

Supervision de Systèmes de Production par les Réseaux de Petri - Application au Pilotage d'AGV

Roberto BACOS JUNIOR¹, José CURY², Jean Jacques LOISEAU¹

¹IRCCyN - Institut de Recherche en Communications et Cybernétique de Nantes
1 Rue de la Noë, Bâtiment S – 502, 44321 NANTES Cedex 03 - France

²Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas, CEP 88040 – 900, Florianópolis, Santa Catarina - Brasil

bacos@irccyn.ec-nantes.fr
cury@das.ufsc.br
loiseau@irccyn.ec-nantes.fr

Résumé— Dans ce document est présentée une nouvelle méthode pour la conception du système de pilotage d'une flotte de chariots unidirectionnels (AGV) dans un atelier de production. L'objectif est d'éviter les accidents et les interblocages courants dans ce type de système de guidage. La motivation de ce travail vient de la complexité du problème de la synthèse de superviseurs, complexité qui augmente très rapidement en fonction du nombre de véhicules. L'emploi de réseaux de Petri pour modéliser le système permet de simplifier le calcul d'un superviseur. Celui-ci doit être complété pour garantir l'absence d'interblocage. Nous proposons une nouvelle architecture de supervision en plusieurs niveaux, répondant à cet objectif. Nous illustrons la méthode sur l'exemple d'une flotte de trois AGV.

Mots-clés— Supervision, Réseaux de Petri, Chariots Auto-guidés, Conflits, Interblocage.

I. INTRODUCTION

L'utilisation et le contrôle des systèmes dynamiques à événements discrets constituent un domaine de recherche très actif, stimulé par la diversité des possibles applications notamment dans les systèmes de flux de données, les systèmes manufacturiers, et dans les réseaux de transport de véhicules.

Dans ce document, nous nous intéressons à la coordination d'une flotte d'AGV (de l'anglais *Automated Guided Vehicles*) dans un petit atelier industriel, pour assurer les transferts des pièces en cours de fabrication entre les différents postes de travail. Leur avantage se situe au niveau de la flexibilité et la réactivité qu'ils apportent au système de production [7]. La difficulté est d'assurer la coordination des véhicules pour le transfert des pièces entre les postes de travail, en essayant d'éviter les conflits d'utilisation dans un même circuit de guidage (dans le cas des AGV, ces conflits sont des accidents et interblocages). Le problème devient très complexe lorsque le nombre de véhicules augmente.

Plusieurs méthodes de synthèse de superviseurs ont déjà été proposées pour ce genre de système, et la plupart d'entre elles sont basées sur la *Théorie de la Supervision*, introduite par Ramadge et Wonham [10], où la modélisation du système et des spécifications sont décrites par des langages et automates. La limitation de cette méthode vient

de l'utilisation d'automates. Une explosion combinatoire du nombre d'états se produit lorsque le système étudié devient complexe.

Pour essayer de réduire cette complexité, l'utilisation des réseaux de Petri (RdP) dans la modélisation du système est proposée par plusieurs auteurs [1], [2], [4], [9]. Cela permet d'avoir des modèles plus réduits, dans la plupart des cas, que les modèles par automates.

Si les réseaux de Petri ont permis de réduire la taille de la représentation d'un système, ils restent encore soumis à des problèmes de complexité. En effet, résoudre des problèmes d'accessibilité, de vivacité (*non-blocage*), de recherche de siphons d'un réseau, conduit à parcourir le graphe d'accessibilité du réseau de manière exhaustive, ce qui fait perdre tout le bénéfice de la compacité de la représentation. Plusieurs solutions ont été déjà proposées pour ce genre de problématique [3], [5], [6], [8], [9], par contre la complexité croît très vite avec la taille des systèmes étudiés.

