



HAL
open science

Composer les réseaux de Petri temporels

Florent Peres, Bernard Berthomieu, François Vernadat

► **To cite this version:**

Florent Peres, Bernard Berthomieu, François Vernadat. Composer les réseaux de Petri temporels. Journal Européen des Systèmes Automatisés (JESA), 2009, 43 (7-9), pp.1001-1015. hal-00438377

HAL Id: hal-00438377

<https://hal.science/hal-00438377>

Submitted on 3 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Composer des réseaux de Petri temporels

Florent Peres^{1,2} — Bernard Berthomieu^{1,2} — François Vernadat^{1,2}

¹ CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France

² Université de Toulouse ; UPS, INSA, INP, INSAE ; LAAS ; F-31077 Toulouse, France
fperes@laas.fr, bernard@laas.fr, francois@laas.fr

RÉSUMÉ. La conception de systèmes complexes conduit souvent à les découper en éléments plus simples, appelés composants. Ceux-ci sont des sous-systèmes non clos qui, combinés (ou composés) entre eux, forment un système clos. Il est bien connu que les réseaux de Petri permettent une telle modélisation à base de composants. Nous nous intéressons ici à la composition parallèle, impliquant la fusion de transitions. L'ajout de contraintes temporelles sur les transitions (cas des réseaux de Petri temporels) ne pose pas de problèmes dans le cadre de la modélisation de systèmes clos. Par contre, la composition parallèle ne prenant pas trivialement en compte les informations temporelles des transitions, la compositionnalité n'est plus assurée. La solution présentée ici est d'étendre les TPN afin de déporter l'information temporelle des transitions composées, de manière à utiliser la composition parallèle traditionnelle.

ABSTRACT. The design of complex systems often leads to break them into smaller pieces, called components. Components are not closed sub-systems but are meant to be combined (or composed) to form a closed system. It is well known that Petri nets allows such a component based modeling. We are interested here in parallel composition, which is about transition fusion. Adding temporal constraints on transitions (time Petri nets case) does not generally bring much problems regarding the modeling of closed systems. In the other hand, parallel composition does not trivially take the temporal informations of transitions into account and, consequently, compositionality is no more ensured. The solution presented here consists of extending TPN to keep temporal information out of the composed transitions, in order to use the traditional parallel composition.

MOTS-CLÉS : Composition, réseaux de Petri temporels, compositionnalité

KEYWORDS: Composition, time Petri nets, compositionality

1. Introduction

Une technique classique pour gérer la complexité des systèmes temps réels est de les exprimer comme des assemblages de *composants*. Ainsi architecturés, des procédures incrémentales de développement, implémentation, test ou vérification de ces systèmes peuvent être mises en œuvre. Le sujet est discuté dans bon nombre de travaux, notamment (Gössler *et al.*, 2003; Bucci *et al.*, 1995; Sifakis *et al.*, 1996).

La *compositionnalité* est une notion essentielle pour la mise en œuvre d'une démarche basée sur des composants. Elle signifie que la sémantique d'un assemblage de composants est obtenue depuis celles des composants individuels et des opérateurs d'assemblage. En termes de vérification, cette propriété garantit notamment que les propriétés de sûreté (*safety*) des composants sont préservées par leurs compositions.

Comme modèle formel de systèmes temps réels, nous utilisons les réseaux de Petri temporels (Merlin *et al.*, 1976), qui étendent les réseaux de Petri par des intervalles temporels associés aux transitions. Ces réseaux sont d'une expressivité suffisante pour les systèmes clos, mais leur assemblage n'est pas trivialement une opération compositionnelle. Nous avons montré toutefois dans (Berthomieu *et al.*, 2006) que le produit synchronisé de réseaux temporels, un outil d'assemblage essentiel, était compositionnel lorsque les transitions synchronisées ne portent pas d'intervalle temporel.

Cet article s'appuie sur ces précédents travaux et propose une extension des réseaux de Petri temporels facilitant la description de composants. Le problème technique traité est celui-ci : étant donné un réseau de Petri temporel décrivant un composant, comment transformer celui-ci en un réseau équivalent, et composable au sens où ses transitions synchronisables ne portent pas d'intervalles temporels ? La notion d'équivalence adéquate dans ce contexte est celle de bisimulation temporisée, qui est une congruence pour le produit de réseaux temporels (soumis aux conditions ci-dessus).

Bien qu'effective, cette transformation peut conduire à une augmentation importante du nombre de transitions du réseau, et surtout du nombre de ses états internes. Pour résoudre ce problème efficacement, cet article propose d'utiliser comme cible des réseaux temporels étendus par deux relations : une relation d'"inhibition" similaire à la relation de priorité introduite dans (Berthomieu *et al.*, 2006) (si $t' \dashv t$, alors t n'est tirable qu'aux instants où t' ne l'est pas) et une relation de "permission" (si $t' \bullet t$, alors t n'est tirable qu'aux instants où t' l'est également). L'introduction de ces relations permet de résoudre le problème précédent de façon beaucoup plus efficace puisque la transformation dans ces réseaux étendus préserve la bisimilarité temporisée forte (plutôt que faible).

La technique est illustrée par quelques exemples. Une application directe est une procédure de traduction des automates temporisés (sans invariants stricts) dans ces réseaux temporels étendus préservant la bisimilarité temporisée, améliorant donc notre proposition (Berthomieu *et al.*, 2006) initiale de traduction.

Cet article est structuré comme suit. Les concepts essentiels concernant les réseaux temporels, leur sémantique, leur équivalence et leur produit sont rappelés en

section 2. La section 3 présente le problème que nous nous attachons à résoudre : la transformation de réseaux temporels en réseaux temporels “composables”, ainsi qu’une solution de base, adéquate mais coûteuse, faisant appel aux techniques de (Berthomieu *et al.*, 2006). La section 4 introduit les réseaux temporels à contraintes d’inhibition/permission (*ipTPN*). Ces réseaux sont ensuite appliqués dans la section 5 à notre problème de transformation en réseaux composables. D’autres applications de ce modèle sont discutées en section 6, notamment un codage efficace des automates temporisés sans invariants en *ipTPN*.

2. Réseaux temporels, sémantique et équivalences, produit de réseaux

2.1. Systèmes de transitions temporisés

Les systèmes de transition temporisés constituent la base sémantique des réseaux de Petri temporels et de leurs extensions.

Définition 1 (Systèmes de transition temporisé, ou STT) *Un système de transitions temporisé est une structure $\langle Q, q^0, \Sigma \cup \{\epsilon\}, \rightarrow \rangle$ où :*

- Q est un ensemble d’états ;
- $q^0 \in Q$ est l’état initial ;
- Σ est un ensemble fini d’actions qui ne contiennent pas l’action silencieuse ϵ ;
- $\rightarrow \subseteq Q \times (\Sigma \cup \{\epsilon\} \cup \mathbb{R}^+) \times Q$ est la relation de transition.

