

Qu'est ce qu'un bon modèle ?

Robert Valette

LAAS-CNRS

F- 31077 Toulouse Cedex 4

e-mail robert@laas.fr

[http ://www.laas.fr/~robert](http://www.laas.fr/~robert)

Version Juillet 1999

1 Introduction

Un modèle formel est une représentation mathématique, abstraite et toujours approchée d'un système réel. Il représente une certaine vue de ce système. Il est le résultat d'une démarche cognitive complexe qui progressivement rend formel et explicite un ensemble d'exigences, de contraintes, de connaissances informelles et partiellement implicites. Une automatisation d'un tel processus paraît bien difficile.

Les exigences (besoins) sont complexes et comprennent toujours des informations concernant les coûts, la sécurité, la fiabilité, la faciliter de fabriquer, de maintenir, de faire évoluer etc. Une modélisation formelle consiste à extraire un fragment de toutes ces exigences. Cette extraction doit être guidée par un but, mais elle est également fortement liée au cadre mathématique choisi pour la modélisation. Dans le contexte de la théorie des réseaux de Petri, ce fragment sera essentiellement construit autour de la structure de contrôle. Une structure de contrôle est une suite d'événements (transitions) et d'activités (places). Elle est souvent vue comme un ensemble de processus communicants.

Le but principal d'une modélisation formelle est en fait souvent de rendre clair et explicite un ensemble d'informations largement implicites. Lors de la construction du modèle formel, un grand nombre d'ambiguïtés doivent être levées. Lever une ambiguïté consiste en général à prendre une décision, à effectuer un choix. Pendant ce processus, des incomplétudes et des incohérences seront mises en évidence. Ceci entraînera d'autres décisions et d'autres choix. Le but de la vérification souvent associée à toute modélisation formelle est d'augmenter la confiance du concepteur du modèle vis-à-vis de son travail (mon modèle correspond-il bien au système étudié, est-il effectivement dépourvu d'incohérences, d'imprécisions etc) et à être convaincant vis-à-vis de ses collègues travaillant sur le même projet (consolidation et validation du modèle). Eventuellement il peut aider dans un processus de certification vis-à-vis d'une autorité externe de contrôle.

Un bon modèle est donc d'abord un modèle que l'on comprend bien et que l'on peut facilement expliquer à ses collègues. Les procédures de vérification doivent être simples et convaincantes. Il faut

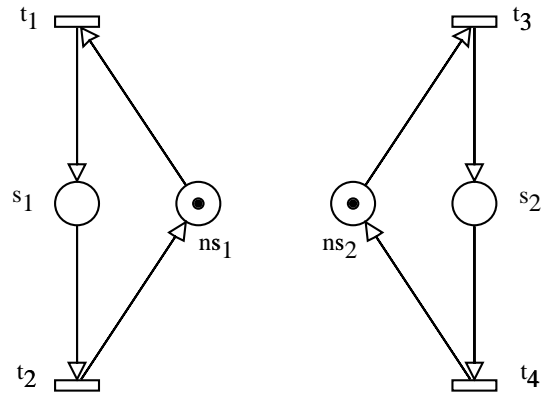


FIG. 1: Les deux sections critiques

si possible que les exigences, les contraintes et les informations exprimées par le modèle puissent être clairement identifiées. La modularité est une caractéristique importante. Un bon modèle n'est pas nécessairement un modèle minimal (par exemple en terme de nombre de places et de transitions pour un réseau de Petri. Bien sûr, il existe une dernière contrainte qui est souvent forte. Souvent la modélisation est faite dans le cadre de l'utilisation d'un outil, par exemple de simulation. Il faut alors que l'approche de modélisation soit compatible avec les possibilités de l'outil et qu'elle n'utilise pas des techniques de modélisation susceptibles d'entraîner une perte d'efficacité de l'outil. Dans le cas d'un simulateur, certaines configuration peuvent en effet entraîner des calculs inutiles. L'utilisation d'un outil peut s'avérer très contraignante. Le texte qui suit se veut générique et donc indépendant de tout outil.

