A Synthesis Approach for Reconfigurable Manufacturing Systems Design Based on Petri Nets

Armand Toguyeni, Nathalie Dangoumau, Eun Joo Lee

LAGIS – CNRS UMR 8146

Ecole Centrale de Lille

Cité scientifique BP 48

59651 Villeneuve d'ASCQ Cedex, France

Phone: +33-3-20-33-54-49, Fax: +33-3-20-33-54-18

E-mail: Armand.Toguyeni@ec-lille.fr, Nathalie.Dangoumau@ec-lille.fr, lee.eun_joo@ec-lille.fr

Abstract: This work addresses the problem of the design of the control of Reconfigurable Manufacturing Systems by a synthesis approach. Our approach is inspired by Supervisory Control theory but is based on Petri nets (PN) models. The plant and the user specifications are both modeled by PN. The Ramadge and Wonham original works uses finite automata and many authors investigate synthesis approaches based on this formalism. However, finite automata present some drawbacks such as the difficulty to model parallelism, synchronization and resource sharing. The approaches based on this formalism are generally limited by the combinatorial explosion that occurs when attempting to model complex systems. On the other hand, Petri Net is a more powerful tool for modeling parallelization or synchronization. Our approach is also based on the Ghaffari's Region Theory. This theory propose to synthesis control places that are added in an initial PN plant model in order to obtain a closed-loop PN that respect the user specifications. To make easier the use of our approach, we have developed a Petri net tool that assists a designer to build the plant controllers.

Keywords: Petri net, controller synthesis, forbidden sequences, Reconfigurable Manufacturing System, Resource sharing, Constrained Synchronous Reachability Graph.

Armand Toguyéni is a Professor of Computer Sciences and Discrete Events Systems (D.E.S.) at the Ecole Centrale de Lille (France). He has in charge the Department of Computer Sciences of the Institut de Génie Informatique et Industriel de Lens. Born in Dakar (Senegal) in 1964, he obtained in 1988 the Engineer Diploma of the "Institut Industriel du Nord" (French "Grande Ecole") and the same year his Master Degree in Computer Sciences. He obtained a Ph.D. in Automatic control for Manufacturing and Discrete Events systems in 1992 and his "Habilitation à Diriger des Recherches" in 2001. Professor Toguyeni's research area is the improvement of the Quality of Service of DES. More precisely, his research concerns computer networks, transport system and manufacturing system. In manufacturing systems field, his current researches are about the development of models for the design of Reconfigurable Controllers depending on the state of the plant and the production objective. His works also concerns Fault Detection and Isolation techniques for Flexible Manufacturing Systems (FMS).

Nathalie Dangoumau is an Associate Professor of Computer Sciences and Discrete Events Systems (D.E.S.) at the Ecole Centrale de Lille (France). She obtained a Ph.D. in Automatic control for Manufacturing and Discrete Events systems in 2000. Her research is about reconfiguration and modes management for Flexible Manufacturing Systems (FMS).

Eun Joo Lee was born in South Corea, on July 15, 1970. She obtained in 1999 the the Master Degree in Computer Sciences of the National University of Changwon (South Corea). She is PhD in Production Management, Automatic Control and Computer Sciences of the Ecole Centrale de Lille (France) since 2006. Her research is about the design of Reconfigurable Manufacturing Systems based on the use of synthesis technique such as supervisory control and the use of Petri Net models.

1. Introduction

In the eighties, the concept of flexible manufacturing systems (FMS) was introduced to develop new systems of manufacturing production able to produce the small ones and average series of products. But today, the capacity of reconfiguration is become a major problem to improve the functioning of industrial processes. Indeed, actually a main objective is to adapt quickly the system in order to start a new production or to react to failures. This leads some authors to introduce the concept of Reconfigurable Manufacturing Systems (RMS) as a final solution to maintain industrial productions in westerns countries [1], [2]. Such type of system is characterized by its reconfigurability capacity that is obtained both by the flexibility of the plant and the adaptatively of its control software. These features can be ensured by introducing redundancies between the operations held by the plant resources. This lead to design control software that is modular, generic and whose parameters can be modifiable depending on the plant functioning mode.

In this paper, our aim is to address the problem of designing Petri Nets controllers to control Reconfigurable Manufacturing Systems with respect to user specifications. Our proposition is a supervisory control theory like approach but proposes to model user specifications by Petri nets. Ramadge and Wonham have investigated supervisory control problems based on automata formalism [3]. However, finite automata present some drawbacks such as the difficulty to model parallelism, synchronization and resource sharing. The approaches based on this formalism are generally limited by the combinatorial explosion that occurs when attempting to model complex systems. On the other hand, Petri Net (PN) is a more powerful tool since it allows modeling control such as parallelization or synchronization. For this reason, many works try to propose a synthesis approach based on PN formalism [4] [5].

This study concerns more specifically the design of resource controllers. In this context, the second section defines the problematic and our study context. The third section addresses resource controllers' design. We will first present a new problem based on Petri Net modeling, the "Forbidden Sequences of State-Transitions Problem" (FSSTP). Following that, we will propose the "Constrained Synchronous Reachability Graph" (CSRG) to synthesize plant PN model and user specifications' PN model. Our approach is also based on Region Theory and allows obtaining a close-loop behavior PN by the addition of control places in the original plant PN models. In the last section, we propose a software tool that allows automating this approach in the context of a decentralized synthesis.