$(q, a, q') \in \rightarrow$ est noté $q \xrightarrow{a} q'$. Σ^ϵ désigne $\Sigma \cup \{\epsilon\}$. Les transitions dans $Q \times \Sigma^\epsilon \times Q$ sont appelées transitions *discrètes*, celles dans $Q \times \mathbb{R}^+ \times Q$ sont des transitions *continues*. Les transitions continues obéissent aux conditions suivantes ($\forall d, d' \in \mathbb{R}^+$) :

- délais nul** : $q \xrightarrow{0} q' \Leftrightarrow q = q'$;
- additivité** : $q \xrightarrow{d} q' \wedge q' \xrightarrow{d'} q'' \Rightarrow q \xrightarrow{d+d'} q''$;
- continuité** : $q \xrightarrow{d+d'} q' \Rightarrow (\exists q'')(q \xrightarrow{d} q'' \wedge q'' \xrightarrow{d'} q')$;
- déterminisme temporel** : $q \xrightarrow{d} q' \wedge q \xrightarrow{d} q'' \Rightarrow q' = q''$

Définition 2 (Produit de STT) *Soit deux STT $S_1 = \langle Q_1, q_1^0, \Sigma_1^\epsilon, \rightarrow_1 \rangle$ et $S_2 = \langle Q_2, q_2^0, \Sigma_2^\epsilon, \rightarrow_2 \rangle$. Il est supposé que toute action de Σ_i étiquette au moins une transition de S_i . Le produit de S_1 par S_2 est le STT $S_1 \parallel S_2 = \langle Q_1 \times Q_2, q_1^0 \parallel q_2^0, \Sigma_1^\epsilon \cup \Sigma_2^\epsilon, \rightarrow \rangle$ où \rightarrow est la plus petite relation satisfaisant :*

$$\frac{q_1 \xrightarrow{a}_1 q'_1}{q_1 \parallel q_2 \xrightarrow{a} q'_1 \parallel q_2} \quad (a \in \Sigma_1^\epsilon \setminus \Sigma_2) \quad \frac{q_2 \xrightarrow{a}_2 q'_2}{q_1 \parallel q_2 \xrightarrow{a} q_1 \parallel q'_2} \quad (a \in \Sigma_2^\epsilon \setminus \Sigma_1)$$

$$\frac{q_1 \xrightarrow{a}_1 q'_1 \quad q_2 \xrightarrow{a}_2 q'_2}{q_1 \parallel q_2 \xrightarrow{a} q'_1 \parallel q'_2} \quad (a \in \Sigma_1 \cup \Sigma_2 \cup \mathbb{R}^+)$$

Définition 3 Bisimilarité temporisée : Soient $S1 = \langle Q_1, q_1^0, \Sigma_1^\epsilon, \rightarrow_1 \rangle$ et $S2 = \langle Q_2, q_2^0, \Sigma_2^\epsilon, \rightarrow_2 \rangle$ deux *STT* et $\sim \subseteq Q_1 \times Q_2$. Alors $S1$ et $S2$ sont fortement temporellement bisimilaires ssi $q_1^0 \sim q_2^0$ et, quand $q_1 \sim q_2$ et $a \in \Sigma_1 \cup \Sigma_2 \cup \{\epsilon\} \cup \mathbb{R}^+$:

- (1) $q_1 \xrightarrow{a}_1 q'_1 \Rightarrow (\exists q'_2)(q_2 \xrightarrow{a}_2 q'_2 \wedge q'_1 \sim q'_2)$
- (2) $q_2 \xrightarrow{a}_2 q'_2 \Rightarrow (\exists q'_1)(q_1 \xrightarrow{a}_1 q'_1 \wedge q'_1 \sim q'_2)$

En pratique, la bisimilarité temporisée forte est souvent trop contraignante. Une relation d'équivalence plus grossière, cachant les transitions silencieuses, est obtenue à partir de la relation \xrightarrow{a} , définie depuis \xrightarrow{a} comme suit ($a \in \Sigma \cup \mathbb{R}^+, d \in \mathbb{R}^+$) :

$$\frac{q \xrightarrow{a} q'}{q \xrightarrow{a} q'} \quad \frac{q \xrightarrow{a} q' \quad q' \xrightarrow{\epsilon} q''}{q \xrightarrow{a} q''} \quad \frac{q \xrightarrow{\epsilon} q' \quad q' \xrightarrow{a} q''}{q \xrightarrow{a} q''} \quad \frac{q \xrightarrow{d} q' \quad q' \xrightarrow{d'} q''}{q \xrightarrow{d+d'} q''}$$

Deux systèmes de transitions temporisés seront dits *faiblement temporellement bisimilaires* lorsque les conditions (1) et (2) ci-dessus sont satisfaites, avec les relations \xrightarrow{a}_i remplacées par \xrightarrow{a}_i ($i \in \{1, 2\}$).

La bisimilarité temporisée, forte (\approx) ou faible (\approx_w), est préservée par le produit \parallel : Pour tout *STT* s, s_1, s_2 et $\sim \in \{\approx, \approx_w\}$, on a $s_1 \sim s_2 \Rightarrow s \parallel s_1 \sim s \parallel s_2$.

2.2. Réseaux de Petri temporels

Cette section rappelle la sémantique des réseaux de Petri temporels étiquetés et leur opération de produit. Les réseaux temporels étiquetés ajoutent aux réseaux temporels un alphabet d'action et une fonction d'étiquetage des transitions par ces actions. \mathbf{I}^+ est l'ensemble des intervalles réels non vides à bornes rationnelles. Pour tout $i \in \mathbf{I}^+$, $\downarrow i$ désigne sa borne inférieure, et $\uparrow i$ sa borne supérieure (si i est borné) ou ∞ . Pour tout $\theta \in \mathbb{R}^+$, $i \dot{-} \theta$ désigne l'intervalle $\{x - \theta \mid x \in i \wedge x \geq \theta\}$.

Définition 4 (Réseau de Petri temporel – TPN) Un Réseau de Petri temporel (ou *TPN*) est un tuple $\langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, Is, \Sigma^\epsilon, L \rangle$ dans lequel :

- $\langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0 \rangle$ est un réseau de Petri, avec pour places P , transitions T , marquage initial $m^0 : P \rightarrow \mathbb{N}^+$ et $\mathbf{Pre}, \mathbf{Post} : T \rightarrow P \rightarrow \mathbb{N}^+$,
- $Is : T \rightarrow \mathbf{I}^+$ est une fonction appelée Intervalle Statique,
- Σ est un ens. fini d'Actions, ou Labels, ne contenant pas l'action silencieuse ϵ ,
- $L : T \rightarrow \Sigma^\epsilon$ est une fonction d'Étiquetage des transitions.