Définir une démarche générique pour obtenir un bon modèle semble difficile, même dans le cadre d'une théorie bien précise comme celle des réseaux de Petri. La démarche doit en effet s'adapter aux caractéristiques du problème traité et comme nous venons de le voir à l'outil utilisé. Nous allons ici simplement présenter deux exemples simples en montrant, pour chacun, deux façons de modéliser et en comparant les qualités des modèles obtenus.

2 L'exemple de l'exclusion mutuelle

Considérons un exemple extrêmement simple : un mécanisme d'exclusion mutuelle entre deux sections critiques. La figure 1 montre le problème initial. Pour chaque section, on ne différencie que l'état du processus hors de la section (places ns_1 et ns_2) et l'état du processus lorsqu'il est en train d'exécuter la section (places s_1 et s_2).

2.1 Première approche de modélisation

Le problème consiste à empêcher l'entrée du second processus dans sa section critique si le premier est en train d'exécuter la sienne et réciproquement. Une solution qui paraît naturelle est de rajouter des tests aux transitions t_1 et t_3 d'entrée dans les sections critiques. Ces test sont des boucles élémentaires permettant de n'autoriser l'entrée d'un processus dans une section critique que lorsque l'autre n'y est pas. On obtient alors le réseau de la figure 2. Si l'on énumère l'ensemble des marquages accessibles de

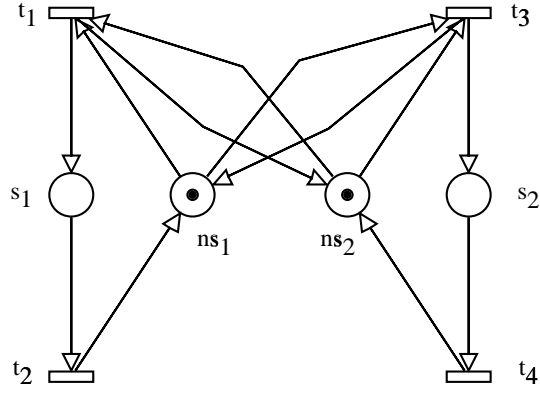


FIG. 2: Représentation intuitive de l'exclusion mutuelle

ce réseau, on trouve que les seuls états possibles sont $ns_1 \otimes ns_2$ (le connecteur \otimes exprime simplement la présence simultanée de jetons dans les places correspondantes) qui est le marquage initial, $s_1 \otimes ns_2$ et $ns_1 \otimes s_2$. Donc le modèle est correct.

Toutefois, la preuve de l'exclusion mutuelle ne peut pas être faite par l'approche algébrique, c'est-à-dire par la recherche d'un invariant de place f adéquat en résolvant l'équation :

$$f^T \cdot C = 0 \quad (1)$$

où f^T est le transposé d'un vecteur associant une pondération à chaque place et C est la matrice d'incidence.

En effet, la matrice d'incidence ne prend pas en compte les boucles élémentaires, c'est-à-dire que la matrice d'incidence du réseau de la figure 2 est exactement la même que celle du réseau de la figure 1 ; et donc les solutions f également. Or, dans le cas du réseau de la figure 1, il est clair que le marquage $s_1 \otimes s_2$ est accessible.

L'exemple considéré ici est trop simple pour que la nécessité d'énumérer les états soit un problème. Dans un cas complexe la situation peut être différente. De plus, nous rappelons que les résultats d'une étude par énumération des marquages est remise en cause par toute modification (ajout d'un jeton ou composition avec un autre réseau de Petri) alors que l'approche fondée sur des invariants de place ne sera, par exemple, pas remise en cause si l'on fusionne le réseau considéré avec un autre uniquement par fusion de transitions.

2.2 Deuxième approche de modélisation

On cherche d'abord à décrire les états des diverses entités impliquées dans le système, puis on construit le réseau de Petri en combinant les divers sous-modèles. Il faut dans le cas présent remonter aux entités. Il y a bien sûr les deux processus, dont on reprend la représentation de la figure 1. Mais il y a également la cause de l'exclusion mutuelle, qui provient d'une ressource (éventuellement virtuelle) partagée entre les deux processus. Seul l'un de ces processus peut l'utiliser à la fois. Les états de cette ressource sont :