2. Study Context and Problems

To control a manufacturing system, the LAGIS's approach can be summarized by the design of two main functions: sequential control and piloting control (Fig. 1).

Sequential control aims to define the sequence of order to send to the plant to make a finished product from raw parts. To reach this goal, one can decompose this function in two tasks corresponding to two viewpoints: the Coordination Graph of Transport System (CGTS hereafter) and the Operating Sequences [6]. They are both Petri nets (PN) based models. To take account of the plant flexibilities they integrate indeterminisms that are solved in real time by the piloting function. CGTS controls the transfers and the routing of part between the different resources that compose the plant. At this level, the problems are both to solve the constraint introduced by buffers placed between the machines and by the allocation of transport resources.

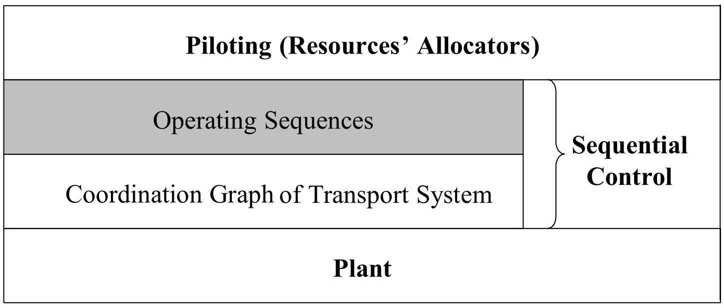


Figure 1. Hierarchical View of RMS Control Architecture

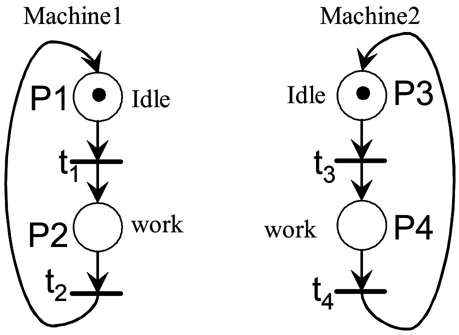
Operating sequences are PN models that describe the operations to apply to transform raw parts in finished products. These operations are tasks performed by transformation or assembling machines. Because of the parallelism of parts' production, some resources can be required simultaneously by several parts: it is a conflict situation. To solve conflict situations it is necessary to design resources' allocators.

The second part of control architecture is the piloting function. The function of piloting consists both in defining the sequence an dates of raw parts entry in the plant and also in solving in real time the residual indeterminism of the models in sequential control. Our approach considers of course conflict situations but also the choice of variant alternatives in operating sequences models and part routing in the CGTS.

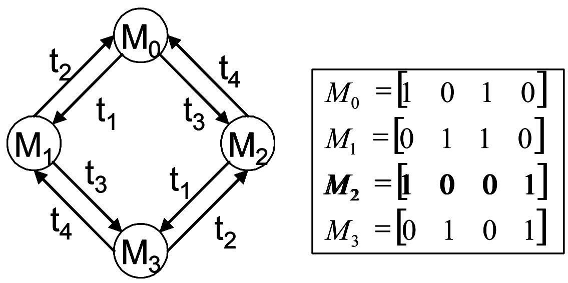
3. A Petri Net Based Synthesis Approach

3.1. A user specification problem: Forbidden sequences of state-transitions problem

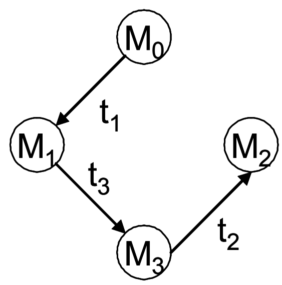
The FSSTP is different from a Forbidden State Problem (FSP) or a Forbidden State-Transition Problem (FSTP) [5] [7]. The example of Fig. 2 allows explaining informally this concept.



A. Plant PN Model



B. Reachability graph



C. A Forbidden Sequence

Figure 2. Example of the Forbidden Sequences of State-Transitions Problem

Fig. 2.A illustrates the plant PN model corresponding to two machines 1 and 2, evolving in parallel, independently, without any constraint. Fig.2.B gives its corresponding reachability graph. The plant behavior can also be expressed by the set of legal transition sequences. Based on language theory this set can be summarized by a regular expression. Fig.2.B shows that several sequences allow moving from marking M_0 to marking $M_2 : t_3, t_1t_2t_3, t_1t_3t_2 \cdots$. Let us suppose that the user specification requires inhibiting the sequence $t_1t_3t_2$ (Fig.2.C). In this problem, all markings remain admissible and all transitions remain fireable. This problem cannot be assimilated to a FSTP by a trivial process consisting in suppressing some transitions. For example, if one tries to suppress $M_0 \xrightarrow{t3} M_2$ to avoid the sequence $t_1t_3t_2$ then it will become impossible to perform sequence such as t_3t_4 that is not forbidden. In fact, the desired behavior corresponds to the set of sequences ($\left(t_1t_2\left(t_3t_1\left(t_2t_4 + t_4t_2\right)\right)^* t_3t_4\right)^*$). This regular expression specifies the two machines working as a producer-consumer system (M1 is the producer and M2 the consumer) with the constraint that the producer can have only an advance of one part with respect to the consumer production. For example, this constraint can be induced by a buffer with a unitary

capacity placed between the two machines. To solve this problem our idea is to translate the FSSTP in a FSTP. This can be performed by adding control places to the original plant PN models like we propose it in this paper. This is done based on a new interpretation of region theory proposed in [5], [8].