Notre idée est de construire un superviseur capable d'interdire les situations de *conflit* (accidents et interblocages), en essayant de laisser le plus grand degré de liberté aux AGV et sans avoir la nécessité de parcourir le graphe d'accessibilité du RdP. Pour cela, nous proposons une nouvelle architecture de synthèse de superviseurs basée sur la structure du système de guidage et sa modélisation par réseaux de Petri. Cette nouvelle méthode permettra de résoudre le problème de coordination dans un système de transport par AGV, en utilisant plusieurs niveaux de supervision, où chaque niveau sera responsable pour la supervision d'une nouvelle abstraction du processus. Bien évidemment, le système supervisé final sera libre des conflits et ira respecter les spécifications de fonctionnement imposées par le cahier de charges du système.

Dans la section suivante nous montrerons quelques rappels de la *Supervision par les Réseaux de Petri*, ainsi que notre nouvelle architecture pour la supervision d'un système d'AGV. Ensuite, nous allons appliquer ces résultats à un exemple académique de complexité expressive, suivi de quelques commentaires sur notre contribution et les perspectives futures.

II. CONCEPTS DE LA SUPERVISION PAR LES RDP

A. Introduction

Dans le cadre de la théorie classique de la supervision de Ramadge et Wonham [10], les éléments de base d'un problème de supervision sont le *modèle du procédé* et l'expression des *spécifications* que le superviseur à concevoir doit permettre de valider.

Lorsque le procédé est modélisé par un réseau de Petri \mathcal{P} , les spécifications sont données par des *inégalités linéaires sur le marquage*, de la forme :

$$L \cdot \mu_p \leq b \quad (1)$$

où :

- $L \in \mathbb{N}^{n_c \times m}$ est la matrice de contraintes¹ ;
- $b \in \mathbb{N}^{n_c}$ est un vecteur ;
- m est le nombre de places du réseau de Petri ;
- n_c est le nombre de contraintes ;
- $\mu_p \in \mathbb{N}^m$ est le marquage du procédé.

Ces contraintes sont quelquefois appelées *contraintes d'exclusion mutuelle généralisées* [1].

Étant donné l'ensemble des spécifications, c'est-à-dire, des contraintes sous la forme d'une matrice L et d'un vecteur b , nous allons définir l'ensemble des *marquages légaux* par :

$$M(L, b) = \{\mu_p \in \mathbb{N}^m \mid L \cdot \mu_p \leq b\} \quad (2)$$

Donc, le but d'un superviseur Rdp est de garantir, au cours du temps, que le vecteur de marquage du réseau de Petri reste dans l'ensemble des *marquages légaux* $M(L, b)$, sous l'hypothèse que le marquage initial lui-même est *légal*.

Les marquages ne satisfaisant pas l'inégalité (1) s'interprètent comme des *marquages interdits*, que le procédé à superviser ne doit jamais atteindre.

Le *formalisme matriciel* de la supervision par les réseaux de Petri permet un calcul plus simple d'un superviseur que dans le cas classique, où les modèles utilisés sont des automates et des langages formels. De plus, le *produit synchrone* utilisé dans le cas des automates n'est pas nécessaire dans les Rdp, car tout est joué dans les marquages du système. Par conséquent, le graphe des marquages accessibles du système supervisé est réduit, comme la complexité du modèle.

B. Supervision Basée sur les Invariants de Place

La méthode de la supervision basée sur les invariants de place, introduite par Moody, Yamalidou, Lemmon et Antsaklis [9], aussi appelée SBIP, présente une technique pour le contrôle des réseaux de Petri où la spécification est décrite, comme expliqué précédemment, par un système d'inégalités. La présentation est faite en supposant le cas le plus simple : *aucune concurrence, contrôlabilité et observabilité totale du réseau*.

Étant donné un modèle du procédé sous la forme d'un réseau de Petri \mathcal{P} , le but du contrôleur est de *restreindre* le comportement du procédé afin que seuls les marquages qui font partie de l'ensemble des marquages légaux puissent être atteints, en d'autres termes, le superviseur SBIP doit

forcer le procédé à obéir à l'ensemble de contraintes du type (1).