Pour $f, g : P \rightarrow \mathbb{N}^+$, $f \geq g$ signifie $(\forall p \in P)(f(p) \geq g(p))$ et $f \{+|- \} g$ associe $f(p) \{+|- \} g(p)$ avec chaque p . Un marquage est une fonction $m : P \rightarrow \mathbb{N}^+$. $t \in T$

est sensibilisée pour m ssi $m \geq \mathbf{Pre}(t)$. L'ensemble des transitions sensibilisées par m dans le réseau N est noté $\mathcal{E}_N(m)$.

Définition 5 (Sémantique des TPN) La sémantique d'un TPN $\langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, Is, \Sigma, L \rangle$ est le système de transition temporisé $\langle S, (m^0, Is^0), \Sigma, \rightarrow \rangle$ où :

- Is^0 est la fonction Is restreinte aux transitions sensibilisées par m_0
- Les états de S sont des paires (m, I) dans lesquelles m est un marquage et $I : T \rightarrow \mathbf{I}^+$ associe un intervalle temporel à chaque transition sensibilisée par m ,
- $(m, I) \xrightarrow{L(t)} (m', I')$ ssi $t \in T$ et
 - 1) $m \geq \mathbf{Pre}(t)$
 - 2) $0 \in I(t)$
 - 3) $m' = m - \mathbf{Pre}(t) + \mathbf{Post}(t)$
 - 4) $(\forall k \in T)(m' \geq \mathbf{Pre}(k) \Rightarrow I'(k) = \mathbf{if } k \neq t \wedge m - \mathbf{Pre}(t) \geq \mathbf{Pre}(k) \mathbf{ then } I(k) \mathbf{ else } Is(k))$
- $(m, I) \xrightarrow{d} (m, I')$ ssi $(\forall k \in T)(m \geq \mathbf{Pre}(k) \Rightarrow d \leq \uparrow I(k) \wedge I'(k) = I(k) \dot{-} d)$

Une transition t peut être tirée depuis (m, I) si t est sensibilisée par m (condition 1) et tirable instantanément (2). Dans l'état cible, les transitions restées sensibilisées pendant le tir de t (t exclue) retiennent leurs intervalles, les autres sont associées à leurs intervalles statiques respectifs (4). Une transition continue par d est possible ssi d n'est pas plus grand que l'un des $\uparrow I(t)$.

Caractère Borné : Un TPN est borné si le marquage de toute place admet une borne supérieure. Le caractère borné implique la finitude de l'ensemble des marquages accessible (depuis m^0) du réseau. Le caractère borné est indécidable pour les TPN, mais il existe un nombre de conditions suffisantes pour cette propriété (Berthomieu *et al.*, 1983). Tous les réseaux considérés dans cet article seront supposés bornés.

Une notion de produit pour les réseaux de Petri étiquetés est utilisée dans de nombreux travaux, une définition apparaît notamment dans (Hack, 1976). Dans la définition ci-dessous, ce produit est étendu aux TPN, en supposant que les transitions synchronisées portent toutes l'intervalle statique trivial $[0, \infty[$.

Définition 6 (Produit de deux réseaux de Petri temporels) Considérons deux réseaux temporels étiquetés $N_1 = \langle P_1, T_1, \mathbf{Pre}_1, \mathbf{Post}_1, m_1^0, Is_1, \Sigma_1^\xi, L_1 \rangle$ et $N_2 = \langle P_2, T_2, \mathbf{Pre}_2, \mathbf{Post}_2, m_2^0, Is_2, \Sigma_2^\xi, L_2 \rangle$ ne partageant aucune place ni transition et tels que leurs transitions non silencieuses portent toutes l'intervalle $[0, \infty[$. Leur produit synchronisé est le réseau temporel $N = N_1 || N_2$ obtenu comme suit :

- N est initialement constitué de la réunion des réseaux N_1 et N_2 dans lesquels les transitions étiquetées dans $\Sigma_1 \cap \Sigma_2$ ont été supprimées,

– Ensuite, pour chaque paire $(t_1, t_2) \in T_1 \times T_2$ telle que $L_1(t_1) = L_2(t_2) \neq \epsilon$, on ajoute une transition $t_{1 \times 2}$, étiquetée $L_1(t_1)$, d'intervalle $[0, \infty[$, définie depuis t_1 et t_2 par :

$$\begin{aligned} (\forall p \in P)(\mathbf{Pre}(t_{1 \times 2})(p) &= \mathbf{Pre}(t_1)(p) + \mathbf{Pre}(t_2)(p)) \\ (\forall p \in P)(\mathbf{Post}(t_{1 \times 2})(p) &= \mathbf{Post}(t_1)(p) + \mathbf{Post}(t_2)(p)) \end{aligned}$$

Le produit \parallel est commutatif et associatif. Pour les réseaux de Petri et les réseaux temporels soumis aux conditions exprimées dans la définition 6, il est de plus compositionnel. Le théorème suivant est démontré dans (Berthomieu *et al.*, 2006) :

Théorème 1 (Compositionnalité restreinte du produit \parallel de TPN)

Soit $\lceil n \rceil$ le STT associé au TPN n et soient

$$\begin{aligned} N_1 &= \langle P_1, T_1, \mathbf{Pre}_1, \mathbf{Post}_1, m_1^0, Is_1, \Sigma_1^\epsilon, L_1 \rangle \\ N_2 &= \langle P_2, T_2, \mathbf{Pre}_2, \mathbf{Post}_2, m_2^0, Is_2, \Sigma_2^\epsilon, L_2 \rangle \end{aligned}$$

deux TPN étiquetés tels que :

$$(\forall t_1 \in T_1)(\forall t_2 \in T_2)(L_1(t_1) = L_2(t_2) \Rightarrow Is_1(t_1) = Is_2(t_2) = [0, \infty[)$$

Alors $\lceil N_1 \parallel N_2 \rceil = \lceil N_1 \rceil \parallel \lceil N_2 \rceil$

Définition 7 (TPN composable) Nous dirons qu'un TPN étiqueté $\langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, Is, \Sigma^\epsilon, L \rangle$ est composable lorsque toutes ses transitions étiquetées portent l'intervalle trivial $[0, \infty[$: $(\forall t \in T)(L(t) \neq \epsilon \Rightarrow Is(t) = [0, \infty[)$

Le produit de TPN composables est toujours défini. Deux réseaux temporels sont temporellement (fortement ou faiblement) bisimilaires si leurs sémantiques en termes de STT le sont. Comme pour les STT, la bisimilarité temporisée de TPN composables est préservée par le produit \parallel : Pour tout TPN composables n, n_1, n_2 et $\sim \in \{\approx, \approx_w\}$, on a $n_1 \sim n_2 \Rightarrow n \parallel n_1 \sim n \parallel n_2$.