Formally, one defines an FSSTP as it follows:

Definition 1. A Forbidden Sequences of State-Transitions Problem (FSSTP) is a problem of supervision based on Petri nets. The wished behavior of plant model (N_p, M_0) must satisfy the following conditions:

(a) $M_0 \in DB$ (DB: Desired behavior)

(b) $\forall M \in DB$, $\exists \sigma \mid (M_0 \xrightarrow{\sigma} M) \in DB$

(σ: Sequence of firing transitions)

(c) $\exists (M \xrightarrow{\gamma} M') \in R(N_p, M_0)$ and $(M \xrightarrow{\gamma} M') \notin DB$

(γ: Sequence of forbidden transitions, $R(N_p, M_0)$: Reachability graph of Petri Nets model considered)

3.2. Synthesis of Petri net Controller: Constrained Synchronous Reachability Graph

As inspired by Ramadge and Wonham original theory, our approach is based on the plant PN and specification PNs that encode the user specifications. As our intention is to obtain an FSTP, specifications PNs models are written to express the firing conditions of some transitions. This means that the transitions of specification PNs models correspond to some transitions of the plant model. Consequently, to synthesis these PN models, we propose the constrained synchronous reachability graph (CSRG). The CSRG results in more small size than the reachability graph obtained by a cartesian product of the reachability graphs of the two types of models. It is useful to determine the forbidden sequences of state-transitions.

3.3. Introduction to the theory of regions

The synthesis problem for nets consists in deciding whether a given automaton is isomorphic to the reachability graph of a net and then constructing it. This problem has been solved in the literature for various types of net ranging from elementary nets to Petri nets. Its objective here is to determine a pure PN (N, M_0), characterized by an incidence matrix C and having a set of transitions T, such as his reachability graph is isomorphic to $RG(N, M_0)$ (an automaton which arcs are labeled by the elements of T).

Reference [8] proposes a new adaptation of the theory of regions that enables to add control places to an initial PN controller in order to control forbidden behavior. The Ghaffari's approach is summarized by the following theorem.

Theorem 1 [8]: There exist a PN, (N, M_0) which reachable graph is $RG(N, M_0)$, iff there exist a set P_c of control places $\{(M_0(p_c), C(p_c,.))\}$ as:

1. Each place p of (N, M_0) satisfies the cycle equation, (1), of $RG(N, M_0)$;

$$\sum_{t \in T} C(p,t)\vec{\delta}[t] = 0, \forall \delta \in \Delta \quad (1)$$

2. Each place p of (N, M_0) verifies the reachability equation, (2), of marking of $RG(N, M_0)$;

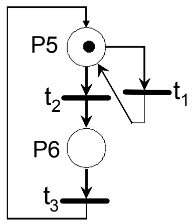
$$M_0(p) + C(p,.)\vec{\Gamma}_M \geq 0, \forall M \in R \quad (2)$$

3. To each pair (M, t) such as t does not fire from M, it exists at least one control place p_c which satisfies (3.3) of state separation.

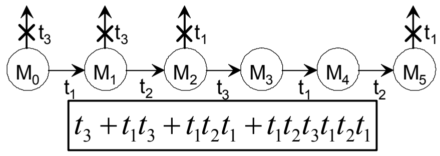
$$M(p_c) = M_0(p_c) + C(p_c,.)\vec{\Gamma}_M + C(p_c, t) < 0 \quad (3)$$

3.4. Application of Theory of Regions to the producer-consumer

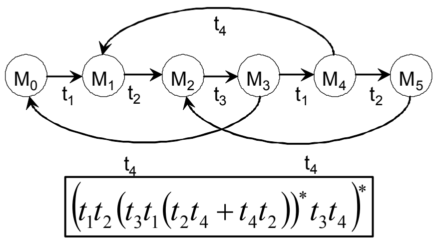
Let us consider the control of a small factory composed of the machines: machine1 the producer and machine2 the consumer. The behavior of this system must respect some constraints due to an intermediary buffer placed between the two machines.



A. Specification PN model



B. The set of forbidden sequences of state-transitions



C. The set of authorized sequences

Figure 3. Specification PN Model (A) and Result of CSRG Algorithm I (B and C)

Fig. 3.A illustrates the plant model describes by two PNs that control separately the two machines when considering them without the constraint introduced by the buffer. Considering that the buffer capacity is one part, the user specifications are the following:

- Machine 2 cannot be run when the buffer is empty,
- Machine 1 cannot be run when the buffer is full.

In the proposed approach, the designer must first translate these informal specifications in a Petri net model (see **Figure 3**.A). We obtain the set of forbidden sequences of state-transitions and the set of authorized sequence by algorithm I (see **Figure 3**.B and **Figure 3**.C). We can look all marking and all sequence of system example according to our CSRG algorithm. Now let us apply the result of CSRG algorithm by using the theory of regions [7], [8]. It is important that each control place (that we note 'P_C') added must verify the three equations defined by the theorem 1 in [7]. This graph contains three cycles that are all composed of the same four transitions: t1, t2, t3 and t4. Therefore all these cycles have the same equation (4):

$$C_c(p_c,t1) + C_c(p_c,t2) + C_c(p_c,t3) + C_c(p_c,t4) = 0 \qquad (4)$$

In the equations the notation C(p,t) represents a component of PN incidence matrix with 'p' a place and 't' a transition.

