



HAL
open science

Identification de SED au moyen de réseaux de Petri : une approche pour la représentation structurelle des comportements observés

Ana-Paula Estrada-Vargas, Jean-Jacques Lesage, E. López-Mellado

► **To cite this version:**

Ana-Paula Estrada-Vargas, Jean-Jacques Lesage, E. López-Mellado. Identification de SED au moyen de réseaux de Petri : une approche pour la représentation structurelle des comportements observés. IEEE Conférence Internationale Francophone d'Automatique (CIFA'12), Jul 2012, Grenoble, France. pp. 919-924. hal-00716049

HAL Id: hal-00716049

<https://hal.science/hal-00716049>

Submitted on 9 Jul 2012

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.

Identification de SED au moyen de réseaux de Petri : une approche pour la représentation structurelle des comportements observés

Ana Paula Estrada-Vargas ^{***}, Jean-Jacques Lesage ^{*}
Ernesto López-Mellado ^{**}

**Ecole Normale Supérieure de Cachan. 61, av. du Président Wilson, 94235 Cachan Cedex, France
(email : Jean-Jacques.lesage @lurpa.ens-cachan.fr)*

***CINVESTAV Unidad Guadalajara. Av. del Bosque 1145, Col. El Bajío 45019 Zapopan, Mexico
(email : aestrada/elopez@gdl.cinvestav.mx)*

Résumé : Dans cet article nous présentons une approche d'identification passive et de type boîte noire pour les Systèmes à Événements Discrets (SED), constitués d'un contrôleur et d'un procédé interagissant en boucle fermée. Cette approche permet de construire de manière systématique un réseau de Petri interprété (RdPI) qui modélise le comportement du SED, à partir de la seule observation des signaux échangés entre contrôleur et procédé. Dans un premier temps, nous montrons pourquoi les approches d'identification existantes sont encore mal adaptées à la prise en compte des contraintes physiques et technologiques des systèmes à identifier. Par la suite nous développons notre approche, qui se caractérise par la génération d'un RdPI qui exprime directement par sa structure les comportements caractéristiques des SED, comme les parallélismes, les alternatives ou les partages de ressources.

Mots-clés : Systèmes à événements discrets, identification, réseaux de Petri, expérimentation.

1. ETAT DE L'ART

L'identification consiste à rechercher un modèle mathématique d'un système dynamique à partir de données expérimentales et de connaissances disponibles a priori. Dans le cas des SED, les modèles recherchés sont des machines à états (Automates Finis (AF), RdP, ...). On parle d'identification *boîte noire* lorsque l'on ne dispose d'aucune connaissance a priori concernant le système à identifier ; l'identification est dite *passive* lorsque le processus d'identification se déroule pendant le fonctionnement normal du système, qui n'est alors soumis qu'aux sollicitations ambiantes. Dans ce travail, nous nous intéressons à l'identification passive et de type boîte noire des SED.

Ce n'est que récemment que la communauté scientifique internationale s'est intéressée à la recherche d'algorithmes d'identification spécifiquement dédiés aux SED. Une étude détaillée peut en être trouvée dans (Estrada-Vargas, 2010a), nous en présentons ci-dessous une synthèse non exhaustive.

Plusieurs des approches proposées sont assez analogues aux techniques d'apprentissage de langages, développées dans le domaine de l'informatique (Gold 1967). Ainsi, dans (Cabasino, 2007) et (Cabasino, 2008) on construit un RdP qui génère le même langage que celui généré par le SED identifié. Plusieurs hypothèses retenues dans ces travaux en limitent fortement la praticabilité. Tout d'abord il est supposé que le langage généré par le SED est connu dans sa totalité (hypothèse qui a été invalidée dans (Roth, 2010a)). Par ailleurs, toutes les transitions du RdP sont supposées connues, ce qui nécessite une connaissance a priori du comportement du système incompatible avec une approche d'identification de type boîte noire.

Dans (Ould El Medhi, 2006) plusieurs algorithmes sont proposés pour synthétiser un RdP par propagation d'un ensemble d'événements. Aucune distinction n'étant faite entre les événements d'entrée et de sortie du SED, le RdP ainsi construit ne peut traduire les relations causales d'entrées-sorties qui sont le propre de tout système réactif.

Une construction incrémentale d'un RdP interprété (RdPI) à partir d'une séquence observée des sorties du contrôleur est proposée dans (Meda-Campaña, 2005). Par hypothèse, cette approche n'est valide que pour les systèmes dont le comportement peut être caractérisé par la seule évolution de leurs sorties, ce qui encore une fois ne correspond pas aux besoins pour l'identification des systèmes réactifs.

Les évolutions observées des entrées et des sorties du SED sont prises en compte dans (Klein, 2005). Le modèle identifié est donné sous la forme d'un AF, bien adapté aux objectifs de diagnostic revendiqués par les auteurs, mais qui constitue un modèle sémantiquement trop pauvre pour une représentation performante et explicite des comportements d'un SED complexe. Ainsi par exemple, les parallélismes sont « mis à plat » dans l'AF identifié ou, dans l'approche distribuée proposée dans (Roth, 2010b), répartis dans les AF associés aux différents sous-systèmes. Par ailleurs, seuls les SED ayant un comportement cyclique sont considérés.

