



HAL
open science

Une approche basée sur les réseaux Pétri pour la vérification et la validation formelle des modèles de processus métier dans les entreprises congolaises

Blaise Fyama, Ruphin Nyami

► To cite this version:

Blaise Fyama, Ruphin Nyami. Une approche basée sur les réseaux Pétri pour la vérification et la validation formelle des modèles de processus métier dans les entreprises congolaises. 2018. hal-01864701

HAL Id: hal-01864701

<https://hal.science/hal-01864701v1>

Preprint submitted on 30 Aug 2018

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.

« Une approche basée sur les réseaux Pétri pour la vérification et la validation formelle des modèles de processus métier dans les entreprises congolaises »

Blaise FYAMA

Faculté des Sciences Informatiques, Université Liberté, 2179, Avenue du 30 juin, Lubumbashi, RD Congo

bfyama@gmail.com

Ruphin NYAMI

Faculté des Sciences Informatiques, Université Liberté, 2179, Avenue du 30 juin, Lubumbashi, RD Congo

ruffinnyami125@gmail.com

Abstract

Business Process Modeling (BPMN) notation is a standard used in the requirements specification phases. The plurality of BPMN formalism and the complexity of the system to be developed constitute a source of errors that requires formal verification and validation. The BPMN suffer from the lack of effective analytical approaches, and given the importance of business processes in companies, it is a real danger to let go of anomalies such as interlocking, liveliness and multiple repetition. Precisely, the audit serves to determine in advance whether a process model presents certain desirable behaviors. This article proposes a formal semantics of the BPMN elements defined in terms of mapping to Petri nets in order to apply the different existing analysis techniques.

Résumé

La notation de la modélisation des processus métier (BPMN) constitue un standard utilisé dans les phases de spécification de besoins. La pluralité du formalisme BPMN et la complexité du système à développer constituent une source d'erreurs qui nécessite une vérification et validation formelle. Le BPMN souffrent de l'absence d'approches effectives d'analyse, et vu l'importance des processus métiers dans les entreprises, c'est un vrai danger de laisser passer des anomalies telles que l'inter-blocage, la vivacité et la répétition multiple. Justement, la vérification sert à déterminer à l'avance si un modèle de processus présente certains comportements souhaitables. Cet article propose une sémantique formelle des éléments BPMN définis en termes de mappage aux réseaux de Pétri afin d'y appliquer les différentes techniques d'analyse existantes.

Mots-clés

Réseaux de pétri, BPMN, vérification, validation et transformation formelle, de modèle du métier, modèle de processus métier, Spécification de besoin, matrice d'incidences, diagramme d'activité UML.

Keywords

Petri Networks, BPMN, Verification, Validation and Formal Transformation, Business Model, Business Process Model, Requirement Specification, Outcome Matrix, UML Activity Diagram.

0. INTRODUCTION

Pendant ces dernières années, le BPM (Business Process Management) est devenue un standard dans la gestion de processus du métier, cette gestion donne à l'entreprise les moyens de gérer ses processus métiers de manière informatisée (modélisation, simulation, exécution et audit) et permet une optimisation, adaptation aux besoins en temps réel. La description des processus métiers avec le BPMN (Business Process Model and Notation) apportent une vision du métier réel, et constituent un excellent instrument de formalisation et d'analyse, dans la construction des systèmes. A ce titre, ils occupent une place centrale dans les systèmes d'information d'entreprise d'aujourd'hui.

Cependant les analystes des systèmes sont confrontés à des difficultés lors de l'utilisation du BPMN dans l'étude de processus métiers transversaux et complexes. Puisqu'une conception efficace d'un tel système rend nécessaire une vérification formelle mathématiquement précise à chaque phase de la modélisation.

Pour des buts d'analyse et de vérification plusieurs études ont été menées dans le contexte de l'intégration du modèle BPMN avec des modèles formelle ou encore dans le contexte de la transformation de modèle.

Cette étude sur «une approche basée sur les réseaux Pétri pour la gestion de processus métier » vise à répondre à ces interrogations en s'attelant sur les points suivants :

- La vérification, validation et transformation formelle des modèles de processus métier ;
- Présentation sommaire des réseaux de Pétri ;
- La gestion de processus métier ;
- Règles de transformations de Modèles de processus métier vers les réseaux de Pétri ;

1. La vérification, validation et transformation formelle de Processus métier.

Parmi plusieurs études existantes dans la littérature qui s'intéresse à la vérification des modèles de processus métier, on peut citer les travaux sur l'outil AToM³ [1] lequel outil, permet une transformation automatique des diagrammes de processus métier et d'activités UML vers les équivalents en réseaux de Pétri imbriqués. Cet outil constitue ainsi une base pour la vérification et l'analyse formelle des diagrammes d'activités UML. AToM³ [2] est un outil visuel pour la modélisation et la méta-modélisation multi formalismes. Pour la méta-modélisation, AToM³ supporte la modélisation et pour la transformation de modèles, AToM³ supporte la réécriture de graphes qui utilise les règles de grammaire de graphes pour guider visuellement la procédure de transformation. Les règles sont spécifiées par l'utilisateur et ordonnées selon des critères dépendants des caractéristiques du modèle à transformer. [3, p. 84]

Sans oublier l'outil vUML [4] offre également une vérification automatique des modèles UML. Le comportement des objets dans cet outil est spécifié à l'aide du diagramme d'état-transition ; il invoque le modèle checker SPIN [5], l'outil transforme automatiquement les différents modèles vers des spécifications en un langage dit PROMELA. Cette vérification est établie par l'outil sans intervention humaine.