The reachability equations are obtained based on the sequences that enable to reach the different marking of the CSRG from the initial marking 'M_0'. They correspond to the set of equations (5). Equation (5.a) explains that the initial marking of the control places must be positive or null like the marking of all places in an ordinary Petri net. Equations (5.b) to (5.f) translate effectively the authorized sequences. For example, (5.b) formulates that the transition 't1' must be fireable from initial marking to reach the marking 'M_1'

$$M_0(p_c) \geq 0 \tag{5.a}$$

$$M_1 = M_0(p_c) + C(p_c, t1) \geq 0 \tag{5.b}$$

$$M_2(p_c) = M_0(p_c) + C(p_c, t1) + C(p_c, t2) \geq 0 \tag{5.c}$$

$$M_3(p_c) = M_0(p_c) + C(p_c, t1) + C(p_c, t2) + C(p_c, t3) \geq 0 \tag{5.d}$$

$$M_4(p_c) = M_0(p_c) + 2C(p_c, t1) + C(p_c, t2) + C(p_c, t3) \geq 0 \tag{5.e}$$

$$M_5(p_c) = M_0(p_c) + 2C(p_c, t1) + 2C(p_c, t2) + C(p_c, t3) \geq 0 \tag{5.f}$$

Finally, the forbidden sequences of state-transitions correspond to event condition separation. They are four equations.

$$M_0(p_{c1}) + C(p_{c1}, t3) < 0 \tag{6}$$

$$M_0(p_{c2}) + C(p_{c2}, t1) + C(p_{c2}, t3) < 0 \tag{7}$$

$$M_0(p_{c3}) + C(p_{c3}, t1) + C(p_{c3}, t2) + C(p_{c3}, t1) < 0 \tag{8}$$

$$\begin{aligned} & M_0(p_{c4}) + C(p_{c4}, t1) + C(p_{c4}, t2) + C(p_{c4}, t3) + \\ & C(p_{c4}, t1) + C(p_{c4}, t2) + C(p_{c4}, t1) < 0 \end{aligned} \tag{9}$$

These equations specify the inhibition conditions of the forbidden sequences of state-transitions. Equation (6) means that a control place, 'P_C', must inhibit the firing possibility of the transition 't_3' from initial marking. Equation (7) formulates that the sequence '$t_1 t_3$' is not permitted because in this case the buffer is empty and then machine 2 cannot be run. Equation (8) specifies that the transition 't_1' does not be fired again after a cycle limited to the machine1. Equation (9) corresponds to a work in parallel of the two machines and when the producer is faster than the consumer. When the buffer is full it is necessary to inhibit a new run of machine1.

3.5. Automation of the extraction of the theory of regions equations

The extraction of the theory of regions equation is based on an iterative interpretation of a constrained synchronous reachability graph (CSRG) resulting of the fusion of plant and user models taking into account controllability requirements. This interpretation is based on the two following rules:

Rule 1: *A transition of the specification model can only be fired when it is enabled in the two models.*

Rule 2: *An enabled transition of the plant PN model must be inhibited if this transition exists in the two models and if it is not simultaneously enabled in the specification model.*

Remark: The two previous rules imply that the resulting model can be seeing in a preliminary approximation as the merging of several PNs with common transitions. However the controllability problem requires that we refine this preliminary model to build a maximally permissive controller respecting the control requirements.

Based on the theory of regions, our aim is to propose an algorithm that by exploring the CSRG, allows constructing three set of sequences: the set of authorized sequence, the set of forbidden sequences of state-transitions and the set of cycles. So before presenting this algorithm let us give some definitions of terms and notations that will be used in the rest of the paper.

M_0 *is the initial marking,*

AM is the authorized marking,

PM is the set of processed marking,

AS is the set of authorized sequences,

FS is the set of forbidden sequences of state-transitions,

CS is the set of Cycles.

Theorem 2: If it exits a transition t_{ik} / $M_i[t_{ik}>M_k$ and t_{ik} verifies rule 1, then $\forall$ σ ∈ AS / $M_0[\sigma >M_i$, $\sigma.t_{ik} \in$ AS.

Proof: If t_{ik} verifies rule 1, this means that this transition is fireable both in plant PN model and specification PN model. If, it exists a sequence σ ∈ AS / $M_0[\sigma >M_i$, this means that M_i is belonging to AM. In consequence, as $M_i[t_{ik}>M_k$, it implies that M_k is also belonging to AM. In consequence the sequence $\sigma.t_{ik}$ allows reaching an authorized marking M_k from M0. It means that $\sigma.t_{ik}$ is belonging to AS.

Theorem 3: If it exits a transition t_{ik} / $M_i[t_{ik}>M_k$ and t_{ik} verifies rule 2, then $\forall$ σ ∈ AS / $M_0[\sigma >M_i$, $\sigma.t_{ik} \in$ FS.

Proof: If t_{ik} verifies rule 2, this means that this transition is not fireable in the specification PN model. This means that the transition from M_i to M_k is forbidden and then M_k is belonging to FM. If it exists a sequence σ ∈ AS / $M_0[\sigma >M_i$, as $M_k \in$ FM, the sequence $\sigma.t_{ik}$ is necessary a illegal sequence and then belongs to FS.