- res lorsqu'elle est disponible,

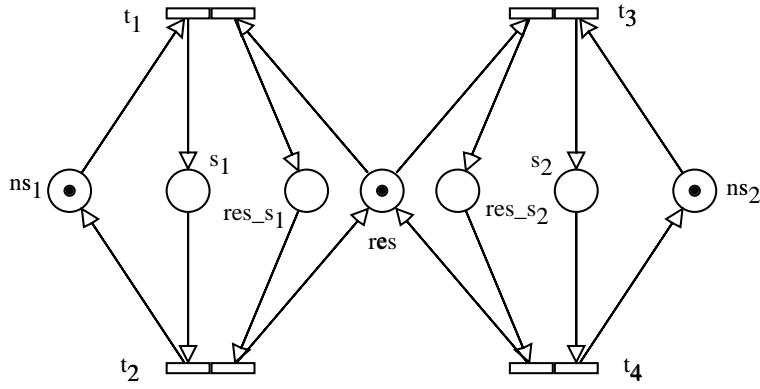


FIG. 3: Deuxième construction

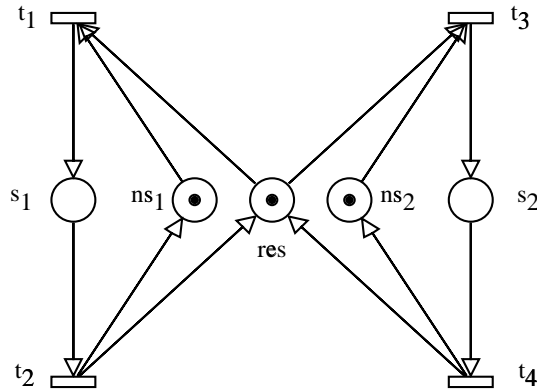


FIG. 4: Représentation explicitant la ressource

- res_{s_1} lorsqu'elle est utilisée par le processus 1 pour la section critique s_1 ,
- res_{s_2} lorsqu'elle est utilisée par le processus 2 pour la section critique s_2 .

Il suffit alors de remarquer que les événements correspondant à l'entrée dans la section critique s_1 par le processus 1 et le changement d'état de la ressource de res à res_{s_1} sont identiques (et respectivement pour le processus 2). Puis qu'il en est de même pour la sortie de la section critique. On obtient donc le réseau de la figure 3 par fusion des transitions correspondant aux événements communs.

On peut alors remarquer que les états *ressource occupée par la section critique 1* et *exécution de la section critique 1* sont les mêmes et qu'il est inutile de la coder par les deux places identiques s_1 et res_{s_1} . Nous obtenons alors le réseau de la figure 4.

Ce réseau possède trois invariants de place :

$$M(ns_1) + M(s_1) = 1 \quad (2)$$

$$M(ns_2) + M(s_2) = 1 \quad (3)$$

$$M(res) + M(s_1) + M(s_2) = 1 \quad (4)$$

Nous pouvons noter que l'invariant 4 prouve qu'il n'y a jamais simultanément un jeton dans les places s_1 et s_2 , c'est-à-dire qu'il prouve l'exclusion mutuelle.

Si ce réseau est un motif composé avec un réseau global par la fusion d'une (ou de plusieurs) des transitions t_1, t_2, t_3 ou t_4 , la preuve sera préservée. Cette représentation, qui peut paraître moins naturelle au premier abord que la précédente est donc meilleure.

Ce qui différencie fortement les deux approches, c'est que dans le second cas, l'ensemble des marquages accessibles est exactement caractérisé par l'ensemble des solutions de l'équation fondamentale :

$$M = M_0 + C \cdot \bar{s} \quad (5)$$

3 Les feux de carrefour Allemands

Cet exemple est tiré de :

Jörg Desel

How to model traffic lights

Petri net newsletter 56, Gesellschaft für Informatik, Fachgruppe 0.0.1

En Allemagne, les couleurs diffèrent légèrement de celles des feux Français. Le cycle comprends 4 états différents :

$$m_0; m_1; m_2; m_3; m_0; \dots \quad (6)$$

avec

- dans l'état m_0 , la lampe rouge (*red*) est allumée (“*on*”),
- dans l'état m_1 , les lampes rouges (*red*) et jaune (*yellow*) sont allumées (“*on*”),
- dans l'état m_2 , la lampe verte (*green*) est “*on*”,
- dans l'état m_3 , la lampe jaune (*yellow*) est “*on*”.