Ahmad [6] propose l'utilisation de filet de Pétri pour la validation et la vérification des propriétés comportementales des modèles BPMN et également les réseaux de Pétri pour simuler et réaliser une analyse de performance pour le nouveau modèle.

Selon AISSAN [7] « les différentes approches de transformation peuvent être classées en deux catégories ». La première catégorie s'intéresse à la transformation des modèles UML vers des formalismes performants (favorisant une analyse mathématique) comme les réseaux de Pétri ou encore les algèbres de processus. La deuxième catégorie qui s'intéresse à la combinaison entre les deux notations c'est-à-dire construire des modèles performants à l'aide de notations BPMN. Parmi les approches qui favorisent l'intégration des modèles BPMN avec des formalismes formelles, on cite l'approche [8] qui le résultat des efforts de l'équipe UML, cette approche introduit l'utilisation du langage de spécification Z comme domaine sémantique des modèles UML. Pour augmenter la sémantique du diagramme d'activité [9] combine la logique temporaire avec ce dernier pour bénéficier de la logique du diagramme d'activité est habituellement décrite à l'aide d'OCL (Object Constraint Language) qui n'est pas trop adopté dans ce cas car il ne représente pas les aspects temporaires.

Meena [10] et son équipe de recherche mettent en avant l'exploitation de la dérivation des diagrammes d'activités modélisant le Workflow d'un système. Les auteurs définissent toute la sémantique des règles de dérivation établies qui sont simples et très efficaces. Les auteurs proposent un outil graphique pour la modélisation des diagrammes d'activités UML et ils développent leur propre outil de vérification des RdPs. Ils fournissent un fichier de sortie du réseau de Pétri résultant sous forme standard (PNML : Pétri Net Markup Language). On souligne aussi qu'ils ne tiennent pas compte des diagrammes d'activités partitionnés pourtant très répandus dans la modélisation de Workflow.

Ehrig [9] et son équipe de recherche expose un outil « TIGER » (Transformation based Generation of Modeling environnement). Cet outil a pour but essentiel de présenter une application graphique pour la dérivation de modèle graphique (source) vers d'autres modèles graphiques (destination). Les auteurs présentent un exemple particulier qui est celui de la dérivation des diagrammes d'activités vers les réseaux de Pétri. Après la dérivation du modèle, l'outil propose un moyen de vérification et de validation du modèle RdP. L'accent est mis beaucoup plus sur l'aspect visuel de l'outil ainsi que son éventuelle intégration avec d'autres outils ; c'est pourquoi l'on souligne l'absence des définitions des règles sémantiques.

L'étude menée par **ELMENSOURI** [3] présente une l'utilisation d'ECATNets qui sont l'union du concept des réseaux de Pétri et des types abstraits algébriques comme formalisme cible de la transformation de modèles BPMN vers les réseaux de pétri. Le passage du modèle des processus métiers vers le modèle réseaux de Petri est réalisé par la transformation de graphes à l'aide de l'outil ATOM3 [3, p. 69].

2. Présentation sommaire des Réseaux de Pétri.

Carl Adam Pétri est un mathématicien allemand qui a défini un outil mathématique très général permettant de décrire des relations existant entre des conditions et des évènements, de modéliser le comportement de systèmes dynamiques à évènements discrets.

- début des travaux 1960-1962 : ont donné lieu à de nombreuses recherches.
- 1972-1973, utilisation de cet outil pour la description d'automatismes logiques, ce qui a débouché sur le Grafset.

Il existe différents types de réseaux de Pétri : temporisés, hiérarchisés, interprètes, stochastiques, colores, continus et hybrides [11].

Il est caractérisé par son aspect graphique qui nous aide à comprendre facilement le système modélisé. Il permet de simuler les activités dynamiques, concurrentes et parallèles [11].

Bref, il ne suffit pas de modéliser un système, mais plutôt comment réussir à appréhender le comportement des systèmes technologiques de plus en plus complexes, afin de les concevoir, de les réaliser et/ou de les commander à partir d'un cahier des charges.

2.1. Définition Graphique

Un réseau de Pétri (*RdP*) est un graphe biparti orienté valué. Il a deux types de nœuds [12].

1. *les places* : notées graphiquement par des cercles. Chaque place contient un nombre entier (positif ou nul) de marques (ou jetons). Ces derniers sont représentés par des points noirs.
2. *les transitions* : notées graphiquement par un rectangle ou une barre. Une transition qui n'a pas de place en entrée est appelée transition source et une transition qui n'a pas de place en sortie est appelée transition puits.

Les places et les transitions sont reliées par des arcs orientés où :

- *Un arc* relie, soit une place à une transition, soit une transition à une place mais jamais une place à une place ou une transition à une transition.
- Chaque arc est étiqueté par une valeur (ou un poids), qui est un nombre entier positif. L'arc ayant k poids peut être interprété comme un ensemble de k arcs parallèles. Un arc qui n'a pas d'étiquette est un arc dont le poids est égal à 1.