Dans (Dotoli, 2008), un RdPI est construit de manière à reproduire une séquence d'événements et les sorties correspondantes. La relation de causalité entre la séquence d'événements et les sorties doit être déterministe. De plus, l'ensemble des événements à prendre en compte doit être connu a priori, ce qui encore une fois n'est pas compatible avec une approche d'identification de type boîte noire.

2. PROBLEMATIQUE ET OBJECTIFS DES TRAVAUX

Il résulte de l'état de l'art que l'identification des SED est un domaine de recherche récent dans lequel des premiers résultats théoriques prometteurs ont été obtenus. Il convient désormais de les prolonger, de manière à faire de l'identification une technique expérimentale performante, offrant une réelle alternative à la modélisation par connaissance. Pour cela, les challenges à relever portent sur l'essentiel sur deux aspects :

- i) la capacité à prendre en compte les contraintes physiques et technologiques des SED industriels. Il faut donc : manipuler les signaux échangés entre contrôleur et processus et non des événements théoriques dénués de sens physique, prendre en compte l'impossibilité d'observer la totalité de l'espace d'état du SED ou du langage qu'il génère en un temps fini, intégrer les algorithmes de traitement des contrôleurs industriels (Automates Programmables Industriels (API), ...) et leur incidence sur le traitement des informations échangées avec le procédé, ...
- ii) la proposition d'algorithmes qui permettent d'identifier des systèmes de taille réaliste (plusieurs dizaines ou centaines d'E/S) en un temps acceptable, même si les algorithmes d'identification sont classiquement exécutés hors ligne, à partir de données expérimentales acquises en temps réel.

Dans des publications antérieures (Estrada-Vargas, 2010b), (Estrada-Vargas, 2011), nous avons proposé un algorithme d'identification performant, basé sur la reproduction du langage généré par le SED et partiellement observé, par un RdPI et visant à répondre aux objectifs ci-dessus. Cependant, le RdP ainsi généré est un graphe d'état et ne permet pas de traduire dans sa structure les comportements caractéristiques des SED (comme les parallélismes, les alternatives, les partages de ressources ...).

Dans cet article, nous proposons une approche radicalement différente, basée sur l'étude des relations causales entre le contrôleur et le procédé. Pour ce faire, après avoir rappelé les bases et les notations des RdPI, nous indiquons les problèmes physiques et expérimentaux posés par l'identification des SED industriels dans la section 4. Notre approche est ensuite présentée dans la section 5 et la construction du RdPI qui représente le seul comportement observable du système est détaillé dans la section 6.

3. BASES ET NOTATIONS DES RDPI

Définition 1 : Un RdP ordinaire non marqué G est un quadruplet $G = (P, T, I, O)$, où : $P = \{p_1, p_2, \dots, p_{|P|}\}$ et $T = \{t_1, t_2, \dots, t_{|T|}\}$ sont des ensembles finis disjoints de places et de transitions ; $I(O) : P \times T \rightarrow \{0,1\}$ est une application qui définit les arcs orientés reliant les places aux transitions (les transitions aux places).

La matrice d'incidence de G est $C = C^+ - C^-$, où $C^- = [c_{ij}^-]$; $c_{ij}^- = I(p_i, t_j)$ et $C^+ = [c_{ij}^+]$; $c_{ij}^+ = O(p_i, t_j)$ sont respectivement les matrices d'incidence avant et arrière. $M : P \rightarrow Z^+$ permet d'associer un marquage (nombre entier positif de jetons) à chaque place, il est usuel de représenter le marquage d'un RdP par un vecteur de dimension $|P|$. Un RdP *sauf* est tel que le marquage de chaque place est borné à 1.

Définition 2 : Un réseau de Petri marqué $N = (G, M_0)$ est un RdP ordinaire G doté d'un marquage initial M_0 .

Dans un RdP marqué, une transition t_j est validée pour un marquage M_k si $\forall p_i \in P, M_k(p_i) \geq I(p_i, t_j)$. Une transition validée t_j peut être tirée. Un nouveau marquage M_{k+1} est alors atteint. Le nouveau marquage est $M_{k+1} = M_k + Cu_k$, où $u_k(i) = 0, i \neq j, u_k(j) = 1$.

Les RdP interprétés (RdPI), tels que définis par exemple dans (David, 1994) permettent d'associer des variables d'entrées et de sorties à un RdP ordinaire.

Définition 3 : Un RdPI marqué (Q, M_0) est un RdP interprété $Q = (G, V, \Sigma, \Phi, \lambda, \varphi)$ doté d'un marquage initial M_0 , où : G est un RdP ordinaire, $V = \{v_1, v_2, \dots, v_r\}$ est un ensemble de variables, $\Sigma = \{\alpha_1, \alpha_2, \dots, \alpha_s\}$ est un ensemble d'événements, et $\Phi = \{\phi_1, \phi_2, \dots, \phi_q\}$ est un alphabet de sortie.