L'idée de la méthode SBIP est que cette contrainte d'inégalité sur le marquage peut être transformée en une égalité en introduisant une variable d'écart non négative μ' . Alors, pour un ensemble de contraintes sur le marquage, nous avons :

$$L \cdot \mu_p + \mu' = b \quad (3)$$

Cette variable d'écart μ' représente un vecteur de marquage de dimension n_c . Chacun des éléments de ce vecteur représente le marquage d'une nouvelle place p_i ($i = 1, \dots, n_c$) qui va être ajoutée au réseau de Petri \mathcal{P} du procédé, et dont le rôle est de diriger les jetons supplémentaires afin que la contrainte d'inégalité correspondante soit tout le temps satisfaite. L'ensemble des places p_i assure donc que la somme pondérée des jetons dans les places du réseau, qui correspondent à la spécification donnée, est toujours inférieure ou égale à b_i .

Ces places qui servent à maintenir l'inégalité satisfaite sont appelées *places de supervision*, elles font partie d'un nouveau réseau de Petri qui représente le superviseur \mathcal{S} . La structure de ce réseau est calculée en observant que l'introduction des variables d'écarts introduit des *invariants de place* dans le système contrôlé défini par l'équation (3).

Définition 1 (Matrice d'incidence du Procédé) - La matrice d'incidence d'un réseau de Petri \mathcal{P} peut être définie comme une matrice $\mathcal{C}^{\mathcal{P}}$ où chaque élément $\mathcal{C}^{\mathcal{P}}_{ij}$ de cette matrice peut être défini par :

- $\mathcal{C}^{\mathcal{P}}_{ij} > 0$, s'il existe un arc reliant une transition t_j à une place p_i ;
- $\mathcal{C}^{\mathcal{P}}_{ij} < 0$, s'il existe un arc reliant une place p_i à une transition t_j ;
- $\mathcal{C}^{\mathcal{P}}_{ij} = 0$, s'il n'existe pas d'arc entre la transition t_j et la place p_i (ou dans le cas de l'existence de boucle).

Le superviseur introduit par la méthode SBIP se définit en terme d'une matrice d'incidence $\mathcal{C}^{\mathcal{S}}$. Cette matrice vient comme une *complétion* des lignes de la matrice d'incidence du procédé $\mathcal{C}^{\mathcal{P}}$.

Définition 2 (Matrice d'Incidence de Supervision) - Soit une matrice d'incidence du procédé ($\mathcal{C}^{\mathcal{P}}$) et des contraintes à satisfaire par ce procédé (L et b), le superviseur associé est donné sous la forme d'un Rdp défini par :

$$\mathcal{C}^{\mathcal{S}} = -L \cdot \mathcal{C}^{\mathcal{P}} \quad (4)$$

Et pour son marquage initial :

$$\mu_s^0 = b - L \cdot \mu_p^0 \quad (5)$$

Donc, la matrice d'incidence du *système supervisé* \mathcal{C} sera donnée par :

$$\mathcal{C} = \begin{bmatrix} \mathcal{C}^{\mathcal{P}} \\ \mathcal{C}^{\mathcal{S}} \end{bmatrix}$$

Lorsque le coefficient de cette matrice en position $\mathcal{C}^{\mathcal{S}}_{ij}$ est positif, cela correspond à un arc allant de la transition t_j à la i -ème des places de supervision. Lorsqu'au contraire le coefficient $\mathcal{C}^{\mathcal{S}}_{ij}$ est négatif, il correspond à un arc allant de la i -ème des places de supervision à la transition t_j .

¹ \mathbb{N} est l'ensemble des nombres entiers.

C. Nouvelle Méthode de Synthèse des Superviseurs

La méthode SBIP ne garantit pas la non-occurrence d'*interblocages*, comme nous pouvons observer dans la *Figure 1*. Les AGV n'ont pas la possibilité d'évoluer dans le système, donc, ils se trouvent dans un *blocage mortel* (deadlock).