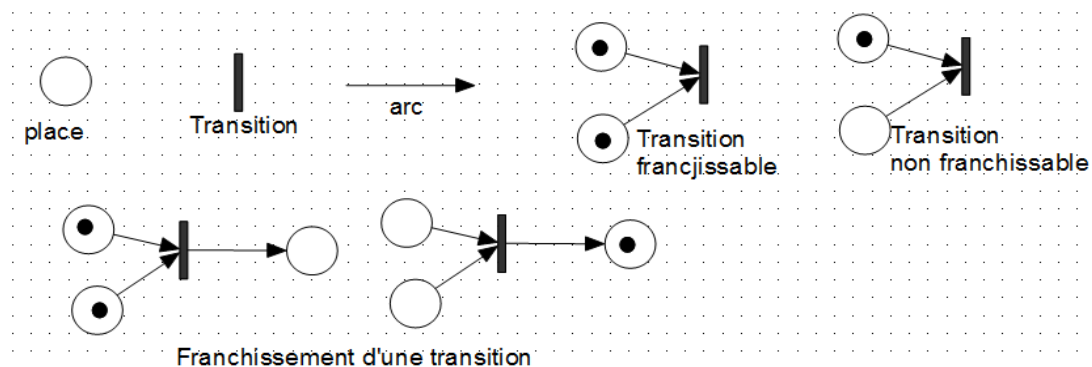


Figure 1 : Symboles et principes de base des Réseaux de Pétri

2.2. Définition Formelle

Un réseau de Pétri est un graphe orienté biparti défini par un quadruplet $\mathbf{R} = (\mathbf{P}, \mathbf{T}, \text{Entrée}, \text{Sortie})$ [12] où :

- $\mathbf{P} = \{p_1, p_2, p_3, \dots, p_n\}$ est un ensemble fini de places \mathbf{P}
- $\mathbf{T} = \{t_1, t_2, t_3, \dots, t_n\}$ est un ensemble fini de transitions \mathbf{T}
- **Entrée** (ou Pré) est une application, **Entrée** : $\mathbf{P} \times \mathbf{T} \rightarrow \mathbf{N}$, appelée application d'incidence avant. Notée Pré (p,t) ou Entrée(p,t) ou encore $I(p,t)$, contient la valeur entière « n » associée à l'arc allant de « p » à « t ».
- **Sortie** notée Post (p, t) ou Sortie (p, t) ou encore $O(p, t)$ contient la valeur entière « n » associée à l'arc allant de « t » à « p ».

2.3. Marquage d'un Réseau de Pétri

Un marquage est dénoté par un vecteur du nombre de jetons dans chaque place : la $i^{\text{ème}}$ composante correspond au nombre de jetons dans la $i^{\text{ème}}$ place [11].

Le **marquage** d'un RdP est précisé par la présence à l'intérieur des places d'un nombre fini (positif ou nul), de marques ou de jetons. Une place est donc vide ou marquée. Lorsque la place représente une condition logique, la présence d'un jeton indique que cette condition est vraie; fausse dans le cas contraire. Une place donc peut représenter une ressource du système, elle peut contenir plusieurs jetons. Une place contient un nombre ≥ 0 de jetons.

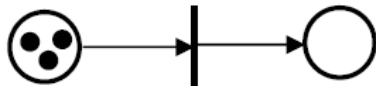


Figure 2 : Marquage d'un réseau de pétri

Le **marquage initial**, M_0 , d'un RdP correspond à la distribution initiale des jetons dans chacune des places du RdP, qui précise l'état initial du système. Dans ce cas, on parle du **Rdp Marqué** par opposition à un **Rdp non marqué**, c'est-à-dire pour lequel le marquage initial n'est pas précisé.

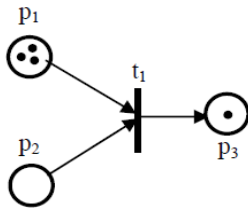
2.4. Dynamique des Réseaux de Petri

2.4.1. Validation d'une transition

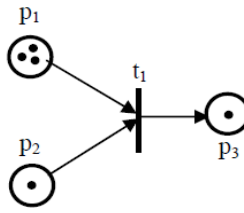
L'évolution d'un Réseau de Pétri correspond à l'évolution de son marquage au fil du temps (évolution de l'état du système) : il se traduit par un déplacement des jetons pour une transition t de l'ensemble des places d'entrée vers l'ensemble des places de sortie de cette transition. Ce déplacement s'effectue par le franchissement de la transition t selon des règles de franchissement [11].

a) Transition validée

On dit qu'une transition est validée (on dit aussi sensibilisée ou franchissable) si toutes les places en entrée de celle-ci possèdent au moins une marque. Une transition source est par définition toujours validée [12].



La transition t_1 n'est pas validée
car $M(p_2)=0$



La transition t_1 est validée car
 $M(p_1) > 0$ et $M(p_2) > 0$

Figure 3 : Franchissement d'une transition

b) Franchissement d'une transition

Si la transition est validée, on peut effectuer le franchissement de cette transition : on dit alors que la transition est franchissable [12].

Le franchissement consiste à :

- retirer $W(p, t)$ jetons dans chacune des places en entrée p de la transition t .
- ajouter $W(t, p)$ jetons à chacune des places en sortie p de la transition t .