3. Des TPN aux composants

3.1. Assemblages

Nous considérons les systèmes construits de la façon suivante :

- Les composants élémentaires sont des TPN étiquetés composables (Déf. 7) ;
- Les opérateurs d'assemblages sont :
 - Le produit de TPN (composables) ;
 - Le renommage de transitions non silencieuses ;
 - Le masquage de transitions (remplacement de leurs étiquettes par ϵ) ;

- Le renforcement d'une contrainte temporelle d'une transition silencieuse.

La bisimilarité temporisée, faible ou forte, est clairement une congruence pour ces opérateurs. Comme d'autre part elle capture toute les propriétés de sûreté et de vivacité raisonnablement utiles pour nos systèmes (Browne *et al.*, 1988), nous la considérerons comme une notion d'équivalence adéquate pour nos composants.

3.2. Modéliser les composants

Les réseaux temporels possèdent les capacités nécessaires pour modéliser les composants d'un système temps réel, individuellement. Mais il ne produisent pas nativement des modèles composables au sens de la définition 7.

Il nous faut donc une méthode systématique pour obtenir depuis un modèle de composant à base de *TPN* un modèle équivalent (c'est-à-dire faiblement ou fortement temporellement bisimilaire) mais dont les transitions étiquetées portent toutes l'intervalle trivial $[0, \infty[$. Nous nous intéresserons dans cette section à l'étude des possibilités de telles transformations offertes par les *TPN*. Plus précisément, le problème que nous souhaitons résoudre est le suivant :

Étant donné un réseau de petri temporel étiqueté, comment transformer celui-ci en un réseau étiqueté faiblement temporellement bisimilaire dont toute transition étiquetée porte l'intervalle $[0, \infty[$?

Bien que complexe, cette transformation est possible pour une large classe de réseaux : ceux dont une borne est connue pour le marquage de chaque place et dont les intervalles temporels associés aux transitions sont soit non bornés, soit fermés à droite (aucun intervalle n'a la forme $[\alpha, \beta[$ ou $] \alpha, \beta]$). La condition sur le caractère borné peut être levée si l'on admet l'usage d'arcs inhibiteurs généralisés (permettant de tester que le marquage d'une place est inférieur à une constante donnée), mais nous ne présenterons ici que la transformation plus simple applicable aux réseaux bornés.

Soit \mathcal{N} le réseau de départ, borné et restreint comme indiqué ci-dessus. Nous allons traduire ce réseau en un réseau équivalent dont aucune transition non silencieuse n'est temporisée, obtenu par le produit d'un réseau \mathcal{N}' et d'un ensemble de réseaux temporels $C(t)$. Les étapes de la transformation sont les suivantes :

1) A chaque transition temporisée non silencieuse t de \mathcal{N} est associé un *TPN* $C(t)$ codant l'évolution de l'horloge de t , basée sur le codage introduit dans (Berthomieu *et al.*, 2006). Le contexte nous permet toutefois quelques simplifications : les bornes supérieures des intervalles temporels étant ici considérées comme des conditions d'urgence, seuls les motifs de traduction concernant les conditions d'urgence et les gardes de la forme $\gamma \geq \alpha$ ou $\gamma > \alpha$ sont nécessaires ; noter que ceux-ci ne nécessitent pas les priorités introduites dans (Berthomieu *et al.*, 2006).

A titre d'illustration, les réseaux $C(t)$ codant l'évolution de l'horloge de t lorsque celle-ci porte l'intervalle $[3, 5]$ ou $[3, \infty[$ sont montrés dans la Figure 1. Les transitions étiquetées " $r(t)$ " restaurent l'état initial du réseau ; elles sont tirables à tout instant.

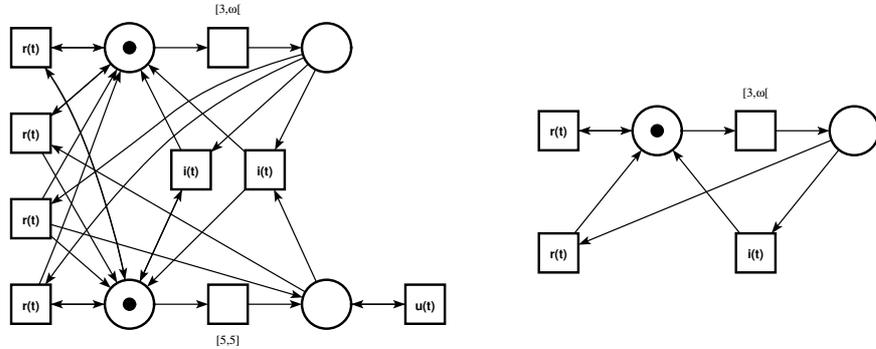


Figure 1. Réseau $C(t)$ lorsque t porte l'intervalle $[3, 5]$ (resp. $[3, \infty[$)

Celles étiquetées “ $i(t)$ ” (noter qu’elle ne portent pas elles-mêmes d’intervalles temporels) font de même mais ne sont tirables que dans l’intervalle temporel $[3, 5]$ (resp. $[3, \infty[$) relativement à la date d’établissement du marquage initial ou à la date du tir de la dernière transition étiquetée “ $r(t)$ ”. Celle étiquetée “ $u(t)$ ” n’est tirable qu’à partir de la date 5 (comprise).

2) Le réseau \mathcal{N}' est obtenu depuis \mathcal{N} comme suit (initialement, $\mathcal{N}' = \mathcal{N}$) :

L’ensemble des transitions nouvellement sensibilisées après le tir de t depuis le marquage m est l’ensemble des k satisfaisant : $m - \mathbf{Pre}(t) \not\geq \mathbf{Pre}(k)$ et $m - \mathbf{Pre}(t) + \mathbf{Post}(t) \geq \mathbf{Pre}(k)$. Le réseau \mathcal{N} étant borné (par hypothèse), il n’existe qu’un nombre fini de marquages de \mathcal{N} depuis lesquels t est tirable ; soit E_t cet ensemble, calculé depuis le graphe des marquages de \mathcal{N} ou une sur-approximation.