Now, our aim is to build an algorithm that gives the theory of regions equations. Let us assume that a list of all plant model transitions LT is given. A triplet (M_i, t_{ik}, M_k) that means $M_i[t_{ik}>M_k$ defines each transition.

Algorithm 1 : Extraction of theory of regions equations.

The marking M_i is composed of the places of different Petri Nets models.

Inputs: M_0, LT

Outputs: AS, FS, CS

Step 1 $AM \leftarrow \{M_0\}$;

Step 2 *while* $(AM \neq \varnothing)$ *do,*

Step 2.1 M_i = *next_ marking_ of (AM); AM* ← $AM - \{M_i\}$

Step 2.1.1 $T_{Mi} \leftarrow$ *transitions_from* (M_i, LT); ***Step 2.1.2*** *while* $(T_{Mi} = \varnothing)$ *do,*

Step 2.1.2.1 *MT = next_successor_in* (T_{Mi}); $M_i = (t_{ik}, M_k)$: $T_{Mi} \leftarrow T_{Mi} - \{MT\}$;

Step 2.1.2.2 $t_{ik} \leftarrow$ *transition (MT);*

Step 2.1.2.3 $M_k \leftarrow$ *marking (MT);*

Step 2.1.2.4 *(* Applying rule 2*)*

if enabled_in_plant (t_{ik}) *and not (enabled_in_specification* (t_{ik})*) then*

Step 2.1.2.4.1 $\forall \sigma \in AS / M_0[\sigma > M_i$; $FS \leftarrow FS \cup \{\sigma\ t_{ik}\}$;

Step 2.1.2.4.2 $FM \leftarrow FM \cup \{M_k\}$; *end_if;*

else if enabled_in_plant (t_{ik}) *then*

Step 2.1.2.4.3 *If not (* M_k *in PM) then*

Step2.1.2.4.3.1 $\forall \sigma \in AS / M_0[\sigma > M_i$;

$AS \leftarrow AS \cup \{\sigma\ t_{ik}\}$;

Step 2.1.2.4.3.2 $AM \leftarrow AM \cup \{M_k\}$; *end_if* *else*

Step2.1.2.4.3.3 $\forall \sigma' \in AS / M_k[\sigma' > M_i$;

$CS \leftarrow CS \cup \{\sigma' \; t_{ik}\}$;

end_if;

end_if;

end_while;

Step 2.1.3 $PM \leftarrow PM \cup \{M_i\}$;

end_while;

end.

In this algorithm, function 'next_marking_of' gives the next marking to be processed. The function 'transition_from' gives a list of doublet transition,-marking (t_{ik}, M_k), such that $M_i[t_{ik}>M_k$. As an example, if is the anterior marking of two transitions such that $M_i[t_{ik}>M_k$ and $M_i[t_{in}>M_n$ then $T_{Mi} = \{(t_{ik}, M_k), (t_{in}, M_n)\}$.

Based on this algorithm and the theory of regions we calculate the control places that must be added to the original plant PN model to implement the user specifications.

3.6. Solving of the theory of regions equations

To resolve the equations found with the theory of regions, it requires identifying the solutions that verifies the different equations. The space of solutions can be reduced by taking account additional constraints such as the nature of some transitions of the plant PN model. As an example the transitions 't_2' end 't_4' are uncontrollable. So the space state of solutions is reduced by (10) and (11):

$$C(p_c, t2) \geq 0 \qquad (10)$$

$$C(p_c, t4) \geq 0 \qquad (11)$$

To achieve the resolution, the principle consists in considering iteratively each equations of event separation with all the other equations. This means that potentially it is possible to obtain as many control places than event separation equations (see results in **Figure 4**).

a. $M_0(p^1_{c1}) = 0$ $\quad C(p^1_{C1},.) = [0 \;\; 1 \;\; -1 \;\; 0]$ $M_0(p^2_{c1}) = 0$ $\quad C(p^2_{C1},.) = [1 \;\; 0 \;\; -1 \;\; 0]$	c. $M_0(p_{c3}) = 1$ $\quad C(p_{C3},.) = [-1 \;\; 0 \;\; 1 \;\; 0]$
b. $M_0(p^1_{c2}) = 0$ $\quad C(p^1_{C2},.) = [0 \;\; 1 \;\; -1 \;\; 0]$ $M_0(p^2_{c2}) = 1$ $\quad C(p^2_{C2},.) = [-1 \;\; 1 \;\; -1 \;\; 1]$	d. $M_0(p_{c4}) = 1$ $\quad C(p_{C4},.) = [-1 \;\; 0 \;\; 1 \;\; 0]$

Figure 4. Potential Control Places

After that, the next step consists in searching the control places common to the different results. The solution $C(p_{c1},.) = C(p_{c2},.) = (0 \;\; 1 \;\; -1 \;\; 0)$ control the forbidden behavior defined by (6) and (7). Thus, it is selected to inhibit two sequences: 't_3' and '$t_1 t_3$'. A same analysis gives the solution $C(p_{c3},.) = C(p_{c4},.) = (-1 \;\; 0 \;\; 1 \;\; 0)$. Therefore finally two control places enable to restrict the initial plant model like illustrated by .