La règle de franchissement est illustrée par la figure

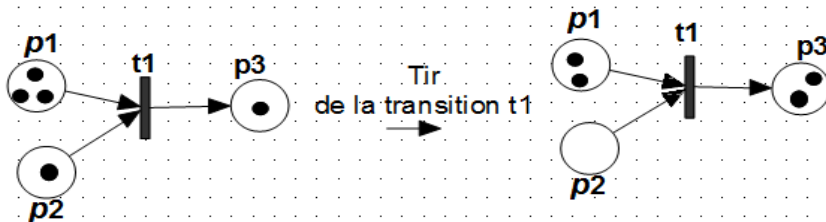


Figure 4 : Franchissement d'une transition

2.5. Modélisation Avec les Réseaux de Petri

Les réseaux de Pétri ont été conçus et utilisés principalement pour la modélisation. Plusieurs systèmes peuvent être modélisés par les réseaux de Pétri, ces derniers peuvent être de natures très diverses : matériel informatique, logiciels informatiques, les systèmes physiques, les systèmes sociaux, etc [13].

2.4.1. Propriétés des réseaux de Petri

a) Conflits et parallélisme :

Un réseau de Pétri est dit sans conflit si et seulement si toute place a au plus une transition de sortie. Un conflit (structurel) correspond à l'existence d'une place P_i qui a au moins deux transitions de sortie T_j, T_k .

Un RdP avec conflit est un réseau qui possède donc une place avec au moins deux transitions de sorties.

Cette situation du conflit correspond à la concurrence à la consommation des jetons à une place.

b) Conflit relatif au marquage ou transitions en conflit effectif

Dans un RdP, deux transitions T_i et T_j sont en conflit effectif pour un marquage M si et seulement si : $M \geq \text{Pre}(\cdot; T_i)$, $M \geq \text{Pre}(\cdot; T_j)$ et M n'est pas supérieur ou égal $\text{Pre}(\cdot; T_i) + \text{Pre}(\cdot; T_j)$.

c) Synchronisation

Les réseaux de Petri ont été utilisés pour modéliser une variété de mécanismes de synchronisation, y compris les problèmes de l'exclusion Mutuelle, producteur/consommateur, lecteurs/écrivains... etc [12].

d) RdP pur

Un RdP pur est un réseau dans lequel il n'existe pas de transition ayant une place d'entrée qui soit à la fois place de sortie de cette transition (boucle élémentaire). Dans le cas contraire, on parle de RdP impur.

e) Accessibilité

L'accessibilité est une propriété fondamentale pour étudier les propriétés dynamiques du système. Un marquage M_n est dit accessible à partir de M_0 , s'il existe une séquence de franchissements permettant de transformer M_0 à M_n .

f) Transition Vivante

Une transition T_j est quasi vivante pour un marquage initial M_0 s'il existe une séquence de franchissements à partir de M_0 contenant T_j . Il s'en suit qu'un RdP est quasi vivant pour un marquage initial M^0 si toutes ses transitions sont quasi vivantes pour ce marquage initial. Conséquemment, une transition qui n'est pas quasi-vivante est inutile!

g) Transition source

Une transition source est une transition qui ne comporte aucune place d'entrée

h) Transition puits

Une transition puits est une transition qui ne comporte aucune place de sortie

i) Réinitialisable et État d'accueil

Un réseau de Pétri s'appelle réinitialisable si pour chaque marquage M dans $R(M_0)$, M_0 est accessible à partir de M .

Un RdP possède un état d'accueil M_a pour un marquage initial M_0 si pour tout marquage accessible M_i il existe une séquence de tirs S telle que $M_i[S > M_a]$. Il s'en suit qu'un RdP est réinitialisable (ou réversible) pour un marquage initial M_0 si M_0 est un état d'accueil.

j) Blocage

Un marquage M d'un réseau (N, M_0) est appelé marquage "*puits*" si aucune transition n'est franchissable depuis M . Un réseau est dit sans blocage si tout marquage accessible depuis M_0 n'est pas un marquage « puits » [12].

La propriété de vivacité examine si une partie ou l'ensemble du réseau peut ou non évoluer. Une transition T_j est vivante pour un marquage initial M_0 si pour tout marquage accessible M_k , il existe une séquence de franchissements à partir de M_k contenant T_j : $M_0 \rightarrow M_k$

Un RdP marqué est vivant pour un marquage initial M_0 si toutes ses transitions sont vivantes pour ce marquage initial.

2.4.2. Les Réseaux de Pétri de Haut Niveau

Pour l'utilisation des réseaux de Pétri dans la modélisation des systèmes réels, plusieurs auteurs ont trouvé qu'il est convenable d'étendre le formalisme de réseau de Pétri pour compacter la représentation de modèle ou pour étendre le pouvoir de modélisation du formalisme de réseau de Pétri. Ce qui a donné naissance aux réseaux de Pétri de haut niveau.

2.4.2.1. Réseau de Petri Coloré

Le développement des réseaux de Petri coloré (CP-nets ou CPN) a été motivé par le désir de développer un langage de modélisation bien-fondé théoriquement et assez versatile en même temps. Ils sont utilisés pour les systèmes de taille et de complexité qu'on retrouve dans les projets industriels.

Contrairement aux réseaux de Pétri ordinaires, chaque place a un type associé déterminant le type de données que la place peut contenir et chaque jeton a une valeur de données dont le type est le même que le type associé à la place.