$\lambda : T \rightarrow C \times E$ est une fonction de labellisation des transitions, où $C = \{C_1, C_2, \dots\}$ est un ensemble de conditions sur les variables et $E = \{E_1, E_2, \dots\}$ est un ensemble d'événements. Dans un RdPI, une transition t_j est tirée *si* la transition t_j est validée, *et si* la condition $C(T_j)$ est vraie, *lorsque* l'événement dans $E(T_j)$ occure.

$\varphi : R(Q, M_0) \rightarrow (Z^+)^q$ est une fonction de sortie, qui associe à chaque marquage de $R(Q, M_0)$ un vecteur de sortie de dimension $q = |\Phi|$. φ est représentée par une matrice $q \times |P|$, telle que si le symbole de sortie ϕ_i est présent (vrai) à chaque fois que $M(p_j) \geq 1$, alors $\varphi(i, j) = 1$, sinon $\varphi(i, j) = 0$.

L'équation fondamentale d'un RdPI est complétée avec la projection du marquage $Y_k = \varphi M_k$, où $Y_k \in (Z^+)^q$ est le k -ième vecteur de sortie du RdPI.

Définition 4 : Une place $p_i \in P$ est dite *observable* si le i -ème vecteur colonne de φ est non nul, i.e. $\varphi(\bullet, i) \neq 0$. Si ce n'est pas le cas, la place est dite *non-observable*. $P = P^m \cup P^n$ où P^m est l'ensemble des places observables et P^n est l'ensemble des places non-observables.

4. IDENTIFICATION EXPERIMENTALE DE SED

Nous nous intéressons aux SED composés d'un contrôleur (plus spécifiquement un API) et d'un procédé interagissant en boucle fermée. Les signaux d'entrée du contrôleur (sorties du procédé) notés I , sont générés par les capteurs du procédé. Les signaux de sortie du contrôleur (entrées du procédé) notés O , sont les ordres destinés aux actionneurs du procédé.

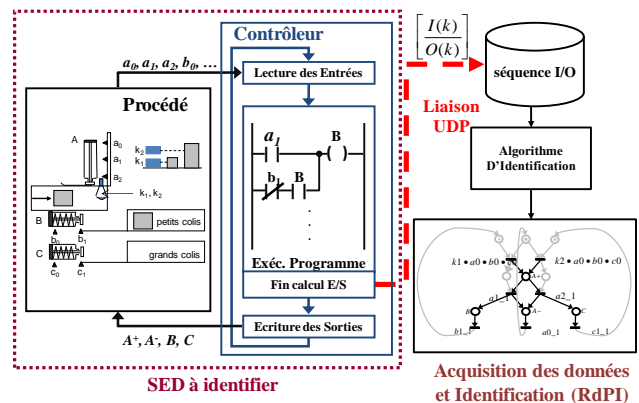


Fig. 1. Le SED à identifier et le processus d'identification.

L'observation du SED en fonctionnement est réalisée à partir des signaux d'entrée/sortie du contrôleur (Fig. 1) qui, lorsqu'il s'agit d'un API, réalise cycliquement trois tâches (indépendamment du langage de programmation utilisé) : la lecture de ses signaux d'entrée, l'exécution du programme pour calculer les nouvelles valeurs des sorties et l'émission des nouveaux signaux de sortie vers les actionneurs.

D'un point de vue expérimental, à chaque fin d'exécution du programme, la valeur courante de toutes les entrées et sorties de l'API (appelée *vecteur I/O*) est collectée et enregistrée dans une base de données. La seule donnée à partir de laquelle est réalisée l'identification est donc une séquence de vecteurs d'I/O, dont la taille dépend de la durée d'observation du système en fonctionnement, de la forme :

$$w = \begin{bmatrix} I(1) \\ O(1) \end{bmatrix}, \begin{bmatrix} I(2) \\ O(2) \end{bmatrix}, \begin{bmatrix} I(3) \\ O(3) \end{bmatrix}, \dots$$

$I(k)$ et $O(k)$ sont respectivement les valeurs des m entrées et des n sorties de l'API, capturées à la fin de son $k^{\text{ième}}$ cycle. Entre deux vecteurs I/O successifs, un *vecteur d'événements*, caractérisant l'évolution observée des entrées/sorties, peut être calculé comme : $E(k) = w(k) - w(k-1)$.

Chaque vecteur $E(k)$ peut par ailleurs être décomposé en deux vecteurs caractérisant pour l'un les évolutions des entrées $IE(k)$ et pour l'autre l'évolution des sorties $OE(k)$:

$$E(k) = \begin{bmatrix} IE(k) \\ OE(k) \end{bmatrix}$$

Au regard de son vecteur d'événements, toute évolution du système observé peut être caractérisée par l'un des quatre comportements suivants :