Pour chaque transition temporisée non silencieuse t de \mathcal{N} :

a) Rajouter à \mathcal{N}' une transition t_u étiquetée “ $u(t)$ ” et telle que $\mathbf{Pre}(t_u) = \mathbf{Post}(t_u) = \mathbf{Pre}(t)$;

b) Répliquer t dans \mathcal{N}' en autant de transitions t_i que E_t admet d’éléments, et pour chaque t_i :

- Augmenter ses conditions de tir de façon à ce que t_i ne soit tirable que depuis le $i^{\text{ème}}$ marquage m_i de E_t , tout en préservant, pour toute place p : $\mathbf{Post}(t_i, p) - \mathbf{Pre}(t_i, p) = \mathbf{Post}(t, p) - \mathbf{Pre}(t, p)$. Ceci nécessite l’adjonction au réseau \mathcal{N}' des places complémentaires des places d’entrée de t_i ;
- Associer à t_i l’intervalle $[0, \infty[$ et un ensemble d’étiquettes : $i(t)$, et $r(t')$ pour chaque t' nouvellement sensibilisée dans \mathcal{N} après le tir de t depuis m_i .

3) Construire le produit du réseau \mathcal{N}' et de la réunion des réseaux $C(t)$ obtenus à l’étape (2). La synchronisation se fera sur les étiquettes “ $u(t)$ ”, “ $i(t)$ ” et “ $r(t)$ ”. Les transitions de \mathcal{N}' portant plusieurs étiquettes seront synchronisées avec toutes les transitions de l’autre réseau portant l’une de ces étiquettes (ceci constitue une extension mineure de la définition 6) ;

4) Dans le réseau produit, pour tout k , associer l'intervalle $[0, 0]$ aux transitions étiquetées $u(k)$, puis supprimer les étiquettes " $r(k)$ " et " $u(k)$ ", et remplacer les étiquettes " $i(k)$ " par $L(k)$.

L'espace imparti nous interdit de le démontrer ici, mais cette transformation produit un réseau équivalent au réseau \mathcal{N} par la bisimulation temporisée faible, et composable (toutes les transitions étiquetées portent l'intervalle $[0, \infty[$). La construction de base décrite peut être optimisée mais elle ajoute en général un nombre important de places et de transitions au réseau de départ. En pratique, et sauf dans certains cas particuliers (réseaux saufs par construction, par exemple), elle est difficilement applicable et accroît notablement le nombre de marquages du réseau.

Pour ces raisons, nous allons étudier la même problématique en utilisant une extension des TPN permettant à une transition, en plus de son intervalle temporel, d'utiliser les informations temporelles des autres transitions.

4. Réseaux de Petri temporels à relations d'inhibition et permission

Les réseaux de Petri temporels ont été étendus dans (Berthomieu *et al.*, 2006) par une relation de priorité améliorant leur expressivité. Nous étendons ce formalisme à l'aide d'une nouvelle relation de *permission*, la relation de priorités étant quant à elle renommée en relation d'*inhibition*.

Définition 8 Un réseau de Petri temporel à relations d'inhibition et permission étiqueté ($ipTPN$) est un tuple $\langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, Is, F, A, \Sigma, L \rangle$, où :

- $\langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, Is, \Sigma, L \rangle$ est un réseau de Petri temporel étiqueté.
- $F \subseteq T \times T$ est la relation d'inhibition ; $(x, y) \in F$ sera noté $x \circ y$.
- $A \subseteq T \times T$ est la relation de permission ; $(x, y) \in A$ sera noté $x \bullet y$.

La sémantique d'un $ipTPN$ est définie à partir de celle des TPN (définition 5) en ajoutant deux règles (2 bis) et (2 ter), concernant les deux nouvelles relations :

Définition 9 (Sémantique des $ipTPN$) La sémantique d'un $ipTPN$ $\langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, Is, F, A, \Sigma, L \rangle$ est le système de transition temporisé $\Gamma = \langle S, (m^0, Is^0), \Sigma, \rightarrow \rangle$ obtenu comme dans la définition 5 en rajoutant les conditions suivantes :

- (2 bis) $(\forall k \in T)(k \circ t \Rightarrow 0 \notin I(k))$
- (2 ter) $(\forall k \in T)(k \bullet t \Rightarrow 0 \in I(k))$

Une transition t sera dite T -sensibilisée si elle tirable dans le TPN sous-jacent. La règle (2 bis) est la même que celle des PrTPN de (Berthomieu *et al.*, 2006), mais alors que la relation de priorité est asymétrique, irréflexive et transitive, la relation

d'inhibition F est, elle, quelconque. Il est donc permis d'avoir $x \dashv x$, ce qui empêche de tirer x tout en gardant la possibilité de tester sa T -sensibilisation. Il est possible d'avoir à la fois $x \dashv y$ et $y \dashv x$ — ce qui n'implique ni $x \dashv x$, ni $y \dashv y$, car il n'est pas requis que F soit transitive — signifiant que x (resp. y) n'est tirable que lorsque y (resp. x) n'est pas T -sensibilisée.

Graphiquement, chaque élément de la relation d'inhibition (resp. de permission) est représenté par un arc orienté, avec une flèche circulaire creuse (resp. pleine). La transition "source" est celle dont la T -sensibilisation est testée. Ainsi, sur la figure 2, on a $t_2 \dashv t_2$, $t_0 \dashv t_1$ et $t_2 \bullet t_0$. Puisque t_2 s'inhibe elle-même, elle ne sera jamais tirable. Néanmoins, elle contraint t_0 en imposant que cette dernière ne puisse pas être tiré avant 1 unité de temps. Puisque t_2 doit également se soumettre aux règles d'écoulement du temps, le temps ne peut pas progresser au delà de 2, à moins de tirer t_0 ou t_1 , car dans ce cas, p_0 devient vide et t_2 est désensibilisée. Même si t_0 ne peut tirer qu'à partir de 1 unité de temps, elle contraint tout de même t_1 , obligeant celle-ci à attendre que t_0 soit tirée pour pouvoir l'être à son tour. Puisque t_0 prend le seul jeton de p_0 , t_1 ne pourra jamais tirer.

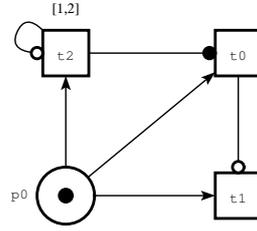


Figure 2. Un exemple d'ipTPN

Le produit de TPN sera étendu pour prendre en compte les deux nouvelles relations. Toute transition t' de $N_1 || N_2$ est soit une transition de $T_1 \cup T_2$, soit le produit d'une transition $t_i \in T_1$ par une transition $t_j \in T_2$. Nous dirons que t' contient $t \in T_1 \cup T_2$ (noté $t \in t'$ ou $t' \ni t$) si $t' = t$ ou t' résulte de la composition de t et d'une autre transition. Pour $R \in \{A, F\}$, le produit assurera les conditions suivantes :

$$\begin{aligned} (\forall (t'_i, t'_j) \in R) (\exists t_i \in t'_i, t_j \in t'_j) ((t_i, t_j) \in R_1 \cup R_2) \\ (\forall (t_i, t_j) \in R_1 \cup R_2) (\forall t'_i \ni t_i, t'_j \ni t_j) ((t'_i, t'_j) \in R) \end{aligned}$$

5. Capacité de composition des ipTPN

Nous reprenons ici l'étude de la section 3.2 dans le but de montrer que les ipTPN que nous venons de définir permettent d'étendre la capacité de composition des TPN.