Les actions d'un CP-net sont représentées par des transitions (qui sont sous forme de rectangle). Pour que la transition soit franchissable, on doit avoir suffisamment de jetons dans les places d'entrée.

2.4.2.2. Réseau de Pétri Objet

Les réseaux de Pétri Objet (*OPN*) étendent le formalisme des réseaux de Pétri colorés avec une intégration complète des propriétés orientées objet y compris l'héritage, le polymorphisme et la liaison dynamique. L'orientation objet fournit des primitives de structuration puissante permettant la modélisation des systèmes complexes [14].

3. Gestion de processus métier

Le Business Process Management Initiative (BPMI), organisme qui anime un courant de standardisation dans le domaine du management par les processus métiers, a mis au point une norme de notation de modélisation des processus métiers ; la « Business Process Modeling Notation (BPMN) ». La spécification BPMN 1.0 a été libérée au public en Mai 2004 [3].

En effet, face à la mondialisation et aux menaces ou aux opportunités environnementales, on s'est rendu compte que le « **business process** » est le paradigme le plus important dans la gestion [15]. L'idée de l'organisation processus devient plus forte. L'option « processus » est devenue un besoin obligatoire, ce qui veut dire que pour être conforme avec la nouvelle philosophie des affaires, l'organisation doit être conçue pour fournir le flot vertical et horizontal des informations nécessaires à la réalisation des objectifs globaux de l'organisation.

L'objectif principal de l'effort de BPMN était de fournir une notation qui soit facilement compréhensible par tous les utilisateurs métiers, depuis les analystes qui créent les premières ébauches des processus, jusqu'aux développeurs techniques responsables de l'implémentation de la technologie qui va s'acquitter de ces processus, et, enfin, les gens d'affaires qui vont gérer et contrôler ces processus [16].

La BPMN sera également soutenue par un modèle interne qui permettra la génération d'exécutables sous le format BPEL (Business Process Executable Langage). Ainsi, BPMN veut créer un pont standard pour l'écart entre la conception et la mise en œuvre des processus métiers [17].

3.1. Définition de processus métier (Business Process)

Un processus est un ensemble partiellement ordonné d'**étapes exécutées** en vue de réaliser au moins un objectif [15]. Une définition plus complète dit qu'« Un processus est

défini comme un enchaînement partiellement ordonné d'exécution d'**activités** qui, à l'aide de **moyens techniques et humains**, transforme des éléments d'**entrée** en éléments de **sortie** en vue de réaliser un **objectif** dans le cadre d'une **stratégie** donnée » [15]. Selon HADDAD [18, p. 15], un Workflow « est une séquence des tâches qui réalisent un processus métier, le flot de données qui supporte ces tâches, et les mécanismes qui permettent de mesurer, suivre et contrôler ces tâches ».

3.2. Correspondance entre éléments BPMN et diagramme d'activité UML

Concrètement, l'UML est le standard utilisé dans la modélisation de la plupart des aspects logiciel et métier [19] alors que le BPMN est le standard utilisé uniquement dans la modélisation des processus métier. Tandis que l'UML 2.5 en sa version bêta est constitué de 14 diagrammes, BPMN n'en propose qu'un seul, très proche du diagramme d'activités.

Le BPMN est facilement compréhensible par :

- l'analyste métier qui modélise les processus conceptuellement ;
- les développeurs en charge de rendre exécutable le processus modélisé ;
- les utilisateurs finaux qui utilisent et suivent la réalisation des processus.

Les principaux éléments constitutifs du processus sont les suivants :

Tableau 1 : Correspondance entre éléments BPMN et diagramme d'activité UML [19]

Terme	Activité UML	Activité BPMN	Explication
Activité/ Processus			Représente un processus et contient (les actions et les tâches)
Action/ Tâche			Unité d'exécution ou une tâche prise en charge par une partition
Partition/Lane			Représente une entité qui a la charge de réalisation d'une action. Il peut s'agir d'un acteur, d'une structure d'entreprise ou d'organisation.
Object Node/ Data Object			Représente les informations échangées entre les actions. Il est possible d'indiquer l'état d'un objet entre crochets.
Transition			Matérialise le passage d'une action à une autre.
Décision			Permet de définir un branchement conditionnel.
Début de processus/			Définit le démarrage d'un processus.
Processus end			Définit le fin d'un processus.

4. La transformation du diagramme de Processus métier en Réseaux de Pétri Coloré

Dans cette partie, nous expliquerons les règles qui permettent de définir comment les concepts du diagramme de processus BPMN peuvent être transformés en des éléments des réseaux de Pétri coloré adéquat. Nous détaillerons également des éléments BPMN qui ne peuvent pas être mappés à des éléments des réseaux de Pétri colorés correspondants. Ces règles peuvent être utilisées pour la traduction automatique des diagrammes de processus métier vers des réseaux de Pétri. En fin nous expliquerons par un cas concret cette transformation en vérifiant certaines propriétés dans la matrice d'incidence.

Pour construire les règles de transformation, nous allons basés sur les transformations décrites dans [2] [20].

4.1. Diagramme de Processus métier et les réseaux de Pétri

Le tableau 2 ci-dessous explique de transformation des éléments BPMN vers des réseaux de Pétri.