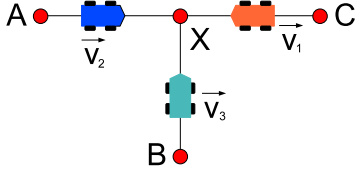


Fig. 1. Deadlock dans une Intersection X

Les méthodes proposées pour résoudre cette question d'interblocage sont basées sur la recherche et le calcul des *siphons élémentaires* du réseau de Petri qui modélise le système [3] [6]. Néanmoins, cette méthode devient très chère sur le plan de la complexité algorithmique pour des systèmes de grande taille.

Nous proposons une nouvelle méthodologie de synthèse des superviseurs RdP où la question des interblocages sera traitée par une simple analyse de la *topologie* du réseau sans avoir besoin de rechercher les siphons. Nous nous intéressons alors aux *intersections* du circuit de guidage où nous vérifions l'existence des *zones critiques*, dans lesquelles les interblocages sont observés.

L'idée de base est qu'à partir d'une nouvelle *abstraction* du procédé, une abstraction de plus haut niveau, le nouveau superviseur va gérer l'accès à ces *zones critiques*, en évitant ainsi les interblocages mortels observés auparavant. Alors, nous nous intéressons à la résolution de ces conflits dans ces *Zones de Conflit*.

Définition 3 (Zone de Conflit) - Une *Zone de Conflit* \mathcal{Z}_X est formée par l'ensemble des toutes possibles trajectoires (places du RdP) entrantes dans une même intersection X (nœud du système). Cette compétition pour l'occupation du moyen *intersection* par les AGV du système caractérise un conflit potentiel et peut devenir un *interblocage*, donc, un superviseur $\mathcal{S}_{\mathcal{Z}_X}$ devra contrôler l'accès à cette zone.

Nous allons définir l'ensemble des places qui font partie d'une telle Zone de Conflit X du système de la *Figure 1* :

$$\mathcal{Z}_X = \{AX, BX, CX\}$$

Donc, l'accès à cette Zone de Conflit \mathcal{Z}_X sera toujours contrôlé par la place de supervision $\mathcal{S}_{\mathcal{Z}_X}$ où un nombre maximal d'AGV pourra y accéder. Cette limitation du nombre d'AGV est donnée par le *nombre de jetons* dans $\mathcal{S}_{\mathcal{Z}_X}$, qui représente l'ensemble total de ressources disponibles (nombre de trajectoires possibles) dans cette zone, **moins une ressource**, car le superviseur doit *réserver* une de ses ressources disponibles pour la sortie des AGV qui occupent déjà cette zone.

Dans un cas plus général, à partir de l'identification de toutes les Zones de Conflit d'un réseau, nous pouvons créer un système de contraintes sur le marquage formé par l'ensemble des places de chaque Zone de Conflit. Ensuite, les

superviseurs de chaque zone seront calculés en utilisant la méthode SBIP présenté dans la section précédente.

À chaque augmentation du nombre d'AGV dans le système, un nouveau niveau de supervision devra être mis en place. Donc, nous devons traiter les *interactions* entre ses Zones de Conflit afin de garantir toujours une ressource disponible pour la sortie des AGV dans les zones. Un exemple plus détaillé sera traité dans le chapitre suivant où cette question sera abordée.

Cette procédure conduit à la conception d'un système supervisé avec une architecture hiérarchisée, comme celle présentée dans la *Figure 2*, où chaque superviseur garantira le non-blocage dans son niveau de supervision.

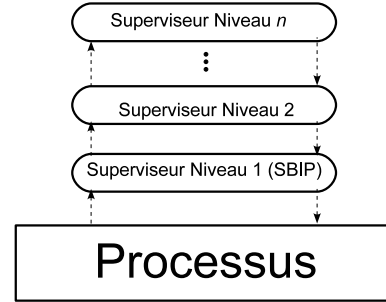


Fig. 2. Système Supervisée avec n niveaux de Supervision

III. PILOTAGE D'UN SYSTÈME D'AGV

A. Les Circuits de Guidage

Dans certains systèmes de production, les différentes stations de travail sont reliées entre elles par des *circuits de guidage*, nous nous intéressons aux circuits de guidage appelés *bidirectionnels*, dont les AGV sont autorisés à traverser les liens du circuit dans les deux sens. De tels circuits ont comme grand avantage la réductions des distances parcourues par les AGV, ainsi que une effective réduction dans le nombre des AGV dans le système vu que l'atteignabilité du circuit augmente et les tâches sont effectuées dans un temps normalement plus petit en comparaison avec les circuits unidirectionnels. Cependant, il y a une augmentation considérable des risques d'accidents et d'interblocages entre AGV qui devra être traité par un *superviseur*.