- 1) $IE(k) \neq 0 \wedge OE(k) \neq 0$
Une évolution des entrées a directement provoqué une évolution des sorties. La causalité I/O est observée dans le même cycle API.
- 2) $IE(k) = 0 \wedge OE(k) \neq 0$
Le contrôleur était arrivé au cycle $k-1$ (ou dans un cycle antérieur) dans un état instable dans lequel, sans nouvelle évolution de la valeur des entrées, une évolution d'état du contrôleur et de la valeur des sorties est permise au cycle k .
- 3) $IE(k) \neq 0 \wedge OE(k) = 0$
(on note $X(k)$ l'état du contrôleur au cycle k)
 - a) $X(k-1) = X(k)$; *une évolution de la valeur des entrées à laquelle le contrôleur n'est pas sensible est survenue.*
 - b) $X(k-1) \neq X(k)$; *une évolution non observable de l'état du contrôleur est survenue.*
- 4) $IE(k) = 0 \wedge OE(k) = 0$
 - a) $X(k-1) = X(k)$; *le contrôleur reste dans un état stable (aucune condition d'évolution d'état n'est satisfaite).*
 - b) $X(k-1) \neq X(k)$; *une évolution non observable de l'état du contrôleur est survenue.*

Tous ces comportements peuvent être traduits par un RdPI sauf : l'évolution des états du SED sera traduite par l'évolution du marquage du RdP (M_j) ; l'évolution des sorties (O_j) sera traduite par la fonction de projection du marquage (Y_j) ; selon que les évolutions d'états du contrôleur sont observables ou non, elles seront traduites par des éléments structurels liés à des places observables ou non-observables.

Même si cette distinction entre comportement observable et comportement non observable a déjà été abordée dans la littérature (Dotoli, 2008), seules les relations causales directes entrées/sorties de type 1, 2 et 3b ci-dessus (Estrada-

Vargas, 2011), ont été décrites jusqu'à présent, ce qui est très nettement insuffisant pour appréhender les systèmes réels. Pour modéliser les quatre comportements présentés plus haut, nous utiliserons des labels associés aux transitions du RdPI. Pour cela, les entrées du contrôleur constituent l'ensemble des variables du RdPI (V) et l'ensemble des événements du RdPI (Σ) est l'ensemble des événements (fronts montants et fronts descendants) sur les variables d'entrée, notés $\{i_{1-1}, i_{1-0}, i_{2-1}, \dots, i_{m-1}, i_{m-0}\}$. De la sorte, $E(T_k)$ pourra être exprimé comme une combinaison d'événements, et $C(T_k)$ comme une condition sur les signaux d'entrée de l'API.

5. NOTRE APPROCHE

Compte tenu de la classification des comportements possibles d'un SED donnée à la section précédente, nous pouvons distinguer deux étapes dans le processus d'identification :

- i) l'identification du *comportement observable* du SED,
- ii) l'identification de son *comportement non-observable*.

5.1 Identification du comportement observable d'un SED

Il faut chercher dans la séquence w de vecteurs I/O observée :

- i) si l'évolution de certaines entrées *provoque toujours* les mêmes évolutions de certaines sorties *dans le même cycle API* (comportement de type 1) ;
- ii) si l'évolution de certaines entrées provoque des évolutions de sorties *différées* (comportement de type 2) après un changement d'état non observable de type 3.b ;
- iii) si l'évolution de certaines entrées n'a *aucune incidence* sur l'évolution des sorties (comportement de type 3.a) ;
- iv) si l'état du contrôleur n'évolue pas (type 4.a) parce qu'il attend une évolution d'entrée à laquelle il est *sensible* (comportement de type 1 ou 2).

5.2 Identification du comportement non-observable d'un SED

Les évolutions de type 3.b et 4.b ne peuvent pas être déduites uniquement à partir de l'observation des évolutions des I/O et peuvent de plus être confondues avec d'autres types d'évolution (type 3.a et 4.a). Des techniques spécifiques d'inférence prenant en compte l'ordre d'apparition des événements peuvent alors être utilisées. Par manque de place, dans la suite de cet article nous ne détaillerons que l'identification du *comportement observable* du SED.

6. IDENTIF. DU COMPORTEMENT OBSERVABLE

L'algorithme 1 décrit la procédure globale d'identification du comportement observable d'un SED que nous avons retenue.

Algorithme 1.

Entrée : une séquence observée d'I/O : w

Sortie : matrice d'incidence observable φC et fonction de labellisation des transitions λ du RdPI sauf identifié

Etape 1. Analyser la séquence w et :

- Calculer la séquence de vecteurs d'événements
- Calculer les événements élémentaires
- Calculer les matrices de causalité directe et indirecte
- Construire les fonctions d'émission des événements de sortie
- Trouver les événements d'entrée à influence différée

Etape 2. Utiliser les données calculées dans l'étape 1 pour :

- Calculer les transitions du RdPI et leur label par λ
- Calculer la matrice d'incidence observable du RdPI φC

Chaque étape de cet algorithme va maintenant être détaillée en utilisant l'exemple du système manufacturier représenté sur la Fig. 2. La fonction de ce système est de trier des colis selon leur taille. Le contrôleur lit les signaux provenant de 9 capteurs : $a_0, a_1, a_2, b_0, b_1, c_0, c_1, k_1, k_2$, et doit générer des signaux à destination de 4 actionneurs : $A+, A-, B, C$. Bien que nous ayons traité cet exemple en observant des séquences d'évolution I/O de plusieurs centaines de vecteurs, nous ne manipulerons ici que de très courtes séquences pour des raisons évidentes de lisibilité.