Tableau 2 : Diagramme de processus métier et Leurs correspondants dans le réseau de Pétri

Eléments du WorkFlow	Correspondance en Réseau de Pétri
Processus	Réseau de Pétri
Ruelle, pool	Place
Activité (tâche)	Transition
Sous Processus	Réseau de Pétri
Tache à instances multiples	Transition
Evénement début	Place
Evénement de Fin	Place
Evénement intermédiaire	Transition
Passerelle non précisé	Place
Passerelle exclusif	Place
Passerelle Parallèle	Place
Passerelle inclusif	Place
Passerelle Complexe	Place
Flux séquentiel	Arc
Flux de Message	Arc
Groupe	Réseaux de pétri
Entrée de données	Jeton
Production de Données	Jeton
Stockage de données	Jeton
Collection de données	jeton

4.2. Règles de Transformation

4.2.1. Activité

Une activité BPMN est une tâche qui peut être réalisée par un humain, un système ou un sous-processus. Chaque activité a un début et une fin et donc une activité ne peut débuter que si la tâche précédente est terminée. Une activité prend un ensemble d'entrées et le produit en ensemble de sortie. Les sorties d'une activité peuvent être fournies comme entrées d'autres activités à l'aide d'un flux séquentiel. HADDAD [18] propose que les activités soient transformées en transition avec une place en entrée et place de sortie dans les réseaux de Pétri.

Dans BPMN, une activité peut avoir des attributs spécifiant son comportement supplémentaire, tels que la boucle et les instances multiples parallèles. Construction de constructions en boucle d'activité les boucles « while-do » et « do-until » [16].

4.2.2. Événements (events)

Les événements servent à identifier un état particulier dans le processus. Ils n'effectuent aucune tâche. Dans BPMN 1.0 il existe 3 types d'événements : **début, intermédiaire et fin**. Comme l'événement change l'état du système dans un diagramme de processus, la transition fait pareil dans le réseau de pétri coloré [20, p. 7].

a) Les événements de début

Marque la première étape d'un processus. Il indique le point de départ et ne symbolise aucune tâche. Il peut recevoir un événement de départ pour déclencher un processus. Dans le cas où il ne reçoit aucun élément de départ. **Dumas** [20, p. 7] et son équipe proposent qu'un événement de début soit mappé une transition silencieuse avec une place en entrée et une place de sortie pour signaler quand commence un processus.

b) L'événement intermédiaire

Il peut être utilisé de 2 manières : au milieu d'un flux pour signaler l'attente d'un type d'événement ou rattaché à une tâche pour faire apparaître une exception au traitement de la tâche. Un événement intermédiaire sans aucun type particulier (cercle vide), il signale qu'une étape est franchie. BPMN 2.0 dispose plusieurs sorties d'événements intermédiaires avec différents symboles (de message, minuterie, escalade, conditionnel, lien, erreur, annulation, compensation, signal, multiple, parallèle multiple) [15]. Un événement intermédiaire est mappé en transition avec une place en entrée et une place en sortie [20, p. 7].

c) Événement de Fin

L'événement de sortie est représenté par un cercle épais. Il indique la fin du processus et ne symbolise aucune tâche. Il peut envoyer un événement de sortie. Il peut être transformé en une place ou en une transition [20, p. 7]. Tout comme l'événement de début, l'événement de fin est transformé en une transition silencieuse avec une place en entrée et une place en sortie pour signaler la fin du processus.

4.2.3. Branchements (gateways)

Il sert à représenter la condition de routage entre le(s) flux en entrée et le(s) flux en sortie. Le branchement n'est pas une tâche et n'effectue aucune action [16]. Lorsque le losange est vide, chaque sortie est une alternative et il n'y a pas de différenciation entre les sorties. Le losange vide est utilisé lorsque le niveau d'abstraction du modèle est élevé et que

l'on ne désire pas compliquer une vue ou bien lorsque la règle de traitement n'est pas connue. Il est donc possible de définir un comportement de traitement précis.

Les Branchements, à l'exception des branchements de décisions basées sur des événements, sont mappés sur de petits module Pétri-net avec transition silencieuses capturant leur comportement de routage [20].

Les différents branchements possibles dans BPMN 2.0 sont :

- **Symbole exclusif** : Branchement qui évalue l'état du processus métier et, en fonction des cas, sépare le flux en un ou plusieurs chemins s'excluant mutuellement.
- **Symbole basé sur un événement**

Branchement semblable à un branchement de type **Exclusif**, tous deux impliquent un chemin dans le flux. Toutefois, dans le cas d'un branchement dépendant d'un événement, vous évaluez quel événement s'est produit, et non quelle condition a été remplie. Ce dernier sera transformé en une transition avec une place en entrée et en sortie dans le réseau de Pétri.

- **Symbole parallèle**

Ce type de branchement se distingue des autres en ce qu'il ne dépend pas de conditions ou d'événements. Au lieu de cela, les branchements parallèles sont utilisés pour représenter deux tâches simultanées dans un flux métier. Dans ce cas il est transformé en une transition avec une place en entrée et n places de sortie.