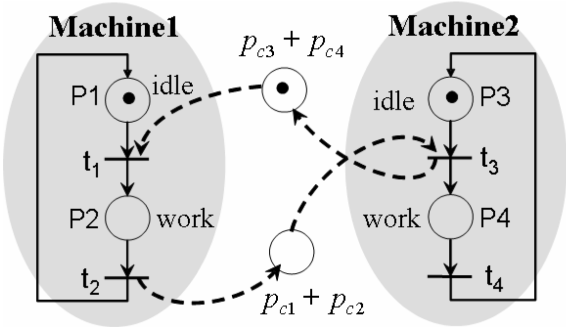


Figure 5. Closed-Loop Behavior Model

4. Modular Synthesis Approach for Resources Coordination

Nowadays, the large scale systems are modeled mostly through the composition of many smaller subsystems representing usually concurrent operations. Each of them is characterized by a partial user specification. It concerns just a part of the global plant. For this reason, modular control is a natural solution to deal with such systems. More specifically we are interested by a decentralized modular approach [10]. Indeed, [11] tries to offer reduction in computational efforts with relation to previous approaches by automata. Nevertheless, there works still requires the computation of the mutual behavior of all interacting supervisors, what may engender an exponential complexity computation. For this reason, we try to actualize the decentralized modular approach by PN. To obtain directly controllers from the entry of the plan model and its specifications, we built a software which is currently developed based on a constraint programming tool *CPLEX*, *visual C++* and *PN editor*. We propose an approach based on five steps to design PN controllers that allow implementing the legal behavior of a given plant.

The role of each step can be summarized as it follows:

Step 1 - Definition of the plant PN models and the specifications PN models that the user expresses.

Step 2 – Decomposition of the plant PN model and the specification PN model to sub systems by the projection P_i, $i = 1,\ldots m$. If the plant model is global model, divide it into several subsystems through projections P_i. Let $G_i = (N_{Pi}, M_0)$, the subsystem model. We use the generating method of subsystems proposed in [11].

Step 3 - Construction of the forbidden sequences of state-transition and the sequence of expected behavior that gives the closed-loop behavior of the plant based on the synthesis of the plant model and the user specifications. These sequences are obtained by algorithm 1 (see section III.C).

Step 4 – Obtaining the sub system controller using the theory of regions that allows adding control places in the initial plant model.

Step 5 – Obtaining the controlled system PN model. The closed behavior is denoted by $L(S/G)$ and the original marked behavior that survives under supervision is denoted by $L_m(S/G) = L(S/G) \cap L_m(G)$.

4.1. Illustration of our approach

To illustrate the methodology proposed in this paper, we use the example of an “industrial transfer line” presented in [11] (see Figure 6).

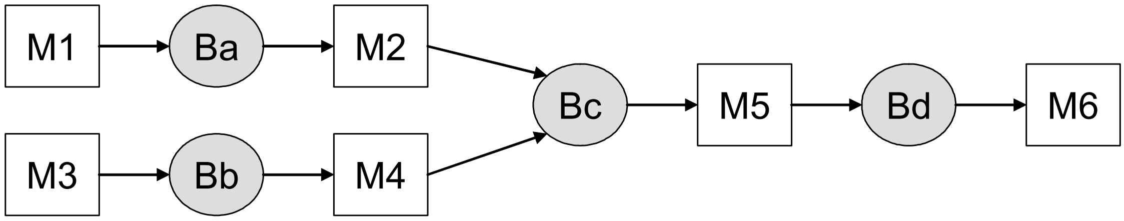
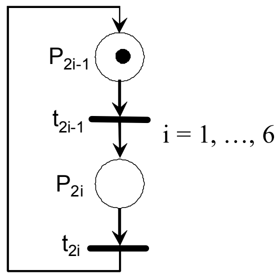
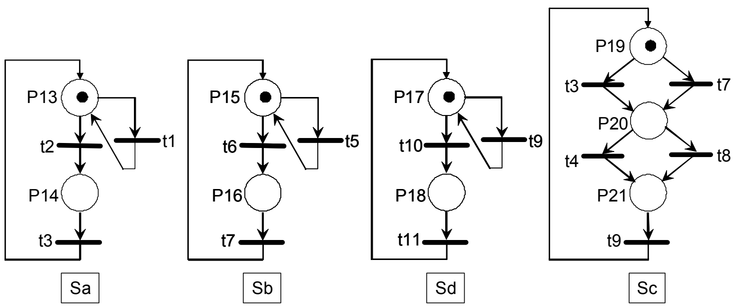


Figure 6. Industrial Transfer Line

Figure 7.A gives the plant model that consists in six PN corresponding to the unconstrained behavior of the six machines that compose the line. These six machines are connected by four buffers of unitary capacity. **Figure 7**.B gives the four specifications PN models modeling the expression of the buffers' constraints by the user (specifications Sa for the buffer Ba, and so on). By *Step 2*, there are decomposed on the process to four sub system: Ga=M1→Ba→M2, Gb=M3→Bb→M4, Gd=5→Bd→M6 and Gc = M2→Bc→M5 or M4 →Bc→M5.



A. Plant PN models



B. Specifications PN models

Figure 7. Plant PN Models and Specifications PN Models

Step 3 of our approach is based on the use of our software tool. For each subsystem as Ga, one uses *PM Editor 3.1* to write the plant PN models and the specification PN model (see

Figure 8). After that, one selects '*Get Reachability Graph*' in the menu to build the CSRG (see **Figure 9**). This result has information to construct the authorized equations, the forbidden equations and the cycle equations corresponding to the current sub system. '*Get the Equations*' is the menu to calculate the equations.