Soit un TPN $R = \langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, Is, \Sigma, L \rangle$ et $W = \{t \in T | L(t) \neq \epsilon\}$ l'ensemble de ses transitions non silencieuses. Soit W' un ensemble distinct de $P \cup T$ et $|W| = |W'|$. On peut définir une bijection entre les éléments de W et de W' ; soit $B : W \rightarrow W'$ cette bijection.

Soit $R' = \langle P, T \cup W', \mathbf{Pre}', \mathbf{Post}', m^0, Is', \Sigma, L' \rangle$, l'ipTPN défini comme suit :

- a) $(\forall t \in T)(L'(t) = L(t))$
- b) $(\forall t \in W)(L'(B(t)) = \epsilon)$
- c) $(\forall t \in W)(\mathbf{Pre}'(B(t)) = \mathbf{Pre}(t))$

- d) $(\forall t \in T)(\mathbf{Pre}'(t) = \mathbf{Pre}(t) \wedge \mathbf{Post}'(t) = \mathbf{Post}(t))$
- e) $(\forall t \in W)(B(t) \rightarrow \bullet t)$
- f) $(\forall t \in W)(B(t) \rightarrow \circ B(t))$
- g) $(\forall t \in T \setminus W)(I'_s(t) = I_s(t))$
- h) $(\forall t \in W)(I'_s(t) = [0, \infty[\wedge I'_s(B(t)) = I_s(t))$

La figure 3 représente un réseau Ex non composable et sa transformation Ex' en réseau composable.

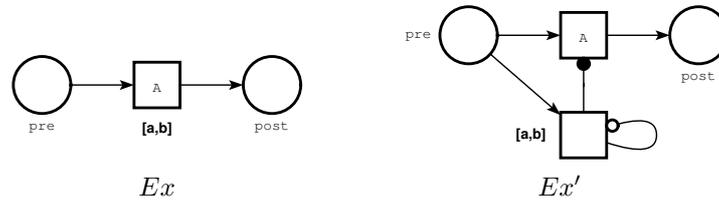


Figure 3. Exemple de transformation

La règle a) recopie les étiquettes de R dans R' . La règle b) étiquette les transitions nouvellement créées par ϵ . Ces dernières admettent les mêmes préconditions que la transition dont nous souhaitons la déportation de l'information temporelle (règle c). Les transitions du réseau R conservent toutes leurs préconditions et postconditions (règle d). Les transitions nouvellement créées contraignent leurs jumelles par la relation de permission, ce qui permet de spécifier la borne inférieure de tir puisque les préconditions sont les mêmes (règle e). Les transitions de W' sont empêchées de tirer par une boucle d'inhibition, cette boucle n'empêche pas la règle normale d'écoulement du temps pour cette transition, et par conséquent, elle contraint sa jumelle (et même tout le réseau) par sa borne supérieure (règle f). Les transitions de R , sauf celles à l'interface gardent leurs intervalles temporels (règle g). Les transitions de R qui sont à l'interface portent dans R' l'intervalle de tir le plus large : $[0, \infty[$ (règle h) ; les transitions de W' portent l'ancien intervalle de tir de leurs jumelles dans W .

On constatera sans qu'il y ait besoin de présenter une preuve, que le réseau R' accomplit exactement les mêmes actions et aux mêmes instants. De plus, les seules transitions rajoutées n'étant pas tirables, nous en concluons que le réseau R' est un réseau en relation de bisimulation temporisée forte avec R .

6. Une applications du pouvoir compositionnel des ipTPN

La comparaison des deux formalismes des TPN et des TA est un sujet de recherche qui a fait l'objet de beaucoup de résultats (par exemple, (Bérard *et al.*, 2005b), (Bérard *et al.*, 2005a), (Cassez *et al.*, 2006), (Bouyer *et al.*, 2006)). Dans nos précédents travaux (Berthomieu *et al.*, 2006), nous avons jeté un nouveau pont entre les deux formalismes : en ajoutant des priorités entre transitions (la relation

d'inhibition F est très proche de celle de priorité), nous avons pu obtenir une traduction garantissant l'existence d'une relation de bisimulation temporisée faible. Cette nouvelle extension nous donne les moyens d'aller plus loin, en nous permettant une traduction garantissant l'existence d'une bisimulation temporisée *forte*.

Définition 10 Un automate temporisé est un tuple $\langle Q, q^0, X, \mathcal{C}, \Sigma, Tr \rangle$, où :

- Q est un ensemble fini de locations
- q^0 est la location initiale
- X est un ensemble fini d'horloges
- \mathcal{C} est un ensemble fini de contraintes. Une contrainte est un triplet $\langle x, c, op \rangle$, avec $x \in X$, $c \in \mathbb{Q}^+$, $op \in \{<, \leq, \geq, >\}$.
- Σ l'alphabet d'action (ne contient pas l'action silencieuse ϵ)
- $Tr \subseteq Q \times 2^{\mathcal{C}(X)} \times \Sigma^\epsilon \times 2^X \times Q$ l'ensemble des transitions

Nous abrègerons $(q, g, a, r, q') \in Tr$ en $q \xrightarrow{g, a, r} q'$. L'ensemble de contraintes g est appelé la garde de la transition. L'ensemble r est l'ensemble des horloges remises à zéro lors du franchissement de la transition.

Soit $\Gamma = \langle Q, q^0, X, \mathcal{C}, \Sigma, Tr \rangle$, un automate temporisé et un réseau de Petri temporel à inhibition/permission $\Delta = \langle P, T, \mathbf{Pre}, \mathbf{Post}, m^0, I_s, \Sigma, L, F, A \rangle$. Soit $\mathcal{B}_{tr} : Tr \rightarrow T_1$ une application bijective avec $T_1 \subseteq T$. Soit $\mathcal{B}_C : \mathcal{C} \rightarrow T_2$ une application bijective avec $T_2 \subseteq T$ et $T_1 \cap T_2 = \emptyset$.