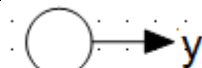
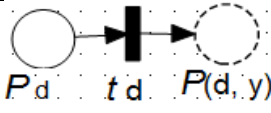

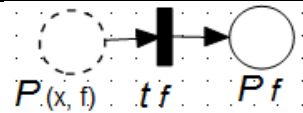

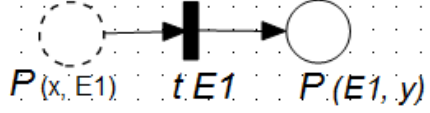
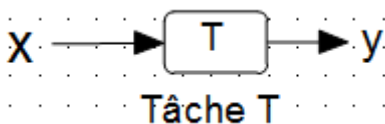
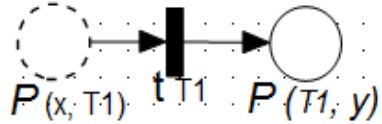
- **Symbole inclusif**

Décompose le processus en un ou plusieurs autres processus. Ce dernier est transformé en une transition avec deux places en entrée une place de sortie dans le réseau de Pétri.

- **Exclusif basé sur un événement**

Démarre un nouveau processus à chaque occurrence d'un événement ultérieur. N'effectuant aucune tâche, ce dernier a comme correspondant, la transition dans le réseau de Pétri.

Tableau 3 : Mappage des objets BPMN en Réseaux de Pétri [20]

Objet BPMN	Objet Réseaux de Pétri	Explications
 Début d	 $P_d \quad t_d \quad P(d, y)$	L'événement de début avec une transition silencieuse
 Fin f	 $P(x, f) \quad t_f \quad P_f$	L'événement de fin avec une transition silencieuse
 Message E	 $P(x, E1) \quad t_{E1} \quad P(E1, y)$	Evénement intermédiaire de message
 Tâche T	 $P(x, T1) \quad t_{T1} \quad P(T1, y)$	Une tâche BPMN

<p>Fourche F1</p>		Branchement parallèle (passerelle)
<p>Jointure J1</p>		Branchement de jointure ou synchronisation (passerelle) transformé en transition en RdPC
<p>Decision D1</p>		Branchement exclusif (X-OR) dont chaque chemin devient une transition en RdPC
<p>Fusion Fu1</p>		Branchement de fusion (avec deux entrées concurrentes) dont l'arrivée de l'une déclenche le processus.
<p>Evénement basé sur une décision ED1</p> <p>Message E1</p> <p>T1</p> <p>Tâche d'envoi T1</p>		Un événement basé sur un événement (inclusif) exécutant une tâche abstraite avec envoi de message.

Légende : x , x_1 ou x_2 représentent les entrées d'objet et y , y_1 ou y_2 représentent les objets en sorties.

4.2.4. Les objets de connexion

Les objets de connexion sont des lignes qui relient des objets de flux BPMN. Il en existe trois types différents : les flux séquentiels, les flux de message et les associations. Les objets de connexion deviennent des jetons dans le réseau de Pétri coloré à l'exception des flux de message qui nécessitent une analyse appropriée.

➤ Les flux de message

Ahmad propose « qu'un flux de messages est utilisé pour montrer la transmission de messages entre deux participants via des actions de communication telles que la tâche d'envoi, la tâche de réception ou le message un événement » [6]. De façon abstraite, il peut être mappé à un endroit avec un arc entrant de la transition modélisant l'action d'envoi et un arc sortant vers la transition modéliser l'action de réception. Un cas particulier pour le mappage est la cartographie d'un flux de messages vers un événement de départ. Un tel flux de messages représente que le processus est instancié chaque fois qu'un message est reçu. Dans ce cas, la transition représente l'envoi du message à un flux de celui-ci à l'endroit qui représente l'événement de démarrage, de sorte que le processus peut être démarré chaque fois qu'un message est reçu. La figure 5 montre les possibilités de mappage des flux de messages :

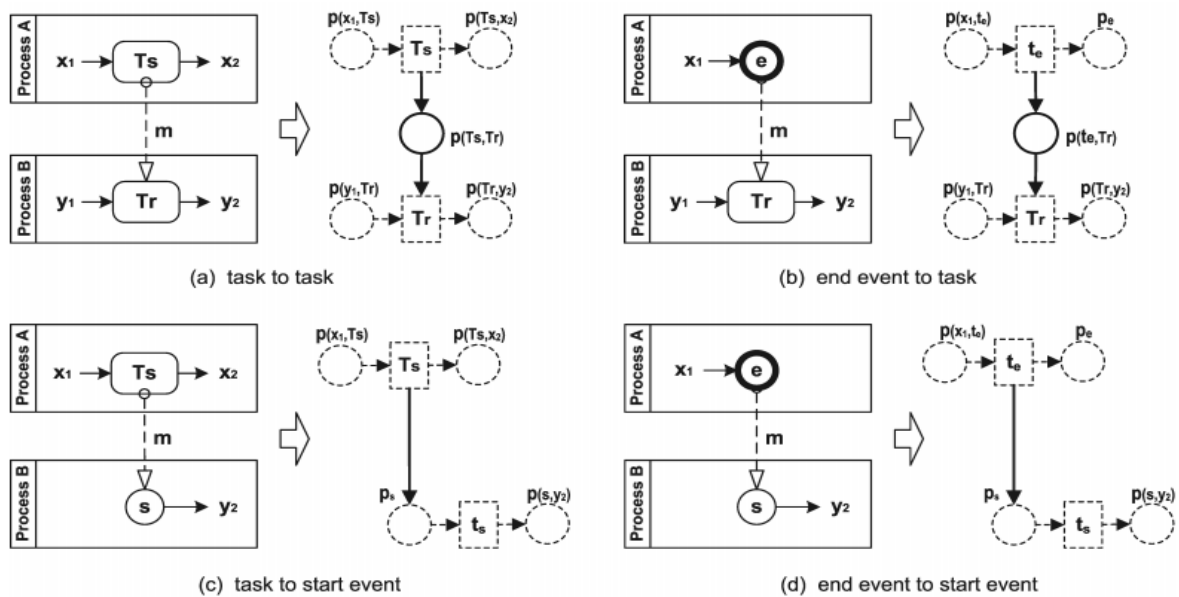


Figure 5 : Mappage des flux de message [6]