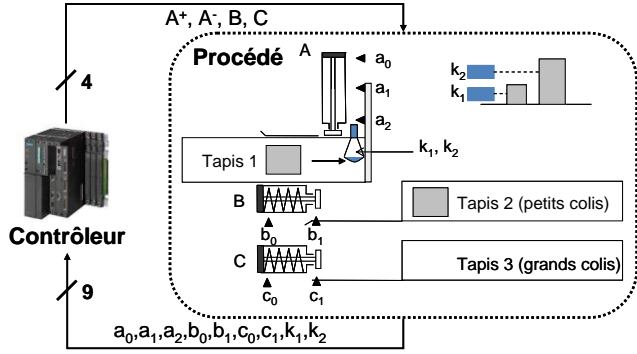


Fig. 2. Système de tri de colis

6.1 Calcul des séquences de vecteurs d'événements

Dans la séquence observée ci-après, un vecteur d'événements calculé par $E(k) = w(k) - w(k-1)$ est intercalé (en pointillés) entre chaque couple de vecteurs I/O (en traits pleins).

$k1$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$k2$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$a0$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$a1$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
$a2$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$b0$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$b1$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$c0$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
$c1$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$A+$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$A-$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
B	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
C	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Fig. 3. Séquence observée d'I/O

6.2 Calcul des événements élémentaires

Les vecteurs d'événements peuvent être décomposés comme un ensemble d'événements élémentaires sur les entrées-sorties (qui seront nommés *événements* dans la suite) :

$$IE(k) = \{IE_{k1}, IE_{k2}, \dots, IE_{kj}\} = \cup IE_{ki} \text{ t.q. } I_i(k+1) - I_i(k) \neq 0$$

$$OE(k) = \{OE_{k1}, OE_{k2}, \dots, OE_{kj}\} = \cup OE_{ki} \text{ t.q. } O_i(k+1) - O_i(k) \neq 0$$

Si aucun événement élémentaire d'entrée (resp. de sortie) n'occure dans $E(j)$ (resp. $OE(j)$), nous noterons $IE(j) = \{\varepsilon\}$ (resp. $OE(j) = \{\varepsilon\}$). Le front montant de la variable v_i est noté v_{i_1} et son front descendant v_{i_0} .

Le tableau 1 donne les événements élémentaires calculés à partir des vecteurs d'événements de la Fig. 3.

Tableau 1. Evénements élémentaires de l'exemple 2

$E(k)$	Evénements élém. d'entrée	Evénements élém. de sortie
$E(1)$	$IE(1) = \{k1_1\}$	$OE(1) = \{A+_1\}$
$E(2)$	$IE(2) = \{a0_0\}$	$OE(2) = \{\varepsilon\}$
$E(3)$	$IE(3) = \{k1_0\}$	$OE(3) = \{\varepsilon\}$
$E(4)$	$IE(4) = \{a1_1\}$	$OE(4) = \{A+_0, A-_1, B_1\}$
$E(5)$	$IE(5) = \{b0_0\}$	$OE(5) = \{\varepsilon\}$

6.3 Calcul des matrices de causalité directe et indirecte

Comme indiqué dans la section 4, l'influence de l'évolution de certaines entrées sur l'évolution des sorties peut très souvent être directement observée dans le même cycle API. Pour découvrir de telles relations causales directes entre entrées-sorties, nous calculons la probabilité conditionnelle d'occurrence d'un événement de sortie OE_k , sachant qu'un événement d'entrée IE_i a occurred dans le même cycle API :

$$Prob(OE_k | IE_i) = \frac{N_{Obs}(OE_k, IE_i)}{N_{Obs}(OE_k)}$$

L'ensemble des $Prob(OE_k | IE_i)$ est consigné sous la forme d'une *Matrice de Causalité Directe (DM)*, dans laquelle chaque $DM_{ik} = Prob(OE_k | IE_i)$. La matrice DM du système de tri, calculée à partir d'une séquence observée beaucoup plus longue que celle présentée en Fig. 3 (plusieurs centaines de vecteurs), est donnée sur la Fig. 4. Des comportements de type 1 (cf. section 4) peuvent être simplement extraits de la matrice DM en observant pour chaque colonne OE_k si des sommes de probabilités OE_k / IE_j sont égales à 1. Toute occurrence d'un des IE_j provoque alors en effet une occurrence certaine de l'événement OE_k . Par exemple, l'occurrence d'un des événements $a1_1$ ou $a2_1$ provoque *toujours* l'occurrence de $A+_0$.

De même, les comportements de type 3.a peuvent être découverts dans la matrice DM sous la forme de lignes nulles. Par exemple, la seconde ligne étant uniformément nulle indique que le contrôleur n'est à aucun moment *directement sensible* à l'événement d'entrée $k1_0$.

