

APPROCHE DE LA SURVEILLANCE DES SYSTEMES PAR RESEAUX DE PETRI SYNCHRONISES FLOUS

Eugenia MINCA*, Florin FILIP**, Ryad ZEMOURI***

*Faculté de Génie Electrique de l'Université Valahia,
Bd. Unirii, nr. 18, 0200, Targoviste, Roumanie
minca@valahia.ro

**Académie Roumaine
Calea Victoriei 125, Sect. 1, București, Roumanie
ffilip@acad.ro

***Laboratoire d'Automatique du CNAM
21 Rue Pinel, 75013 Paris
Ryad.Zemouri@paris.ensam.fr

Résumé. Dans cet article, nous proposons un nouvel outil de modélisation des systèmes à événements discrets avec fonctionnement non autonome qui intègre la logique floue dans un outil de type RdP T-temporisés. Les connaissances incertaines associées à cette activité, exigent des raisonnements spécifiques et des méthodes de modélisation adaptés à une logique différente de la logique des connaissances précises. La logique floue offre le cadre pour l'exploitation des connaissances floues, qui du point de vue qualitatif donne des résultats plus fins que la logique classique. Dans ce contexte, le réseau de Petri flou synchronisé flou (RdPSinF) sera bien adapté dans la modélisation des fonctions détection/décision par une approche transitions-temporisés floue.

Mots clés : détection floue, logique floue, raisonnement flou, réseaux de Petri T-temporisés.

1. INTRODUCTION

Les connaissances incertaines associées à la surveillance des systèmes à événements discrets, exigent des raisonnements spécifiques et des méthodes de modélisation adaptées à une logique différente de la logique classique [1, 6, 7, 8]. Dans la catégorie des outils de modélisation, le réseau de Petri flou est un outil très approprié pour l'étude des systèmes à événements discrets décrits par de telles connaissances imprécises. Un état de l'art très complet des différentes approches de réseaux de Petri flous (RdPF) a été publié par J. Cardoso et B. Pradin-Chezalviel dans [3].

Les différents types de logiques (classique, linéaire et floue) utilisées dans la description des systèmes, génèrent deux grandes catégories de modèles RdPF : la première classe de modèles est représentée par les Systèmes Experts Flous. Dans ce cas, le RdPF est interfacé avec le système surveillé par les informations fournies par les capteurs et représente l'équivalent d'un contrôleur flou pour un système à événement discret [9, 17]. Une autre classe d'applications est modélisée par les RdPF qui expriment l'imprécision ou le flou des connaissances [2, 20]. Ce dernier type de RdPF modélise un système physique en appliquant la logique linéaire [3, 17] au niveau des transformations des ressources. En général, les RdPF utilisent la logique classique [4, 21, 8] au niveau de la sémantique des propositions logiques modélisées. Notre contribution se situe dans la première classe de RdPF.

Pour modéliser la fonction *de surveillance*, on utilise une extension du RdPF [12,13,14] qui intègre par l'aspect temporelle - l'instant d'apparition des défauts dans le système surveillé. Le RdPFS [15,16] est orienté pour la modélisation d'une base des règles logiques floues qui découle de l'expression logique de *l'arbre des défaillances* (ADD), identifié a priori au niveau du système surveillé. L'outil RdPFS modélise la réunion ou l'intersection des raisonnements logiques, en respectant les concepts spécifiques de la logique floue [1, 6]. L'aspect analyse nous offre une information affinée au niveau de chaque défaut par le transfert

des signaux des défauts temporellement synchronisés. Les RdPFS mettent en évidence les caractéristiques de certains points critiques qui matérialisent le *chemin critique* dans la stratégie de la fonction *pronostic*.

La limite de cet outil représente le fonctionnement du réseau : les signaux externe arrivent et valident les transitions simultanément. Même si l'événement externe se passe a un moment donné, il sera injecte dans le réseau, au moment où elles sont lu par l'interface de détection.

Dans ce but, nous proposons un autre outil - réseau de Petri synchronisé flou (RdPSinF). Le RdPSinF décrit le fonctionnement des systèmes qui ne sont pas autonomes. Leur fonctionnement est conditionné par des événements externes et par le temps. On propose un outil qui peut quantifier les relations conditionnelles externes par l'intermédiaire des transitions réceptives aux événements externes. Le même temps, par l'approche flou, on raffine l'information externe de point de vu la fenêtre temporelle ou l'événement est attendu se passer.