La première partie des spécification (la succession des 4 états) sera considérée comme la partie commande (*control*) et la seconde, celle qui spécifie quelles lampes sont allumées sera l'affichage (*display*).

D'une façon évidente, le réseau de Petri de la figure 5 représente la partie commande *control*.

Maintenant il faut rajouter les spécifications concernant les couleurs des lampes allumées. On va pour cela rajouter des places au réseau de Petri, mais il faudra vérifier à la fin que ces places ne modifient en rien la séquence des états spécifiée par la partie commande. Nous ajoutons une place par couleur de lampe. Quand ces places contiennent un jeton, les lampes correspondantes sont allumées, sinon elles sont éteintes. Il faudra vérifier que ces places ne contiennent jamais plus d'un jeton puisqu'elles sont considérées comme étant des représentations de conditions logiques vraies ou fausses. Les trois places sont nommées : *Red*, *Yellow* et *Green*. Nous obtenons alors le réseau de la figure 6. Il a été obtenu par la composition de deux spécifications, la partie commande et l'affichage (*display*).

4 Analyse

Le but de cette analyse est de montrer qu'il n'y a pas de contradiction entre les deux parties de la spécification. Son but est également de montrer que certaines propriétés du réseau de Petri correspondent bien au comportement du système physique. Cela permet au concepteur du modèle d'accroître la confiance

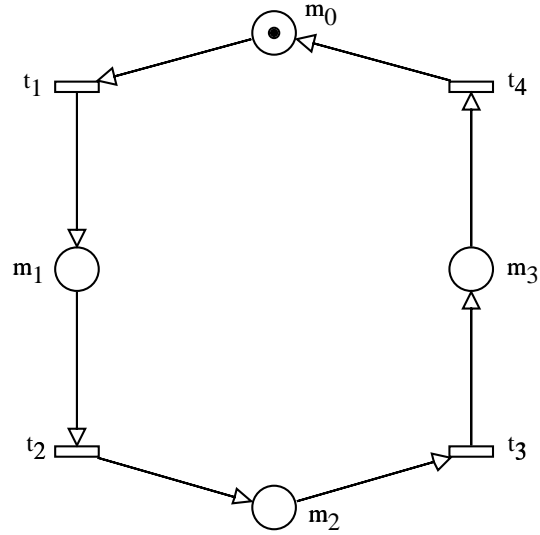


FIG. 5: Réseau de Petri de la partie commande

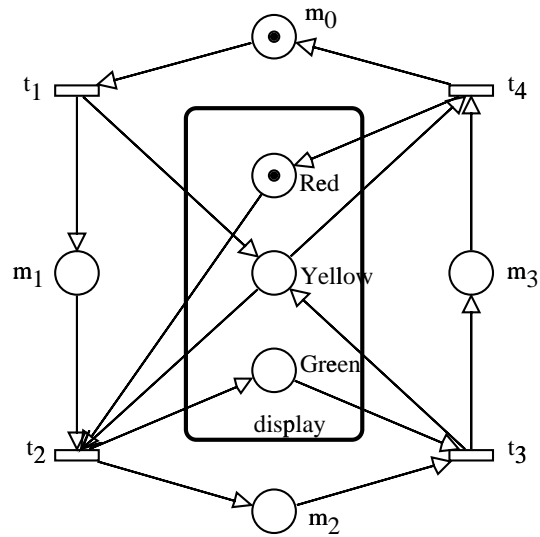


FIG. 6: Réseau de Petri de la commande et de l'affichage

qu'il a dans son modèle et de convaincre ses collègues participant au projet que le modèle est correct et qu'il est donc susceptible d'apporter des informations utiles.