Des problèmes de *conflits* sont souvent confrontés dans les circuits à boucles multiples vu que les liens que relient les stations de travail sont considérés comme des *ressources unitaires*, donc limitées. À partir du moment où une ressource (tronçon du système de guidage) est accordée à un AGV, elle devient *indisponible* aux prochaines demandes d'occupation par d'autres AGV jusqu'à sa libération complète par le véhicule qui la détient. Cette compétition pour l'accès aux ressources libres peut devenir un interblocage *total*, si l'ensemble des ressources sont immobilisés (*deadlock*), ou *partiel* si une seule partie des ressources est bloquée (*livelock*).

B. Exemple d'Atelier Industriel

Nous nous intéressons alors à la gestion du *pilotage* d'un système d'AGV. Pour cela, nous considérons le petit atelier de production de la *Figure 3*. Les stations de travail W_1 et W_2 sont considérées comme étant des machines de production et le poste W_0 correspond à la station de départ des

AGV (garage). Les 3 AGV responsables pour le transfert des pièces entre les postes de travail sont aussi représentés (V_i , pour $i = 1, 2, 3$).

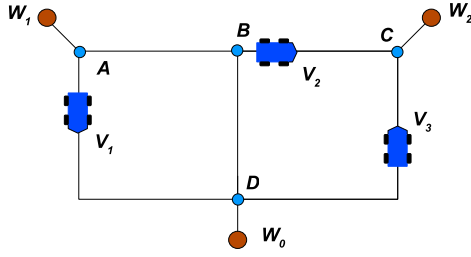


Fig. 3. Atelier de Production avec 3 AGV

Nous supposons que les AGV sont *unidirectionnels*, c'est-à-dire qu'à partir du moment où un d'entre eux s'engage dans un tronçon du système de guidage il ne peut plus revenir en arrière et il doit arriver au bout du tronçon. Par contre, nous considérons les tronçons comme des circuits *bidirectionnels*, c'est-à-dire les AGV peuvent circuler dans les deux directions possibles du tronçon. Cette spécification augmente la complexité de la supervision du système vis-à-vis des accidents et interblocages potentiels occasionés par ce type de configuration.

Pour une bonne évolution du système il est nécessaire d'imposer des *contraintes de fonctionnement* pour garantir les spécifications initiales du procédé. Donc, la coordination (superviseur) du système doit interdire :

- chaque état tel que 2 AGV ou plus transitent un même tronçon du système ou occupent un même poste de travail (*safeness*).

De même, cette coordination doit aussi assurer que :

- le système soit *non-bloquant*, c'est-à-dire qu'il doit éviter les interblocages (*deadlock*, impossibilité d'un AGV de se déplacer) et la famine (*livelock*, impossibilité d'un AGV d'atteindre le prochain poste de travail) ;
- le prochain événement généré par le système doit être parmi la liste des événements permis, c'est-à-dire, nous ne pouvons pas atteindre un état du système qui peut amener à un état interdit.

C. Modèle RdP du Processus

Étant donné le *cahier de charges* avec toutes les contraintes de fonctionnement du système, la modélisation de l'atelier de production de la *Figure 3* s'effectue par un modèle RdP. Chaque tronçon du système représente une place dans le RdP du procédé, et chaque transition représente l'occurrence d'un changement de tronçon par les AGV. Nous considérons que les trois AGV sont disposés initialement dans le garage W_0 comme indiqué dans la *Figure 4*.