La modélisation du système à évènements discrets de surveillance peut être faite par différents types de réseaux de Petri (les RdP ordinaires, les RdP de haut niveau), en partant de l'hypothèse que les défauts possibles sont connus a priori et modélisés par les mécanismes spécifiques (par exemple « *le chien de garde* » [5]). Notre approche prend en considération une modélisation avec un réseau de Petri temporisé au niveau des transitions.

2. RÉSEAUX DE PETRI SYNCHRONISÉS FLOUS (RPSinF)

Nous proposons une variante de réseau de Petri flou inspirée des travaux de Chen [4], adapté au besoin de la modélisation des fonctions de surveillance. Le modèle de Chen est adapté à la modélisation de bases d'informations logiques statiques. Cette approche n'est pas entièrement satisfaisante pour la surveillance dynamique. C'est dans ce sens que nous proposons le réseau de Petri Petri synchronisé flou (RdPSinF) comme une extension des RdPF. Pour modéliser les fonctions de détection/diagnostic nous définissons un modèle flou capable d'intégrer l'instant d'apparition des défauts du système surveillé.

2.1. Définition de l'outil

Les réseaux de Petri synchronisé flou (RdPSinF) sont un outil défini comme étant le n-uplet

$RdPSinF = \langle P, T, E, I, O, F, Sinc, D, M_0 \rangle$ avec :

$P = \{ p_1, p_2, \dots, p_n \}$ ensemble fini des places ;

$T = \{ t_1, t_2, \dots, t_m \}$ ensemble fini des transitions;

$E = \{ E_1, E_2, \dots, E_i \}$ ensemble fini de évènements externes;

$I : T \rightarrow P$ fonction d'entrée dans les places ;

$O : P \rightarrow T$ fonction de sortie des places ;

$F(t) : T \rightarrow [0,1]$ fonction associative qui établi une valeur de crédibilité $\mu = F(t)$ variable dans le temps pour chaque transition $t_i \in T$. μ représente le degré de vérité de la proposition correspondant à la transition. L'instant t correspond à l'instant temporel $t \in d$ quand l'événement externe E_i sera réceptionné par le système modélisé.

$Sinc : T \rightarrow E \cup e$ est une application sur l'ensemble des transitions avec des valeurs sur l'ensemble des événements E réuni avec l'événement e qui este l'événement avec apparition permanente

$D = \{ d_1, d_2, \dots, d_i \}$ ensemble de durées associées aux événements externes qui représente la fenêtre temporelle pour leurs réceptions. Ces durées représentent le même temps que les temporisations associées aux transitions synchronisées avec des événements externes.

M_0 marquage initial du réseau.

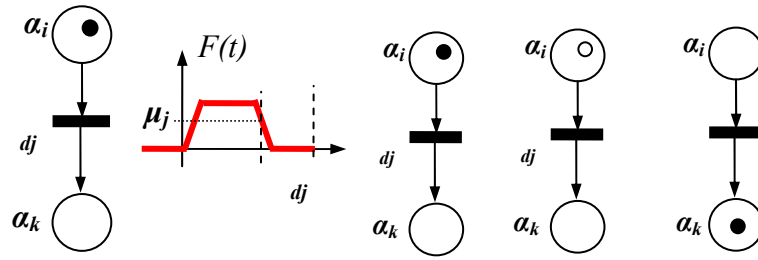


Figure 1

Dans le réseau *RdPSinF*, quand une *transition est valide*, elle peut être franchissable si l'événement associé par *Sinc* sera réceptionnée. On suppose que le système surveillé est modélisé par un RdP temporel (RdPT) qui transmette les informations comme événement externe. L'événement est attendu dans la fenêtre temporelle d_i . Pendant la durée d_i , le jeton a le statut de *jeton réservé pour la transition T_j* (Figure 1). C'est l'aspect T-temporisé du réseau *RdPSinF*. A l'instant où il se produit, la transition transfère au jeton la valeur $\mu_j = F(t)$ qui représente le degré de crédibilité associé à la possibilité d'apparition de l'événement externe. Le jeton qui arrive dans la place suivante, sera chargé avec la valeur floue $\alpha_k = \alpha_i \cdot \mu_j$ (Figure 1). L'outil *RdPSinF* est bien adapté pour la modélisation des règles logiques que pour la modélisation des ressources des systèmes de productions. Dans les deux cas, le réseau a des règles de fonctionnement différentes. Dans la modélisation sont possibles les différentes structures du réseau Petri (Figures 2 a,b,c).