- 1) $P = Q \cup X \wedge T = \{\mathcal{B}_{tr}(t) | t \in Tr\} \uplus \{\mathcal{B}_C(c) | c \in \mathcal{C}\}$
- 2) $(\forall q \in Q \setminus \{q^0\})(m^0(q) = 0) \wedge m^0(q^0) = 1$
- 3) $(\forall t \in Tr)(t = q \xrightarrow{g, a, r} q' \Rightarrow \mathbf{Pre}(q, \mathcal{B}_{tr}(t)) = 1 \wedge \mathbf{Post}(q', \mathcal{B}_{tr}(t)) = 1 \wedge L(\mathcal{B}_{tr}(t)) = a \wedge I_s(\mathcal{B}_{tr}(t)) = [0, \infty])$
- 4) $(\forall x \in X)(m^0(x) = 1)$
- 5) $(\forall t \in Tr)(t = q \xrightarrow{g, a, r} q' \Rightarrow (\forall x \in r)(\mathbf{Pre}(x, \mathcal{B}_{tr}(t)) = 1 \wedge \mathbf{Post}(x, \mathcal{B}_{tr}(t)) = 1))$
- 6) $(\forall t \in Tr)(t = q \xrightarrow{g, a, r} q' \Rightarrow (\forall c \in g)(c = \langle x, d, < \rangle \Rightarrow \mathbf{Pre}(x, \mathcal{B}_C(c)) = 1$
 $\wedge (< \in \{<, \geq\} \Rightarrow I_s(\mathcal{B}_C(c)) = [d, \infty])$
 $\wedge (< \in \{\leq, >\} \Rightarrow I_s(\mathcal{B}_C(c)) =]d, \infty])$
 $\wedge \mathcal{B}_C(c) \multimap \mathcal{B}_C(c)$
 $\wedge (< \in \{<, \leq\} \Rightarrow \mathcal{B}_C(c) \multimap \mathcal{B}_{tr}(t))$
 $\wedge (< \in \{>, \geq\} \Rightarrow \mathcal{B}_C(c) \multimap \bullet \mathcal{B}_{tr}(t)))$

Les transitions de l'automate Γ sont traduites par les transitions du réseau Δ images de \mathcal{B}_{tr} tandis que les contraintes d'horloges de Γ sont traduites par les transitions de Δ images de $\mathcal{B}_C(c)$. Aucune des transitions de l'ensemble $\{\mathcal{B}_C(c) | c \in \mathcal{C}\}$ des transitions traduisant une contrainte n'est tirable car elles s'inhibent elles-mêmes ($\mathcal{B}_C(c) \multimap \mathcal{B}_C(c)$). Les seules transitions tirables sont celles de l'ensemble $\{\mathcal{B}_{tr}(t) | t \in$

T_r } des transitions traduisant celles de l'automate Γ . Il n'y en a pas d'autres car l'ensemble T des transitions de Δ est donné par l'union de ces deux ensembles.

Chaque location de Γ est directement traduite en une place de Δ . Comme la location initiale est q^0 , la place correspondante est initialement marquée (point 2). Le point 3) décrit la traduction du squelette de l'automate. La seule précondition (resp. postcondition) de la transition $\mathcal{B}_{tr}(t)$ est la place représentant la location d'origine (resp. destination) de la transition t de l'automate.

Chaque horloge est codée par une place (point 4)) initialement marquée et dont le jeton n'est jamais enlevé sans être remis aussitôt comme c'est le cas pour la remise à zéro (point 5). Le point 6) gère les gardes, chacune d'elles étant codée par une transition jamais tirable (puisque $\mathcal{B}_C(c) \multimap \mathcal{B}_C(c)$), d'information temporelle ($I_s(\mathcal{B}_C(c))$), liée à une transition tirable par la relation F ou A , selon la contrainte à coder.

Les transitions $\mathcal{B}_{tr}(t)$ ne portent pas d'information temporelle mais sont contraintes par inhibition/permission par les transitions $\mathcal{B}_C(c)$ codant les contraintes c . Examinons maintenant ce codage :

$c = x < d : I_s(\mathcal{B}_C(c)) = [d, \infty[$ et $\mathcal{B}_C(c) \multimap \mathcal{B}_{tr}(t)$ signifie que $\mathcal{B}_{tr}(t)$ est tirable tant que $\mathcal{B}_C(c)$ n'est pas T -sensibilisée. Autrement dit, $\mathcal{B}_{tr}(t)$ n'est tirable que d unités de temps après la dernière remise à zéro de $\mathcal{B}_C(c)$. Parce que x est précondition de $\mathcal{B}_C(c)$, $\mathcal{B}_C(c)$ est tirable dès que d unités de temps se sont écoulées.

$c = x \leq d : I_s(\mathcal{B}_C(c)) =]d, \infty[$ et $\mathcal{B}_C(c) \multimap \mathcal{B}_{tr}(t)$, signifie que $\mathcal{B}_{tr}(t)$ est tirable tant que $\mathcal{B}_C(c)$ n'est pas T -sensibilisée, c'est à dire tant que $\mathcal{B}_C(c)$ n'a pas dépassé d .

$c = x > d : I_s(\mathcal{B}_C(c)) = [d, \infty[$ et $\mathcal{B}_C(c) \bullet \mathcal{B}_{tr}(t)$, ce qui signifie que $\mathcal{B}_{tr}(t)$ n'est tirable que lorsque $\mathcal{B}_C(c)$ est T -sensibilisée, ce qui signifie que $\mathcal{B}_{tr}(t)$ n'est tirable qu'après avoir été continûment sensibilisée pendant plus de d unités temps.

$c = x \geq d : I_s(\mathcal{B}_C(c)) = [d, \infty[$ et $\mathcal{B}_C(c) \bullet \mathcal{B}_{tr}(t)$, signifie que $\mathcal{B}_{tr}(t)$ est tirable dès que $\mathcal{B}_C(c)$ est T -sensibilisée, c'est-à-dire dès que d unités de temps se sont écoulés depuis la dernière remise à zéro.

Nous venons de nous assurer que le codage des contraintes est correct. Nous avons par ailleurs vu que les seules transitions tirables sont exactement celles traduisant les transitions de l'automate temporisé et finalement que les horloges encodées sont remises à zéro au bon moment. Le langage temporisé de l'automate temporisé sera donc nécessairement le même et la bisimulation temporisée forte est assurée.

Dans la figure 4 sont représentés un automate temporisé et sa traduction en *ipTPN*. La location initiale de l'automate est 0. Les transitions sont étiquetées selon le triplet (g, a, r) , les gardes sont séparées par une virgule, les éléments du triplet par un point-virgule. L'ensemble vide est noté par $\{\}$. Cet automate fait donc dans l'ordre un A , puis un B , puis un C . Le premier A n'est produit que si $x \leq 2$, au-delà, l'automate est bloqué dans la location 0 où seul le temps peut diverger. Dans la location 1, un B est produit dès que $y > 3$ est vrai. L'horloge y est remise à zéro lors du franchissement de la transition. La transition faisant un C admet deux contraintes d'horloges $x \geq 3$ et $y < 2$ et remet à zéro l'horloge x lors de son franchissement.