Nous ordonnons les *treize* places du RdP de la *Figure 4* dans l'ordre suivante : ($W_0, W_1, W_2, DA, DB, DC, AD, BD, CD, AB, BA, BC, CB$).

Un vecteur de marquages initial μ_0 possède *treize* composantes qui correspondent aux *treize* places rangées dans cet ordre.

$$\mu_0^T = [3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

À partir du modèle RdP du processus présenté dans la *Figure 4*, nous pouvons construire sa Matrice d'Incidence

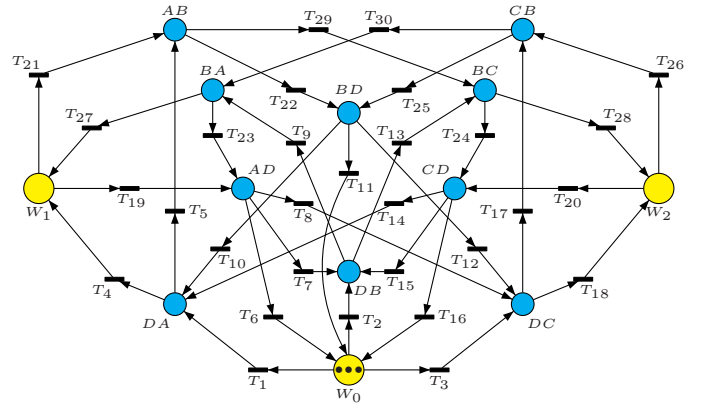


Fig. 4. Le modèle RdP du procédé avec 3 AGV

du Procédé \mathcal{C}^P , de dimension 13×30 (où 13 est le nombre de places du RdP et 30 son nombre de transitions).

D. Le Calcul des Superviseurs

Pour appliquer la méthode SBIP, nous pouvons décrire les spécifications imposées au procédé par un *système de contraintes d'inégalités* du type :

$$L \cdot \mu_p \leq b$$

Pour garantir la spécification d'exclusion mutuelle dans chaque branche du système, nous décrivons le système d'inégalités suivant avec *huit* inégalités :

$$\begin{cases} \mu_{W_0} \leq 3 \\ \mu_{W_1} \leq 1 \\ \mu_{W_2} \leq 1 \\ \mu_{AD} + \mu_{DA} \leq 1 \\ \mu_{BD} + \mu_{DB} \leq 1 \\ \mu_{CD} + \mu_{DC} \leq 1 \\ \mu_{AB} + \mu_{BA} \leq 1 \\ \mu_{BC} + \mu_{CB} \leq 1 \end{cases}$$

La première équation va limiter le nombre d'AGV dans le garage, les deux inégalités suivantes vont garantir l'utilisation exclusive de chaque poste de travail, et les autres cinq inégalités garantissent l'exclusion mutuelle dans chaque tronçon du système.

À partir de la création du système d'inégalités, nous appliquons l'idée de la méthode SBIP qui consiste à transformer les inégalités en égalités, en introduisant des variables d'écart non négatives μ' dans chaque inégalité, donc le système de contraintes devient ainsi :

$$\begin{cases} \mu_{W_0} + \mu_{S_1} = 3 \\ \mu_{W_1} + \mu_{S_2} = 1 \\ \mu_{W_2} + \mu_{S_3} = 1 \\ \mu_{AD} + \mu_{DA} + \mu_{S_4} = 1 \\ \mu_{BD} + \mu_{DB} + \mu_{S_5} = 1 \\ \mu_{CD} + \mu_{DC} + \mu_{S_6} = 1 \\ \mu_{AB} + \mu_{BA} + \mu_{S_7} = 1 \\ \mu_{BC} + \mu_{CB} + \mu_{S_8} = 1 \end{cases}$$

Remarque 1: Le nombre de contraintes initiales à satisfaire donne le nombre de places de supervision *ajoutées* dans le RdP initial.