2.2. RdPSinF pour la modélisation des règles logique. Fonctionnement.

Chaque transition correspond à une règle floue et a associée une fonction $\mu_j = F(t)$ qui décrit l'instant de franchissement possible. La fonction $\mu_j = F(t)$ représente la fonction d'appartenance de la variable floue t à

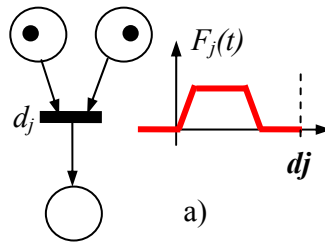


Figure 2.a) Modélisation de la règle logique

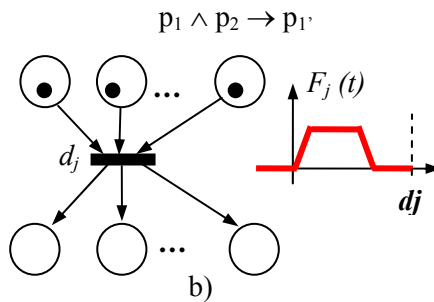


Figure 2.b) Modélisation de la règle logique

$$p_1 \wedge p_2 \wedge \dots \wedge p_i \rightarrow p_1' \wedge p_2' \wedge \dots \wedge p_k'$$

L'ensemble flou défini par la variable linguistique "apparition d'événement E_i ". Etant variable dans le temps, la valeur de crédibilité μ , correspondant à la règle modélisée par la transition, imprime à chaque règle un caractère de *crédibilité dynamique*. L'intervalle $[0 d_j]$ représente la durée totale analysée et en même temps l'intervalle de temporisation associé aux transitions.

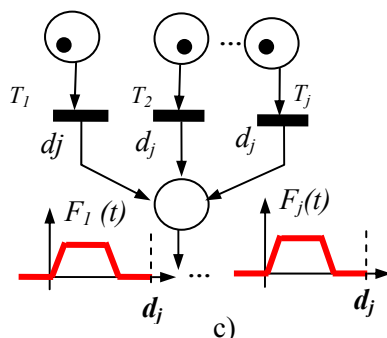


Figure 2.c) Modélisation de la règle logique
 $(p_1 \rightarrow p_k) \vee (p_2 \rightarrow p_k) \vee \dots \vee (p_i \rightarrow p_k)$

Même si l'événement se passe dans la fenêtre temporelle $[0, d_j]$, le tir de la transition sera après la durée d_j . Pendant l'intervalle $[t, t + d_j]$, le jeton de la place P_i est le *jeton réservé* dans la place donc il n'est pas disponible pour aucune autre transition franchissable.

Dans notre approche, on considère les opérateurs : $T(u,v) = \min(u,v)$ et $\perp(u,v) = \max(u,v)$, ainsi que l'opérateur modus ponens généralisé $T_{\text{probabiliste}}(u, v) = u \cdot v$.

Pour les structures présentées dans la Figure 2, l'évolution du marquage est naturelle mais, les valeurs floues associées aux jetons des places suivantes, se déterminent en fonction des définitions suivantes :

Figure 2, a) $\alpha_k = \min(\alpha_1, \dots, \alpha_i) \cdot \mu_j$ (1)

Figure 2, b) $\alpha_1 = \alpha_2 = \alpha_k = \min(\alpha_1, \dots, \alpha_i) \cdot \mu_j$ (2)

Figure 2, c) $\alpha_k = \max(\alpha_1 \mu_1, \alpha_2 \mu_2, \dots, \alpha_j \mu_j)$ (3)

2.3 RdPSinF pour la modélisation des ressources des systèmes de productions. Fonctionnement.

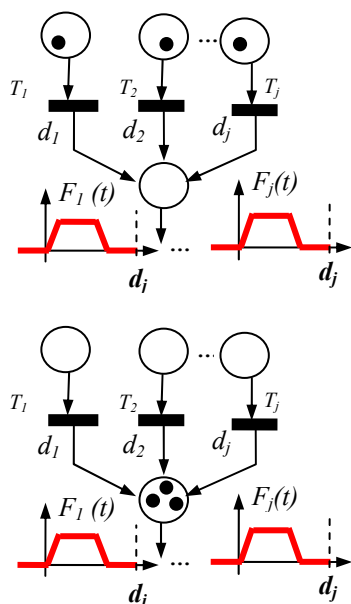


Figure 3. Modélisation des ressources des systèmes de productions avec RdPSinF

Dans ce cas, les durées de temporisations sont différentes pour chaque transition et sont mises en correspondance avec les durées des opérations de fabrication sur les machines-outils. Pour le cas Figure 2.c, on propose une évolution du marquage très différente si on modélise les ressources d'un système de production. Chaque jeton, après le tir de la transition, aura associé la valeur floue $\alpha_k = \alpha_i \cdot \mu_j$.