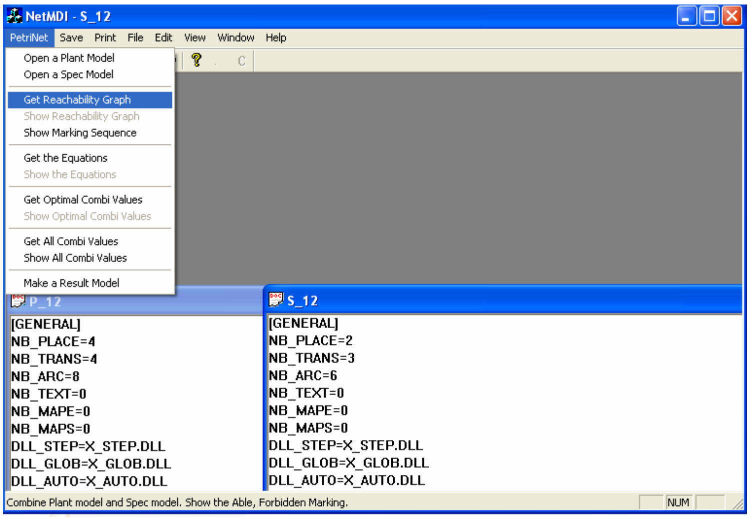


Figure 8. Open Two PN Models

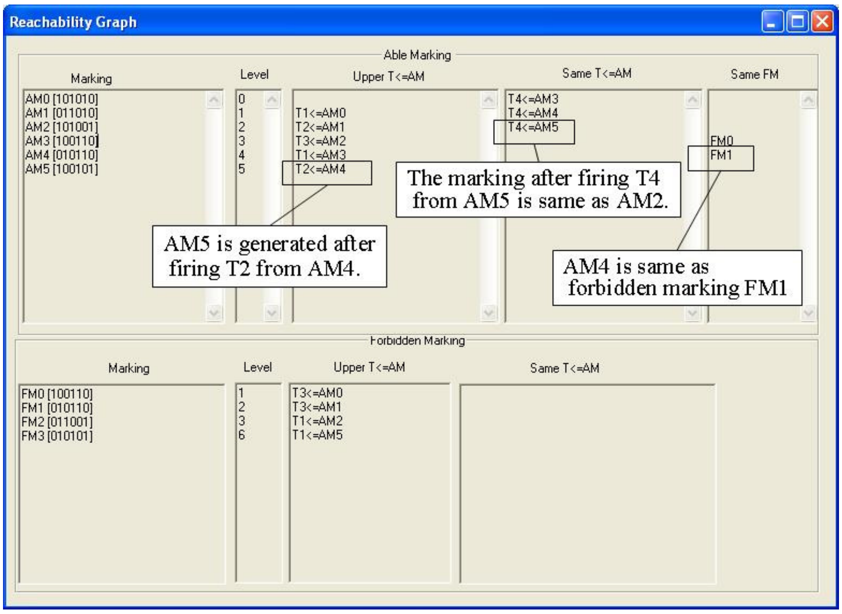


Figure 9. Get the Constraint Synchronous Reachability Graph

One gets the results like in **Figure 10**. It has six authorized equations, four forbidden equations and one cycle equation.

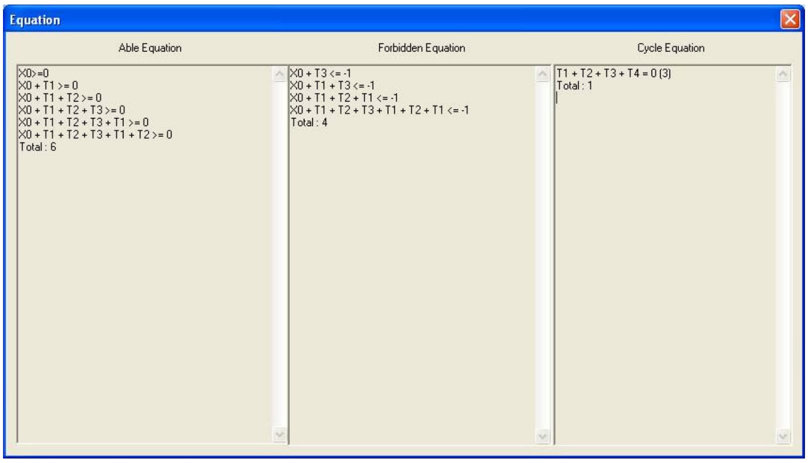
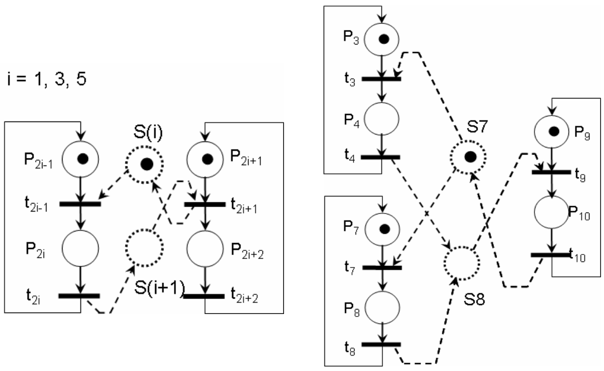


