}) \wedge PeC(eb_{i,j})] \bigwedge_{j\in \overline{J_k}} \neg IS(eb_{i,j})\}$$

Definition 2. The set $Q_{Id(B_{i,k})}$ of intermediate $B_{i,k}$ states is the set of the states in which the controlled system can be after the ”start of $B_{i,k}$” event occurrence and from which the ”end of $B_{i,k}$” event occurrence is authorized. Formally, this set is represented by:

$$Q_{Id(B_{i,k})} = \{q \in Q_{CSM}/IdS(bb_i) \wedge Ct(bb_i) \bigwedge_{j\in J_k} [IdS(eb_{i,j}) \wedge Ct(eb_{i,j})] \bigwedge_{j\in \overline{J_k}} \neg IdS(eb_{i,j})\}$$

Definition 3. The set $Q_{F(B_{i,k})}$ of the final $B_{i,k}$ states is the set of the states in which the controlled system can be after the ”end of $B_{i,k}$” event occurrence. Formally, this set is represented by:

$$Q_{F(B_{i,k})} = \{q \in Q_{CSM}/FS(bb_i) \wedge PoC(bb_i) \bigwedge_{j\in J_k} [FS(eb_{i,j}) \wedge PoC(eb_{i,j})] \bigwedge_{j\in \overline{J_k}} \neg FS(eb_{i,j})\}$$

Definition 4. From a state $q \in Q_{I(B_{i,k})}$ and after the occurrence of ”start of $B_{i,k}$” event, denoted ($s_{B_{i,k}}$), the controlled system is in a state q'. The sv values of this state are the sv values of the state q which are modified by $IdS(bb_i)$ and $IdS(eb_{i,j})$ for $j \in J_k$. The partial transition function is defined by $q' = \delta(s_{B_{i,k}}, q)$. From the state q' and after the occurrence of ”end of $B_{i,k}$” event, denoted ($e_{B_{i,k}}$), the controlled system is in a state q”. The sv values of this state are the sv values of the state q which are modified by $FS(bb_i)$ and $FS(eb_{i,j})$ for $j \in J_k$. The partial transition function is defined by q” $= \delta(e_{B_{i,k}}, q)$. From a state $q \in Q_{I(B_{i,k})}$, the effects of the $B_{i,k}$ behavior on the controlled system can be represented as below:

$$q \xrightarrow{s_{B_{i,k}}} q' \xrightarrow{e_{B_{i,k}}} q''$$

4.5 Operation properties

The seven following properties can be only verified in comparison to the defined forbidden state. All these verifications allow only to know if an operation is not correctly specified.

Property 1. For an operation, it exists at least a $B_{i,k}$ behavior whose $Q_{I(B_{i,k})} \neq \{\emptyset\}$. In other word, it exist at least a state from which an operation (O_i) can be run.

If none state exists from which the operation can be run, the operation is useless. Either the operation can be suppressed of the controlled-system model, or there are one or several elements which are not correctly specified among the following elements: the conditions ($IS(bb_i)$, $IS(eb_{i,j})$ $\forall j \in [0, N]$) and the pre-constraints ($PeC(bb_i)$, $PeC(eb_{i,j})$ $\forall j \in [0, N]$).

Property 2. For an extra sub-behavior, it exists at least a $B_{i,k}$ behavior whose $Q_{I(B_{i,k})} \neq \{\emptyset\}$ and for which the extra sub-behavior is obtained.

If this property is not verify, the extra sub-behavior is unless. Either the extra sub-behavior can be suppressed of the operation model, or there are one or several elements which are not correctly specified among the following elements: the $(IS(bb_i), IS(eb_{i,j}))$ conditions and the $(PeC(bb_i), PeC(eb_{i,j}))$ pre-constraints.

The following property is based on the designer abilities to define the wished operation behaviors.

Property 3. For an operation behavior wished by the manufacturing system designer, the set of the initial state $(Q_{I(B_{i,k})})$ are not empty.

Property 4. $\forall q' \;/\; q' = \delta(s_{B_{i,k}}, q)$, $q' \in Q_{Id(B_{i,k})}$

Property 5. $\forall q" \;/\; q" = \delta(e_{B_{i,k}}, q')$, $q" \in Q_{F(B_{i,k})}$

If one of the previous properties is false, the effects of the operation behavior infringe the associated constraints, then there are one or several elements which are not correctly specified among the operation elements.

Property 6. $\forall q \in Q_{I(B_{i,k})} \cap Q_A$ and $q' = \delta(s_{B_{i,k}}, q)$, $q' \in Q_A$

Property 7. $\forall q' \in Q_{Id(B_{i,k})} \cap Q_A$ and $q" = \delta(e_{B_{i,k}}, q')$, $q" \in Q_A$

From an authorized state, the two previous properties assure that the operation behavior has not for effect to put the controlled system in a forbidden state.

5. CONCLUSION AND FUTURE WORKS

In this paper, problem of controlled-system modelling to synthesize discrete control laws is dealt. As in the scheduling and the automated planning research fields, each resource is described by all the offered operations. To model the lowest level constraints, an extended operation model is proposed. The operation model structure is generic. This model is made up of the basic sub-behavior and of none or several extra sub-behaviors. A sub-behavior is defined by a condition, an effect on a resource or on the product flow, and the associated constraints. From the operation formalization, the operation properties are defined. When the properties are not verified, the controlled-system model is not correctly specified.

Future works will first focus on the algorithm to synthesize a discrete control law from the proposed controlled system model. Second, in the reconfiguration context, the synthesis approach will be integrated in a whole approach integrating diagnosis and prognosis abilities to increase the reactivity in case of resource failures.

REFERENCES

Charbonnier, F., H. Alla and R. david (1999). The supervised control of discret-event dynamic systems. *IEEE Transactions on Control Systems Technology* **7**(2), 175–187.

Chretienne, P., E. G. Coffman, J. K. Lensta and Z. Liu (1997). *Scheduling theory and its applications.* Chichester: John Wiley and Soons.

Combacau, M., P. Berruet, E. Zamaï, P. Charbonnaud and A. Khatab (2000). Supervision and monitoring of production systems. In: *IFAC Conference on Management and Control of Production and Logistics.* Grenoble, France.

Ghallab, M., D. Nau and P. Traverso (2004). *Automated Planning, Theory and Practice.* Elsevier.

Gupta, S. K. (1982). n jobs and m machines job-shop problems with sequence-dependent set-up time. *International Journal of Production Research* **20**(5), 643–656.

Henry, S., E. Deschamps, E. Zama and M. Jacomino (2004). Control law synthesis algorithm for discrete-event systems. In: *IFAC Conference on Management and Control of Production and Logistics.* Santiago, Chili.

IEC (1993). *61131-3, http://www.iec.ch.*

Ramadge, P. and W. Wonham (1987). Supervisory control of a class of discret event processes. *Journal of Control and Optimization.*

ValueMethodology (2005). http://www.value-eng.org/catalog_monographs.php.

Zaytoon, J. (2002). On the recent advances in grafcet. *Production Planning and Control* **13**(1), 86–100.