La matrice d'incidence du réseau de la figure 6 est :

$$C = \begin{array}{cccc|l} & t_1 & t_2 & t_3 & t_4 & \\ \hline & -1 & 0 & 0 & 1 & m_0 \\ & 1 & -1 & 0 & 0 & m_1 \\ & 0 & 1 & -1 & 0 & m_2 \\ & 0 & 0 & 1 & -1 & m_3 \\ & 0 & -1 & 0 & 1 & Red \\ & 1 & -1 & 1 & -1 & Yellow \\ & 0 & 1 & -1 & 0 & Green \end{array} \quad (7)$$

4.1 Est-ce que l'affichage est cohérent avec la commande ?

De la matrice d'incidence, nous pouvons déduire l'invariant de place suivant (il associe le poids -1 à la place *Red*, le poids 1 aux places m_0 et m_1 et 0 aux autres) :

$$M(Red) = M(m_0) + M(m_1) \quad (8)$$

Comme la transition t_2 (sortie de la place *Red*) est également sortie de la place m_1 alors la place *Red* est implicite vis-à-vis des places m_0 et m_1 (elle n'ajoute aucune contrainte supplémentaire au franchissement de t_2).

De même nous avons :

$$M(Yellow) = M(m_1) + M(m_3) \quad (9)$$

et la transition t_2 (respectivement t_4) est également sortie de m_1 (respectivement m_3). Donc la place *Green* est implicite vis-à-vis de m_2 .

De même nous avons :

$$M(Green) = M(m_2) \quad (10)$$

et la transition t_3 est sortie de m_2 . Donc la place *Yellow* est implicite vis-à-vis de m_1 et m_3 .

Cela prouve que la spécification de l'affichage n'ajoute pas de contrainte de nature séquentielle à la spécification de la commande (les places implicites ne servent à rien, elles ne modifient pas les séquences de franchissement des transitions).

4.2 Vérifier que la partie commande est cohérente

Clairement le réseau de Petri de la figure 5 est borné vivant et réinitialisable. Comme le réseau de la figure 6 se ramène au précédent en appliquant trois fois la règle "*place implicite*", il est également borné, vivant et réinitialisable.

L'invariant de place suivant :

$$M(m_0) + M(m_1) + M(m_2) + M(m_3) = 1 \quad (11)$$

peut être déduit à partir de la matrice d'incidence du réseau 6 et nous pouvons déduire les inégalités suivantes :

$$M(m_0) \leq 1 \quad (12)$$

$$M(m_1) \leq 1 \quad (13)$$

$$M(m_2) \leq 1 \quad (14)$$

$$M(m_3) \leq 1 \quad (15)$$

La structure de commande est toujours dans un état et dans un seul état à la fois.

4.3 Vérifier que l'affichage est cohérent

Nous avons vu que les places étant implicites, elles ne modifiaient pas la structure de commande. De plus, des équations 8 et 11 nous obtenons :

$$M(Red) \leq 1 \quad (16)$$

Des équations 9 et 11 nous obtenons :

$$M(Yellow) \leq 1 \quad (17)$$

Des équations 10 et 11 nous obtenons :

$$M(Green) \leq 1 \quad (18)$$

Les lampes sont dans l'état "on" ou "off". Les places contiennent au plus un jeton et il n'y a donc pas de contradiction avec leur interprétation comme proposition logique vrai ou fausse. De plus des équations 11, 8 et 10 nous pouvons montrer formellement qu'il est impossible d'allumer simultanément les lampes rouges et vertes (elles ne peuvent être toutes les deux dans l'état "on" car les places *Red* et *Green* ne peuvent simultanément contenir un jeton. Nous avons en effet :

$$M(Red) + M(Green) = M(m_0) + M(m_1) + M(m_2) \leq 1 \quad (19)$$

5 Un modèle avec moins de places

Dans son article, Jörg Desel montre qu'il existe un autre modèle avec moins de place. Ayant moins de place, il pourrait être plus simple et donc meilleur. Ce modèle est représenté figure 7.

Si l'on énumère l'ensemble des marquages accessibles de ce réseau, on peut vérifier qu'il correspond bien aux spécifications du problème. Les transitions t_1 , t_2 , t_3 et t_4 sont franchies dans le bon ordre et les marquages successifs des places *Red*, *Yellow* et *Green* sont corrects. Toutefois, le rôle de la place *control*, nécessaire pour éviter qu'une infinité de jetons puisse être mis dans la place *Yellow*, par exemple, est difficile à expliquer à un non spécialiste. Cette place est une astuce de modélisation. Elle contient un jeton quand le feu n'est ni rouge, ni vert.