Figure 10. Authorized Equations, Forbidden Equations and Cycle Equation

And then, *Step 4* is implemented based on CPLEX optimizer which finds the controller place assuring user specification. It proposes two optimal controllers: $M_0(p_{c1}) = M_0(p_{c2}) = 0$, $C(p_{c1},.) = C(p_{c2},.) = (0 \quad 1 \quad -1 \quad 0)$ and $M_0(p_{c3}) = M_0(p_{c4}) = 1$, $C(p_{c3},.) = C(p_{c4},.) = (-1 \quad 0 \quad 1 \quad 0)$. The same method allows constructing the controller of each subsystem by *Step 3* and *Step 4* till the solution given by Fig. 11.



a. PN controllers with respect to specifications Sa, Sb and Sd

b. PN controller with respect to specification Sc

Figure 11. PN Controllers Assuring User Specification with Decentralised Synthesis

Finally, the control of the industrial transfer line taking into account the constraints induced by the buffers requires the adding of 8 controller places and 18 arcs to connect the initial plant PN models (see Fig. 12).

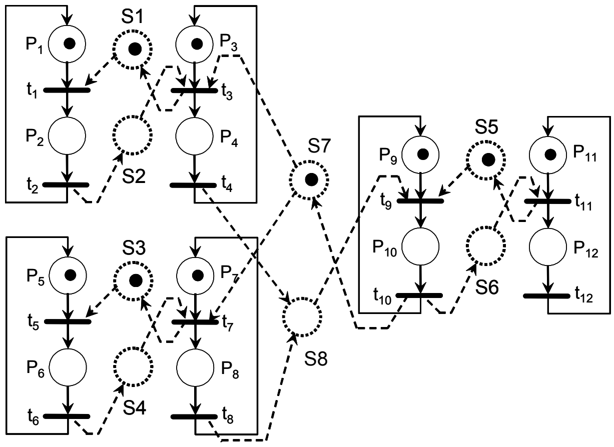


Figure 12. PN Controller of the Industrial Transfer Line Corresponding to the User Specifications

5. Conclusion

We desire that our work contribute to the approaches to synthesize DES controllers with respect to user specifications based on the supervisory approach using the Petri net formalism. In this context we propose an approach to solve the forbidden sequences of state-transitions problem (FSSTP) that is more generic than the classical forbidden state problem.

We have developed a PN tool that assists a designer to build the PN controllers corresponding to the closed-loop behavior that guarantees user specifications. In the future we want to adapt this tool to manage the dynamic reconfiguration of the control part. The idea is that the user specifications depend on the working mode of the plant, and we want to generate new controllers that optimize the control of manufacturing plant. When this tool will be automated to work directly with cyclic scheduling tool, it will become possible to reconfigure on-line manufacturing systems based on their production objectives and synthesis technique.

REFERENCES

1. KOREN, Y., U. HEISEL, F. JOVANE, T. MORIWOKI, G. PRITSHOW, A.G. ULOSY and H. VAN BRUSEEL, **Reonfigurable Manufacturing Systems**, Annals of the CIRP, Vol.48, pp. 527-540, 1999.

2. KOREN, Y., **Reconfigurable Manufacturing and Beyond**, In the summary of keynote Speech of CIRP05 3rd International Conference on Reconfigurable Manufacturing, Ann Arbor, Michigan, USA, May 10-12, 2005.

3. RAMADGE, P. J. and W. M. WONHAM, **Supervisory control of a class of Discrete Event Processes**, SIAM J. Control and Optimization, Vol. 25, No. 1, January 1987.

4. KUMAR, R. and L. E. HOLLOWAY, **Supervisory Control of Petri Net Languages**, IEEE Conference on Decision and Control, Tucson, AZ (December), pp.1190-1195, 1992.

5. GHAFFARI, A., N. REZG and X. XIE, **Algebraic and Geometric Characterization of Petri Net Controllers Using the Theory of Regions**, Proceeding of the 2002 Sixth International Workshop on Discrete Event Systems -WODES'02 (Reims, France) 2002.

6. TOGUYENI, A., **Design of Modular and Hierarchical Controllers for Reconfigurable Manufacturing Systems**, in the Proceedings of CESA'06, Beijing, China, October 4-6, 2006.

7. LEE, E. J., N. DANGOUMAU and A. TOGUYENI, **A Petri net based approach to design controllers for Flexible Manufacturing Systems**, 17th IMACS World Congress Scientific Computation, Applied Mathematics and Simulation-IMACS 2005, France, July 2005.

8. GHAFFARI, A., N. REZG and X. XIE, **Live and Maximally Permissive Controller Synthesis Using Theory of Regions**, in Synthesis and Control of Discrete Event Systems, B. Caillaud et al. (Eds.), Kluwer Academic Publishers, pp. 155-166, 2002.

9. LEE, E. J., A. TOGUYÉNI and N. DANGOUMAU, **Petri net based Controllers for Forbidden Sequences of State-Transitions in the Control of Flexible Manufacturing Systems**, Conceptual Modeling and Simulation Conference - CMS 2005, Marseilles, France, October 20-22, 2005.

10. WONHAM, W.M., **Notes on Supervisory Control of Discrete-Event Systems**: ECE 1636F/1637S, 2004-2005.

11. DE QUEIROZ, MAX H. and JOSE E.R. CURY, **Modular Supervisory Control of Large Scale Discrete Event Systems**, Proceedings of the Fifth Workshop on Discrete Event Systems (WODES 2000), Ghent, Belgium, 21-23 August, 2000.