	A+_1	A+_0	A-_1	A-_0	B_1	B_0	C_1	C_0
$k1_1$	0.111	0.111	0.111	0.111	0.000	0.200	0.000	0.000
$k1_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$k2_1$	0.222	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$k2_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$a0_1$	0.222	0.000	0.000	1.000	0.000	0.000	0.000	0.000
$a0_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$a1_1$	0.000	0.444	0.444	0.000	1.000	0.000	0.000	0.000
$a1_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$a2_1$	0.000	0.556	0.556	0.000	0.000	0.000	1.000	0.000
$a2_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$b0_1$	0.333	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$b0_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$b1_1$	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
$b1_0$	0.000	0.000	0.000	0.111	0.000	0.000	0.000	0.000
$c0_1$	0.111	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$c0_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$c1_1$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$c1_0$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Fig. 4. Matrice de causalité directe pour le système de tri

Pour découvrir les relations causales non directes entre entrées-sorties, il faut cette fois observer la valeur des entrées lorsque les événements de sortie occurrent. Nous calculons pour cela la probabilité d'occurrence d'un événement de

sortie OE_k , sachant que certaines entrées ont une valeur donnée IL_i dans le même cycle API :

$$Prob(OE_k | IL_i) = \frac{N_{obs}(OE_k, IL_i)}{N_{obs}(OE_k)}$$

Nous construisons alors la *matrice de causalité indirecte* (ou de contexte) (IM), dans laquelle chaque $IM_{ik} = Prob(OE_k | IL_i)$. La matrice IM du système de tri est donnée Fig. 5.

	A+ ₁	A+ ₀	A- ₁	A- ₀	B ₁	B ₀	C ₁	C ₀
k1=1	0.444	0.111	0.111	0.333	0.000	0.250	0.200	0.200
k1=0	0.556	0.889	0.889	0.667	1.000	0.750	0.800	0.800
k2=1	0.556	0.000	0.000	0.333	0.000	0.250	0.000	0.200
k2=0	0.444	1.000	1.000	0.667	1.000	0.750	1.000	0.800
a0=1	1.000	0.000	0.000	1.000	0.000	0.500	0.000	0.000
a0=0	0.000	1.000	1.000	0.000	1.000	0.500	1.000	1.000
a1=1	0.000	0.444	0.444	0.000	1.000	0.000	0.000	0.000
a1=0	1.000	0.556	0.556	1.000	0.000	1.000	1.000	1.000
a2=1	0.000	0.556	0.556	0.000	0.000	0.000	1.000	0.000
a2=0	1.000	0.444	0.444	1.000	1.000	1.000	0.000	1.000
b0=1	1.000	1.000	1.000	0.556	0.000	0.000	1.000	1.000
b0=0	0.000	0.000	0.000	0.444	1.000	1.000	0.000	0.000
b1=1	0.000	0.000	0.000	0.111	1.000	1.000	0.000	0.000
b1=0	1.000	1.000	1.000	0.889	0.000	0.000	1.000	1.000
c0=1	1.000	1.000	1.000	0.889	0.000	1.000	1.000	0.000
c0=0	0.000	0.000	0.000	0.111	1.000	0.000	0.000	1.000
c1=1	0.000	0.000	0.000	0.000	1.000	0.000	0.000	1.000
c1=0	1.000	1.000	1.000	1.000	0.000	1.000	1.000	0.000

Fig. 5. Matrice de causalité indirecte pour le système de tri

Dans la matrice IM nous pouvons maintenant mettre en évidence des comportements de type 3.b puis de type 2, en cherchant dans chaque colonne les valeurs de probabilités qui additionnées font 1 ET qui ne sont pas nulles dans la matrice DM . Par exemple, k1=1 ou k2=1, a0=1, b0=1 et c0=1 sont des valeurs d'entrées qui peuvent provoquer A+₁ (IM), même si elles ne sont pas toujours observées dans le même cycle API (DM).

6.4 Construction des fonctions d'émission des événements de sortie

De manière générale, l'occurrence de tout événement de sortie OE_k peut être causé par l'occurrence d'un ou de plusieurs événements d'entrée observés dans le même cycle API et/ou par la réalisation d'une condition logique sur certaines valeurs d'entrées. L'expression générique de la fonction d'émission des sorties est donc donnée par :

$$\chi(OE_k) = G(OE_k) \bullet F(OE_k)$$

où $G(OE_k)$ est une fonction d'événements d'entrées et $F(OE_k)$ est une fonction sur les valeurs logiques des entrées.

Nous calculons $G(OE_k)$ comme une conjonction de disjonctions :

$G(OE_k) = \Pi DisjE_j$, où chaque $DisjE_j = (IE_x \oplus \dots \oplus IE_z)$ est le OU exclusif d'événements d'entrée dont la somme des probabilités d'occurrence dans la matrice DM vaut 1, i.e. qui satisfait les conditions :

$DM_{xj} \neq 0, DM_{yj} \neq 0, \dots, DM_{zj} \neq 0$ et $DM_{xj} + DM_{yj} + \dots + DM_{zj} = 1$
 $F(OE_k)$ est calculé de manière similaire : $F(OE_k) = \Pi DisjL_j$

avec $DisjL_j = (IL_x \oplus IL_y \oplus \dots \oplus IL_z)$, tel que :

$IM_{xj} \neq 0, IM_{yj} \neq 0, \dots, IM_{zj} \neq 0$ et $IM_{xj} + IM_{yj} + \dots + IM_{zj} = 1$
et $DM_{xj} \neq 0, DM_{yj} \neq 0, \dots, DM_{zj} \neq 0$