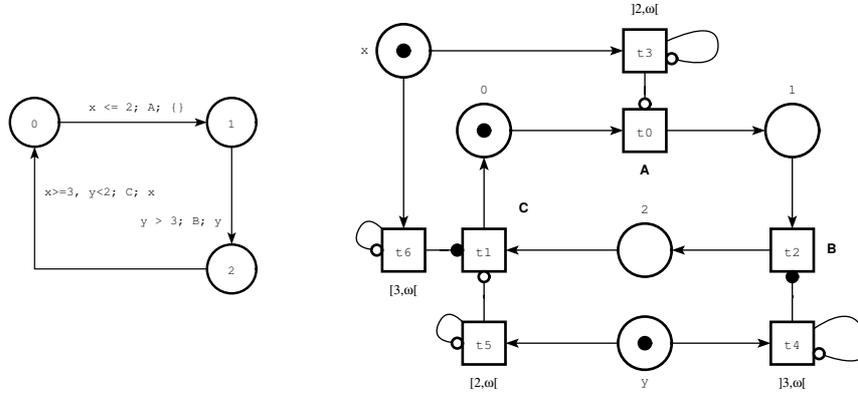


Figure 4. Un automate temporisé et sa traduction en *ipTPN*

Sur la figure 4, les places 0, 1 et 2 ainsi que les transitions t_0 , t_1 et t_2 et les arcs les reliant forment la traduction du squelette atemporel de l'automate. Les contraintes d'horloges sont ensuite greffées. Pour x , la transition t_3 empêche t_0 de tirer après 2 unités de temps et la transition t_6 empêche t_1 de tirer avant que l'horloge interne de t_6 n'atteigne 3 unités de temps. Pour y , la transition t_4 empêche de tirer t_2 avant 3 et à 3 unités de temps, alors que t_5 empêche de tirer t_1 au partir de 2 unités de temps. Les quatre transitions t_3 , t_4 , t_5 et t_6 possèdent une boucle inhibitrice qui interdit leur tir.

La traduction présentée dans cette section concerne les automates temporisés sans invariants. Les invariants non stricts peuvent être pris en compte et traités comme dans (Berthomieu *et al.*, 2006). Le traitement ne préserve plus la bisimilarité forte, toutefois, mais la traduction ainsi obtenue est globalement beaucoup plus compacte que celle de (Berthomieu *et al.*, 2006).

7. Conclusions

A l'heure où les systèmes temps réel sont de plus en plus complexes et où la demande de vérification est toujours plus importante, il est intéressant d'avoir avec soi un outil permettant d'en diminuer la complexité en permettant leur décomposition en éléments plus facilement manipulables.

S'il est clair qu'au niveau de la spécification, cela représente un apport essentiel, celui-ci n'est pas aussi évident en ce qui concerne la vérification : un composant n'est bien souvent utilisé qu'avec des environnements très spécifiques, or vérifier un composant revient à le vérifier pour tous les environnements compositionnellement acceptables. Par conséquent, la vérification par composants peut être encore plus coûteuse que la vérification du système lui-même. Il est donc plus que jamais nécessaire de continuer à explorer la voie de la vérification compositionnelle.

La vérification de ce formalisme n'apporte pas de réels défis en termes algorithmiques mais en promet de nouveaux car beaucoup plus de transitions portent une information partielle (telle que les ensembles $[x, \infty[$), ou simplement n'en portent pas mais sont tout de même comptées comme telles, d'autres ne seront jamais tirables mais contraignent néanmoins le système. Au final, la représentation des états est plus complexe et ouvre de nouvelles problématiques en terme d'optimisations.

Enfin, de manière plus générale, ces travaux vont permettre la spécification et l'intégration beaucoup plus aisée d'observateurs aux sein d'un système conçu (ou traduit) de manière à être composable. En effet, la possibilité de tester si les horloges des autres transitions ont atteint ou non telle ou telle borne, sans avoir besoin de traduire cela en un schéma de codage, lequel est requis avec l'utilisation des TPN à priorités, fournit une base solide à l'intégration de tels observateurs. Ceux-ci sont par exemple utiles pour les techniques de vérification à base d'observateurs mais également pour la spécification d'ordonnanceurs, car un ordonnanceur est aussi un observateur.

8. Bibliographie

- Bérard B., Cassez F., Haddad S., Roux O. H., Lime D., « Comparison of the Expressiveness of Timed Automata and Time Petri Nets », *Formal Modeling and Analysis of Timed Systems (FORMATS), LNCS 3829*, p. 211-225, 2005a.
- Bérard B., Cassez F., Haddad S., Roux O. H., Lime D., « When are Timed Automata weakly timed bisimilar to Time Petri Nets? », *25th Conference on Foundations of Software Technology and Theoretical Computer Science, Springer LNCS 3821*, p. 261-272, 2005b.
- Berthomieu B., Menasche M., « An Enumerative Approach for Analyzing Time Petri Nets. », *IFIP Congress Series*, vol. 9, p. 41-46, 1983.
- Berthomieu B., Peres F., Vernadat F., « Bridging the gap between Timed Automata and Bounded Time Petri Nets. », *4th Int. Conf. on Formal Modelling and Analysis of Timed Systems (FORMATS), Springer LNCS 4202*, 2006.
- Bouyer P., Haddad S., Reynier P.-A., « Extended Timed Automata and Time Petri Nets », *Proc. of 6th International Conference on Application of Concurrency to System Design (ACSD'06)*, IEEE Computer Society Press, Turku, Finland, June, 2006.
- Browne M. C., Clarke E. M., Grumberg O., « Characterizing finite Kripke structures in propositional temporal logics », *T.C.S.*, vol. 59, p. 115-131, 1988.
- Bucci G., Vicario E., « Compositional Validation of Time-Critical Systems Using Communicating Time Petri Nets. », *IEEE Tr. on Soft. Eng.*, vol. 21, n° 12, p. 969-992, 1995.
- Cassez F., Roux O. H., « Structural Translation from Time Petri Nets to Timed Automata », *Journal of Systems and Software*, vol. 29, n° 1, p. 1456-1468, 2006.
- Gössler G., Sifakis J., « Composition for Components-Based Modeling », *Formal Methods for Components and Objects, Springer LNCS 2852*, p. 443-466, 2003.
- Hack M., Petri net languages, Technical Report n° 159, MIT, Cambridge, Mass., 1976.
- Merlin P. M., Farber D. J., « Recoverability of Communication Protocols : Implications of a Theoretical Study. », *IEEE Tr. Comm.*, vol. 24, n° 9, p. 1036-1043, Sept., 1976.
- Sifakis J., Yovine S., « Compositional Specification of Timed Systems », *13th Annual Symp. on Theoretical Aspects of Computer Science (STACS), Springer LNCS 1046*, p. 347-359, 1996.