Chaque jeton indépendamment, aura le statut de jeton réservé pendant l'intervalle $[0, d_i]$ associé à la transition (Figure 3). D'autre part, pour les cas de la Figure 2.a,b, on conserve la même manière de calculer pour les valeurs floues associées aux jetons, après le tir de la transition correspondante.

3 ANALYSE DES RDPSinF

Pour exploiter de façon optimale l'outil proposé, nous présentons les principes de base de l'analyse de l'évolution d'un tel réseau. Pour cela, nous allons focaliser notre étude sur les transitions et plus précisément sur les transitions avec des places concurrentes.

Proposition 1 : Si une transition n'a pas associé un événement externe E , on considère qu'elle a associé l'événement $e =$ événement avec apparition permanente. La fonction F associée, sera la fonction constante $F(t)=1$

Proposition 2 : Si une transition n'a pas associé une durée de temporisation d , on considère qu'elle a la durée nulle donc le marquage après le franchissement sera un marquage instable.

Proposition 3 : Si on modélise la règle logique:

$$d_1 \wedge d_2 \wedge \dots \wedge d_i \rightarrow d_k \text{ ou } (d_1 \rightarrow d_k) \vee (d_2 \rightarrow d_k) \vee \dots \vee (d_i \rightarrow d_k),$$

la variable logique d_k aura la valeur floue associée : $\alpha_k = \min(\alpha_1, \alpha_2, \dots, \alpha_i)$, $\mu_j = \max(\alpha_1 \mu_1, \alpha_2 \mu_2, \dots, \alpha_i \mu_i)$

4. APPLICATIONS

4.1. Utilisation du RdPSinF dans la modélisation des règles logiques. Modélisation de l'arbre de défaillance ADD en utilisant le RdPSinF

i. la base des règles R est une réunion de règles logiques déduites de l'expression logique de l'arbre de défaillance ADD ;

ii. chaque place P modélise un défaut possible e_i elle est mise en correspondance avec une proposition logique d_i : "le défaut i est en train de se produire" ;

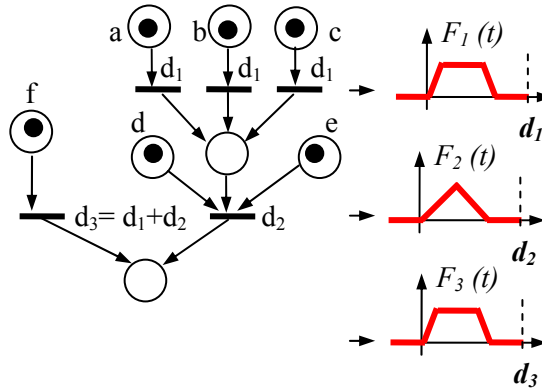


Figure 4. Modélisation d'une règle logique

iii. chaque transition t_i modélise une règle logique R_i , qui exprime l'apparition d'un défaut par combinaison des propositions logiques ;

iv. si une transition modélise l'évolution d'un défaut, connaissant la durée d_i de l'activité susceptible de générer ce défaut, on représente en flou le coefficient de vérité de cet événement ;

ce qui concerne l'évaluation des défauts, sans proposer une manière de correction. Ceci constitue l'objet de nos futures recherches.