6.5 Événements d'entrée à influence différée

L'influence d'un changement de valeur d'une ou plusieurs entrées sur un changement d'état du contrôleur ou/et un changement de valeur de sorties peut survenir plusieurs cycles API après son occurrence. Certains événements d'entrée pourraient donc être ignorés s'ils n'étaient intégrés dans aucune condition, c'est-à-dire s'ils ne provoquent jamais directement (dans le même cycle API) d'évolution de sortie(s). Pour éviter cela, nous construisons le sous-ensemble des entrées $D = \{I_i\}$ tel que les lignes de la matrice DM correspondant à I_{i-0} et à I_{i-1} sont toutes les deux nulles. Nous appelons D l'ensemble des *entrées à influence différée sur les sorties*. Ces entrées correspondent dans la pratique le plus souvent à des informations mémorisées, mais dans notre exemple $D = \{\}$.

Finalement, pour tout signal de sortie O_i , nous pouvons écrire les conditions d'activation O_{i-1} et de désactivation O_{i-0} comme des fragments d'un RdPI sauf donnés sur la Fig. 6.

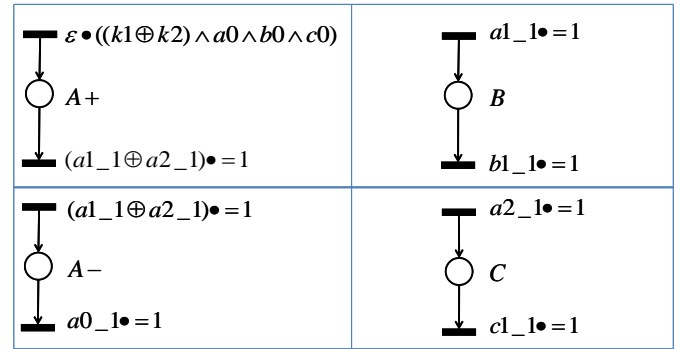


Fig. 6. Conditions d'activation et de désactivation des sorties

6.6 Des conditions d'activation aux matrices d'incidence

Dans le cas général, à chaque cycle API une ou plusieurs conditions sur l'état et les entrées peuvent conduire à l'émission d'une ou plusieurs évolutions de sorties *simultanées*. Ce synchronisme d'activations simultanées peut être explicitement représenté en RdPI par le franchissement d'une unique transition tel que sur la Fig. 7, obtenue par fusion de fragments de RdP obtenus sur la Fig. 6.

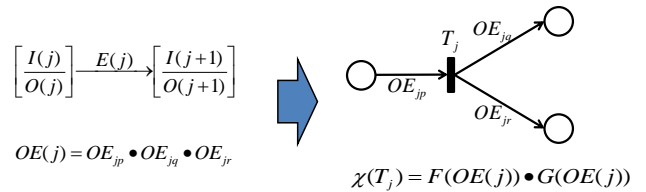


Fig. 7. Activation synchrone de sorties en RdPI

Ce mécanisme de fusion peut être construit systématiquement par la procédure donnée dans l'algorithme 2.

Pour l'exemple du tri de colis (Fig. 2), l'algorithme 2 permet de construire la partie du RdPI de la Fig. 8 représentée en trait plein. Ce RdPI sauf modélise le comportement

observable de ce SED, construit à partir de la seule séquence observée des signaux échangés entre le contrôleur et le procédé.

Algorithme 2.

Entrées : séquence I/O : w , séquence d'événements I/O : E , Matrices DM et IM , ensemble des entrées différées D

Sorties : matrices d'incidence observable φC et fonction de labellisation des transitions λ

$P \leftarrow \{p_1, p_2, \dots, p_n\}$
 $\forall j=1, \dots, |E|$
Si $OE(j) = 0$ **et** $\exists IE_s, \dots, IE_u \in IE(j) \cap D$
Alors $T \leftarrow T \cup t_j$, $\lambda(t_j) = IE_s \bullet \dots \bullet IE_u$, $\forall p_i \in P \varphi C(t_j, p_i) \leftarrow 0$
Si $OE(j) \neq 0$
Alors $\forall OE_{jk} \in OE(j)$
 $\forall DisjE_i \in G(OE_{jk})$, $DisjE_i' \leftarrow DisjE_i \cap IE(j)$
 $G'(OE_{jk}) \leftarrow \Pi DisjE_i'$
 $G(OE(j)) \leftarrow \Pi G'(OE_{jk})$
 $\forall DisjL_i \in F(OE_{jk})$, $DisjL_i' \leftarrow DisjL_i \cap I(j+1)$
 $F'(OE_{jk}) \leftarrow \Pi DisjL_i'$
 $F(OE(j)) \leftarrow \Pi F'(OE_{jk})$
 $T \leftarrow T \cup t_j$, $\lambda(t_j) = F(OE(j)) \bullet G(OE(j))$
 $\forall p_i \in P$
Si $O_{q-1} \in OE(j)$ **Alors** $\varphi C(t_j, p_q) \leftarrow 1$
Sinon Si $O_{q-0} \in OE(j)$ $\varphi C(t_j, p_q) \leftarrow -1$
Sinon $\varphi C(t_j, p_q) \leftarrow 0$

Même si la procédure d'inférence permettant de construire le comportement *non observable* du SED identifié ne peut pas être décrite dans cet article faute de place, le résultat en est tout de même représenté sur la Fig. 8 en traits grisés à titre d'information.