Ce modèle est difficile à construire, mais étant plus compact, présente-t-il des avantages par rapport au modèle précédent dont la construction peut être expliquée facilement ?

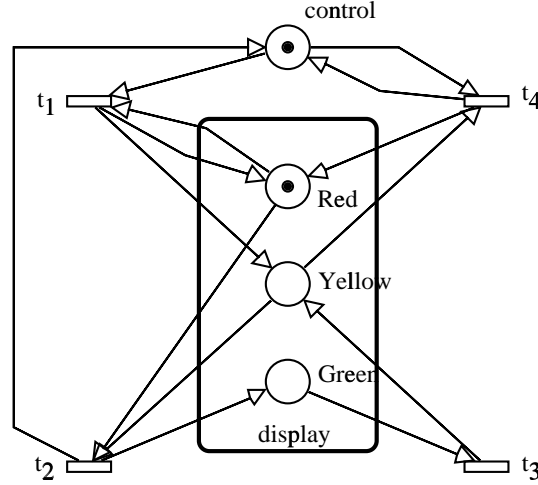


FIG. 7: Réseau de Petri minimal

La matrice d'incidence de ce réseau (7) est :

$$C = \begin{array}{cccc} & t_1 & t_2 & t_3 & t_4 \\ \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & -1 & 1 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} & & & & \begin{array}{l} control \\ Red \\ Yellow \\ Green \end{array} \end{array} \quad (20)$$

Le seul invariant de place que l'on peut déduire est :

$$M(control) + M(Red) + M(Yellow) + M(Green) = 2 \quad (21)$$

Il n'est donc pas possible de montrer par une approche algébrique simple que les places du réseau contiennent au plus un jeton et qu'elles peuvent donc être interprétées comme étant des conditions logiques. Il n'est non plus possible de montrer par cet invariant que les lampes rouges et vertes ne peuvent pas être allumées simultanément. Seule l'énumération des marquages accessibles permet de montrer ces propriétés. Bien entendu, le cas présent est un exemple tellement simple que l'énumération ne pose aucun problème, mais dans la pratique, les réseaux sont complexes et l'énumération des marquages accessibles conduit à une explosion combinatoire.

Une conjecture est que la **compréhensibilité d'un modèle fondé sur un réseau de Petri est assez liée au fait que ses propriétés significatives peuvent être montrées de façon algébrique ou non**. Un modèle clair et facilement vérifiable est un réseau de Petri telle que **l'ensemble des marquages effectivement accessibles** est le même que (ou est très proche de) **l'ensemble des solutions de l'équation fondamentale 5** (M est l'inconnue et \bar{s} un paramètre) ainsi que **l'ensemble des marquages M définis par un base d'invariant de place F** par :

$$F^T \cdot M = F^T M_0 \quad (22)$$

F est une base de l'espace vectorielle des solutions f de (f^t est le transposé de f) :

$$f^T \cdot C = 0 \quad (23)$$

Le réseau de Petri de la figure 7 contient deux boucles élémentaires (entre t_1 et Red et entre t_4 et $control$). Si on efface ces boucles, la matrice C est inchangée et donc la base F . Si maintenant on énumère les marquage accessibles de ce nouveau réseau, on s'aperçoit qu'il est possible de mettre deux jetons dans les places Red , $Yellow$ ou $Green$ et qu'il est également possible de mettre simultanément un jeton dans Red et dans $Green$. Donc il est clair que les propriétés essentielles ne peuvent pas être montrée par des invariants de places ou par l'équation fondamentale.

A part le fait d'avoir trois places en moins, le réseau de la figure 7 ne présente donc vraiment aucun avantage par rapport au réseau de la figure 6.

La dernière remarque est que la modélisation d'un seul feu de carrefour est en général le prélude à celle de l'ensemble des feux d'un carrefour avec leur relations de dépendance et des considération temporelles (temps minimal et maximal pour chaque étape d'un cycle). Pour cela les places de la partie commande m_0 , m_1 , m_2 et m_3 sont indispensables.

En conclusion, bien qu'il soit optimal vis-à-vis du nombre de places, le réseau de la figure 7 peut être considéré comme un moins bon modèle que celui de la figure 6.