RÉFÉRENCES

1. BOUCHON-MEUNIER, B., *La logique floue, Collection Que sais-je*, n.2702, Ed. Presses Universitaires de France, Paris, 1994.
2. CARDOSO, J., VALETTE, R., DUBOIS, D., *Petri nets with uncertain markings*, LNCS Advances in Petri nets, Vol. 483, G. Rozenberg, Ed. Springer Verlag, 1991, pp. 64 - 78
3. CARDOSO, J., PRADIN-CHEZALVIEL, B., *Logic and Fuzzy Petri Nets*, IEEE/SMC International Conference on Systems, Man and Cybernetics : Systems Engineering in the Service of Humans, Le Touquet, France, 17-20 Octobre 1993.
4. CHEN, S.-M., KE, J.-S., CHANG, J.-F., *Knowledge Representation Using Fuzzy Petri Nets*, IEEE Transaction on knowledge and data engineering, Vol. 2, No.3, Septembre 1990.
5. COMBACAU, M., *Commande et Surveillance des Systèmes à Evénements Discrets Complexes : Application aux Ateliers Flexibles*, Thèse de doctorat, Université Paul Sabatier, Toulouse, décembre 1991.
6. DUBOIS, D. AND H. PRADE, *Processing fuzzy temporal knowledge*, IEEE Trans. On Syst. Man and Cyb., 14, n°4, 1989.
7. LOONEY, C.G., *Fuzzy Petri nets for rule-based Decision making*, IEEE Trans. on Systems Man and Cybernetics, Jan-Feb 1988, vol. 18, n°1, pp 178-183.
8. GARG, M.L., AHSON, S.I., GUPTA, P.V., *A Fuzzy Petri net for knowledge representation and reasoning*, Information Processing Letters, vol. 39, 1991, p.165-171.
9. GOMES, L., STEIGER-GARCAO, A., *Programmable Controller design based on a synchronized Colored Petri net model and integrating fuzzy reasoning*, Lecture Notes in Computer Science 935, Springer Verlag, 1995, p. 218-237.
10. GROSDENIER, N., LOONIS, P., *Modélisation du suivi de situations par Réseau de Petri*, Le 12es Rencontres Francophones sur la Logique Floue et ses Applications, Nantes, 2004, France
11. LOONEY, C.G., *Fuzzy Petri nets for rule-based Decision making*, IEEE Trans. on Systems Man and Cybernetics, 1988 Jan-Feb, vol. 18, N 1, pp 178-183.
12. MINCA, E., D. RACOCEANU, N. ZERHOUNI AND I. BREZEANU, *Fuzzy Based Petri Nets for the production Systems Diagnosis*, Proceedings International Conference on Information Systems, Analysis and Synthesis, Electronics, Informatics and Control Systems – Invited Session, Orlando, 2002, Etats-Unis, Vol. XII.
13. MINCA, E., D. RACOCEANU, N. ZERHOUNI, *Monitoring Systems Modeling and Analysis Using Fuzzy Petri Nets*, Studies in Informatics and Control Journal, Institut National de Recherche et de Développement en Informatique, Vol 11, Nr°4, Bucarest, 2002, Roumanie.
14. MINCA, E., RACOCEANU, D., ZERHOUNI, N., BREZEANU, I., *Approche floue par la détection*, Qualita2003, 5ème Congrès International Pluridisciplinaire Qualité et Sécurité de fonctionnement, Nancy, 2003.
15. MINCA, E., D. RACOCEANU, N. ZERHOUNI, *Approche de détection par réseau de Petri flou*, Pentom2003, Performance et Nouvelles Technologies en Maintenance, 26-28 mars, Valenciennes, 2003.
16. MINCA, E., D. RACOCEANU, N. ZERHOUNI, *Fuzzy logic Petri nets for monitoring systems modeling*, IEEE International Conference on Robotics and Automation, Taiwan, 2003.
17. PRADIN CHÉZALVIEL, B., VALETTE, R., *Petri nets and Linear logic for process oriented diagnosis*, IEEE/SMC Int. Conf. on Systems, Man and Cybernetics: Systems Engineering in the Service of Humans, Le Touquet, France, Oct 17-20 1993.
18. PROOS, S., *Analyse des défaillances des systèmes industriels*, Rapport de stage DEA Informatique, Automatique et Productique, Laboratoire d'Automatique de Besançon, sept. 2001
19. RACOCEANU, D., ZERHOUNI, ADDOUCHE, N., *Modular modeling and analysis of a distributed production system with distant specialised maintenance*, Proc. of the 2002 IEEE International Conference on Robotics and Automation, sur CD ROM, 7 pages, pp. 4046-4052, 11-15 mai 2002, Washington, Etats-Unis.
20. VALETTE, R., CARDOSO, J., DUBOIS, D., *Monitoring manufacturing System by means of Petri Nets with Imprecise Markings*, IEEE International Symposium on Intelligent Control 1989, 25-26 sept. Albany N.Y., USA.
21. VALETTE, R., KUNZLE, L. A., *Réseaux de Petri pour la détection et le diagnostique*, Journées Nationales : Sécurité, Surveillance, Supervision, G.R. Automatique, Paris, 17-18 Novembre 1994.

Received May 15, 2008