L'ensemble des contraintes sur les marquages sous la forme d'égalités est défini par :

$$L \cdot \mu_p + \mu' = b$$

Place de Supervision	Transitions d'Entrée ($\bullet p$)	Transitions de Sortie ($p \bullet$)
S_{Z_A}	$T_5, T_{19}, T_{21}, T_{23}$	$T_1, T_9, T_{10}, T_{14}, T_{30}$
S_{Z_B}	$T_9, T_{13}, T_{22}, T_{25}, T_{29}, T_{30}$	$T_2, T_5, T_7, T_{15}, T_{17}, T_{21}, T_{26}$
S_{Z_C}	$T_{17}, T_{20}, T_{24}, T_{26}$	$T_3, T_8, T_{12}, T_{13}, T_{29}$
S_{Z_D}	$T_1, T_2, T_3, T_7, T_8, T_{10}, T_{12}, T_{14}, T_{15}$	$T_{19}, T_{20}, T_{22}, T_{23}, T_{24}, T_{25}$

TABLE I

LISTE DES CONNECTIONS ENTRE LES PLACES DE SUPERVISION ET LES TRANSITIONS DU RDP DU PROCÉDÉ

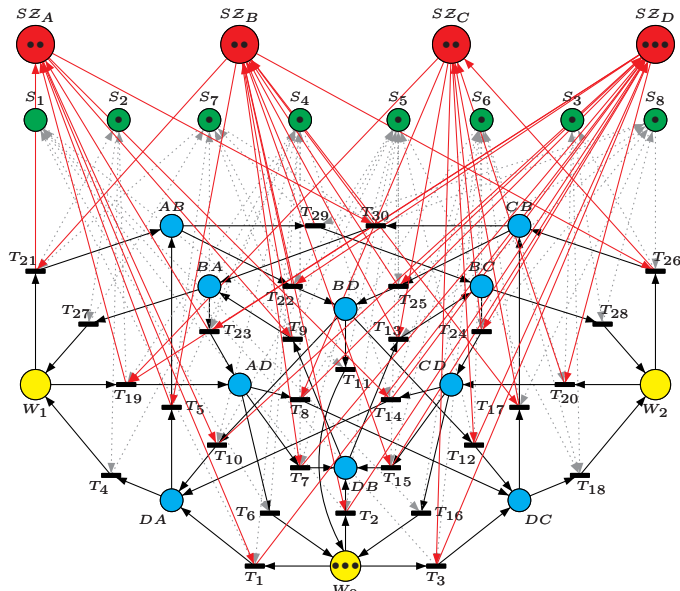


Fig. 7. Le Système Supervisé Final

occupées, ainsi que la place CB, le superviseur S_{Z_A} ne permettra plus l'entrée dans Z_A et il réservera la ressource AB comme sortie pour l'un de ses deux AGV. Le superviseur S_{Z_B} , quant à lui, possède toujours une ressource disponible mais un conflit apparaît si cette même ressource BA est déjà réservée par S_{Z_A} (BA et AB forment le même tronçon), donc, si un 4^e AGV apparaît dans la place DB (qui est une situation parfaitement possible vis-à-vis de S_{Z_B}). Le système devient alors bloqué car les deux superviseurs réservent la même ressource pour la sortie de ses AGV.

Pour éviter cela, nous aurons besoin de cinq places de supervision de 3^e niveau pour garantir les limitations du nombre d'AGV dans deux Zones de Conflit adjacentes ($S_{Z_{AB}}, S_{Z_{AD}}, S_{Z_{BC}}, S_{Z_{BD}}$ et $S_{Z_{CD}}$). Nous remarquons qu'entre Z_A et Z_C nous n'avons pas d'interactions directes, remarquées aussi à travers la Figure 6.

Alors, si le nombre d'AGV augmentait encore une fois d'une unité, nous aurions à calculer les interactions entre trois zones adjacentes et nous aurons quatre autres places de supervision ($S_{Z_{ABC}}, S_{Z_{ABD}}, S_{Z_{ACD}}$ et $S_{Z_{BCD}}$) d'un 4^e niveau de supervision.