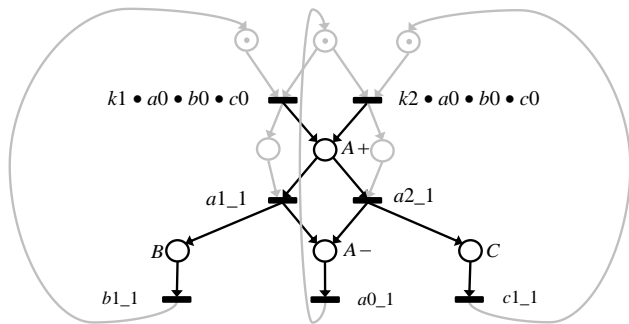


Fig. 8. Réseau de Petri identifié

Par ailleurs, le lecteur remarquera que des fonctions d'activation/désactivation de sorties non triviales ont pu être construites. Par exemple la sortie A est activée ($A+_{-1}$) par la présence d'un colis d'une dimension quelconque ($k1 = 1$ ou $k2 = 1$), seulement si les trois vérins A, B et C sont à leur position initiale ($a0 = 1$, $b0 = 1$ et $c0 = 1$). Aucune des approches existantes d'identification de SED ne permet la construction d'un tel modèle par une approche boîte noire à partir de la seule observation de séquences I/O.

Sur le plan expérimental, tous nos algorithmes d'identification sont implémentés et ont permis de nombreux tests. Ils sont de complexité linéaire ou polynomiale. L'acquisition des signaux permettant la construction de la séquence d'I/O (w) est réalisée grâce à une connexion UDP (User Datagram Protocol), qui n'introduit aucune perturbation causale ni temporelle dans le fonctionnement normal de l'API (Roth, 2010a).

7. CONCLUSIONS

Nous avons présenté une méthode d'identification des SED basée sur l'analyse des signaux échangés entre contrôleur et procédé. Le RdPI sauf ainsi construit donne une représentation concise du comportement observable du SED puisque ses transitions sont étiquetées par des conditions fonctionnellement suffisantes à leur franchissement. De plus ce RdPI est notablement plus explicite que ceux obtenus par les méthodes existantes car il exprime directement par sa structure les comportements caractéristiques des SED (comme les parallélismes, les alternatives, les partages de ressources, ...). Nos travaux en cours portent sur l'identification du comportement non-observable des SED et sur des simplifications possibles dans le RdPI ainsi obtenu.

RÉFÉRENCES

Cabasino M.P., Giua A., Seatzu C. (2007), Identification of Petri Nets from Knowledge of Their Language, *Discrete Event Dynamic Systems*, 17(4), pp. 447-474

Cabasino M.P., Giua A., Seatzu C. (2008), Linear Programming Techniques for the Identification of Place/Transition Nets, *IEEE Int. Conf. on Decision & Control*, Cancun, Mexico, pp. 514-520

David R. and Alla H. (1994), Petri Nets for Modeling of Dynamic Systems, a survey, *Automatica*, 30(2), pp. 175-202

Dotoli M., Fanti M.P., Mangini A.M. (2008), Real time identification of discrete event systems using Petri nets, *Automatica*, 44(5), pp. 1209-1219

Estrada-Vargas A.P., Lopez-Mellado E., Lesage J.-J. (2010a), A Comparative analysis of recent identification approaches for DES, *Mathematical Problems in Engineering*, Vol. 2010, Article ID 453254, 21 pages

Estrada-Vargas A.P., Lopez-Mellado E., Lesage J.-J. (2010b), An Identification Method for PLC-based Automated Discrete Event Systems, *IEEE Int. Conf. on Decision and Control*, Atlanta, USA, pp.6740-6746

Estrada-Vargas A. P., Lesage J.-J., López-Mellado E. (2011), Stepwise Identification of Automated Discrete Manufacturing Systems, *IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Toulouse, France, paper N°4-6, 8 pages

Gold E.M. (1967), Language identification in the limit, *Information and Control*, 10(5), pp. 447-474

Klein S., Litz L., Lesage J.-J. (2005), Fault detection of Discrete Event Systems using an identification approach, *16th IFAC World Congress*, 6 pages, Praha, Czech Rep.

Meda-Campaña M., López-Mellado E. (2005), Identification of Concurrent Discrete Event Systems Using Petri Nets, *IMACS'05 World Congress*, Paris, France, pp.1-7

Ould El Medhi S., Leclercq E., Lefebvre D. (2006), Petri nets design and identification for the diagnosis of discrete event systems, *IAR Annual Meeting*, Nancy, France

Roth M., Lesage J.-J., Litz L. (2010a), Identification of discrete event systems: Implementation issues and model completeness, *7th Int. Conf. on Informatics in Control Automation and Robotics*, Funchal (Portugal), pp. 73-80

Roth M., Lesage J.-J., Litz L. (2010b), Black-box identification of discrete event systems with optimal partitioning of concurrent subsystems, *American Control Conference*, Baltimore, Maryland, USA, pp. 2601-2606