IV. CONCLUSION

Le pilotage d'un système d'AGV devient très complexe lorsque le nombre de véhicules et les possibles trajectoires augmentent. Pour éviter les conflits d'utilisation des ressources du système, un superviseur (ou plusieurs) est proposé pour interdire certaines situations indésirables notées dans le cahier de charges du système. Nous proposons une

nouvelle architecture pour la synthèse des superviseurs afin de garantir les spécifications initiales de non-blocage et sans accidents. Cette méthode se base sur la topologie du réseau (les *intersections*), sans avoir la nécessité de parcourir le réseau à la recherche des siphons. L'exclusion mutuelle garantie par les invariants de place et le traitement des conflits par les **Zones de Conflits** vont garantir que les spécifications soient respectées et le système n'aura pas d'interblocage.

Des comparaisons entre les résultats de la synthèse des superviseurs de l'atelier de la Figure 3 ont été faites, entre l'utilisation des automates et de la Théorie de la Supervision de Ramadge-Wonham et entre notre nouvelle méthode de synthèse. D'après l'utilisation des automates, le système supervisé possède p états et t transitions dans l'automate correspondant, soit $n = 1.387$ et $t = 6.858$, tandis que notre méthode donne un système supervisé de 25 places et 30 transitions.

Les perspectives de ce travail se situent à deux niveaux. Le premier est le développement d'un algorithme pour l'implémentation de notre méthode, le second est la preuve de la *permissivité maximale* du superviseur calculé par notre méthode.

RÉFÉRENCES

- [1] A. GIUA, F. DICESARE, M. SILVA, « Petri Net Supervisors for Generalized Mutual Exclusion Constraints, » *Proc. of 12th IFAC World Congress*, Sydney, Australia, pp. 267 – 270, 1993.
- [2] M. V. IORDACHE and P. J. ANTSAKLIS, « Supervision Based on Place Invariants : A Survey, » *Technical Report of the ISIS Group at the University of Notre Dame*, 2004.
- [3] M. V. IORDACHE and P. J. ANTSAKLIS, « Design of T-Liveness Enforcing Supervisors in Petri Nets, » *IEEE Transactions on Automatic Control*, Vol. 48, pp. 1962 – 1974, 2003.
- [4] B. H. KROGH, « Controlled Petri Nets and Maximally Permissive Feedback Logic, » *Proc. 25th Allerton Conf. on Communications, Control and Computing*, Illinois, pp. 317 – 326, 1987.
- [5] Z. LI, M. ZHAO and N. WU, « A Survey and Comparison of Petri Net-Based Deadlock Prevention Policies for Flexible Manufacturing Systems, » *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 38, pp. 173 – 188, 2008.
- [6] Z. LI, M. ZHAO and R. ZHU, « Supervisory Control of a Modular Class of Discrete Event Systems Based on Elementary Siphons of Petri Nets, » *International Conference on Control and Automation*, Budapest, Hungary, pp. 1181 – 1186, 2005.
- [7] S. MIRDAMADI, « Pilotage d'un atelier de production en temps réel à base de simulation de flux à événements discrets, » 8^e Congrès des Doctorants de l'École Doctorale Systèmes de Midi-Pyrénées EDSys, Albi-Carmaux, France, 2007.
- [8] J. O. MOODY, P. J. ANTSAKLIS, *Supervisory Control of DES Using Petri Nets*, Kluwer Academic Publishers, 1998.
- [9] J. O. MOODY, K. YAMALIDOU, M. D. LEMMON, P. J. ANTSAKLIS, « Feedback Control of Petri Nets Based on Place Invariants, » *Proceedings of 33rd IEEE Conference on Decision and Control*, Lake Buena Vista, Florida, pp. 3104 – 3109, 1994.
- [10] P. J. RAMADGE and W. M. WONHAM, « Supervisory Control of a Class of Discrete Event Processes, » *SIAM Journal of Control and Optimization*, Vol. 25, pp. 206 – 230, 1987.