4.2.5. Les éléments d'organisation

a) Pool ou piscine

Utilisé pour contenir un processus unique et complet. Le principe exige dit qu'un workflow ne peut pas sortir d'un pool [10] : il convient de transférer les actions d'une piscine à une autre en utilisant les événements. Il n'existe pas une notation dans les réseaux de Pétri.

b) Ligne d'eau (swimlane)

Utilisée pour identifier qui fait quoi dans un pool [16]. Un flux séquentiel peut franchir les lignes d'eau comme si elles n'existaient pas. Elles ont une fonction purement organisationnelle. La ligne d'eau n'a pas de correspondance dans le réseau de Pétri. Cependant B. Mouna [8] propose une transformation vers un jeton de ressource place.

c) Groupe

Utilisé pour réunir des éléments graphiques dans un workflow. Il est sans conséquence pour les flux séquentiels. Le groupe sera transformé en un module du réseau de Pétri.

4.2.6. Consignes exceptionnelles de mappage de nœuds BPMN

Comme nous l'avons souligné dans la section 2, les réseaux de Pétri est un graphe biparti dont un nœud n'est pas précédé par un autre nœud du même type. Dans le diagramme de processus cette exigence n'est pas respectée.

- Si le nœud source dans un workflow est un événement de début et le nœud cible est une activité, alors remplacé par un arc dans le RdPC ;
- Le nœud source dans un workflow est un événement de début et le nœud cible est un branchement exclusif, inclusif, complexe, non défini, alors remplacé par un arc, place et transition dans le RdPC ;
- Si le nœud source dans un workflow est un nœud de tâche ou d'action et le nœud cible est un nœud d'action, ou un branchement de fusion (passerelle parallèle), ou nœud de fusion (branchement inclusif), alors remplacer par un arc et une place,
- Si le nœud source dans un workflow est un nœud de tâche et le nœud cible est un nœud de décision (branchement exclusif) alors remplacer par un arc et place et arc.
- Si le nœud source dans un workflow est un nœud fourche (branchement parallèle) ou un nœud de jointure (branchement inclusif) et le nœud cible est un nœud de tâche ou de jointure (branchement inclusif) alors remplacer par un arc et place;
- Si le nœud source dans un workflow est un nœud fourche (branchement parallèle) ou jointure et le nœud cible est nœud de décision, nœud de fusion ou nœud final alors remplacer par un arc ;
- Si le nœud source dans un workflow est un nœud de décision ou nœud de fusion d'objet et le nœud cible est un nœud d'action (tâche), nœud de fourche ou nœud de jointure alors remplacer par un arc ;
- Si le nœud source dans un workflow est un nœud de décision ou nœud de fusion ou nœud d'objet et le nœud cible est un nœud de décision ou nœud de fusion ou nœud finale alors remplacer par un arc.

4.2.7. Etude de cas

Dans cet exemple nous allons présenter une modélisation par diagramme de Processus de l'équation de deuxième degré. En mathématique, une équation du second degré, ou une équation quadratique, est une équation équivalente à : $\mathbf{ax}^2 + \mathbf{bx} + \mathbf{c} = \mathbf{0}$.

Où les lettres a, b et c désignent des nombres, avec **a** différent de 0 et x l'inconnu. Le terme second degré provient du fait que le polynôme définissant l'équation est de second degré.

Pour pouvoir résoudre une telle équation, il faut tout d'abord calculer le discriminant Δ . Pour le calculer, c'est facile, il suffit d'appliquer cette formule : $\Delta = \mathbf{b}^2 - \mathbf{4ac}$ On le calcule. Ensuite, selon le résultat, on va pouvoir connaître le nombre de solutions qu'il y a, et les trouver s'il y en a.

Si $\Delta < 0$, rien de plus simple : il n'y a pas de solution.

Si $\Delta = 0$, il y a une seule solution à l'équation : c'est $x = -b / (2a)$

Si $\Delta > 0$ il y a deux solutions qui sont $x_1 = (-b - \sqrt{\Delta}) / (2a)$ et $x_2 = (-b + \sqrt{\Delta}) / (2a)$

Une telle équation admet au maximum deux solutions. Suivant les valeurs des coefficients a, b et c et l'ensemble de nombres dans lesquels sont recherchées ces solutions,

elle peut en avoir une seule, double ou aucune. Dans le cas où les solutions recherchées sont des nombres complexes, il existe toujours deux solutions, éventuellement confondues.

Considérant que la solution de cette équation est connue de tous, nous passons directement à la notation BPMN pour représenter informatiquement cette équation, donc la modélisation dans le diagramme de processus comme le démontre la figure 6 suivante :

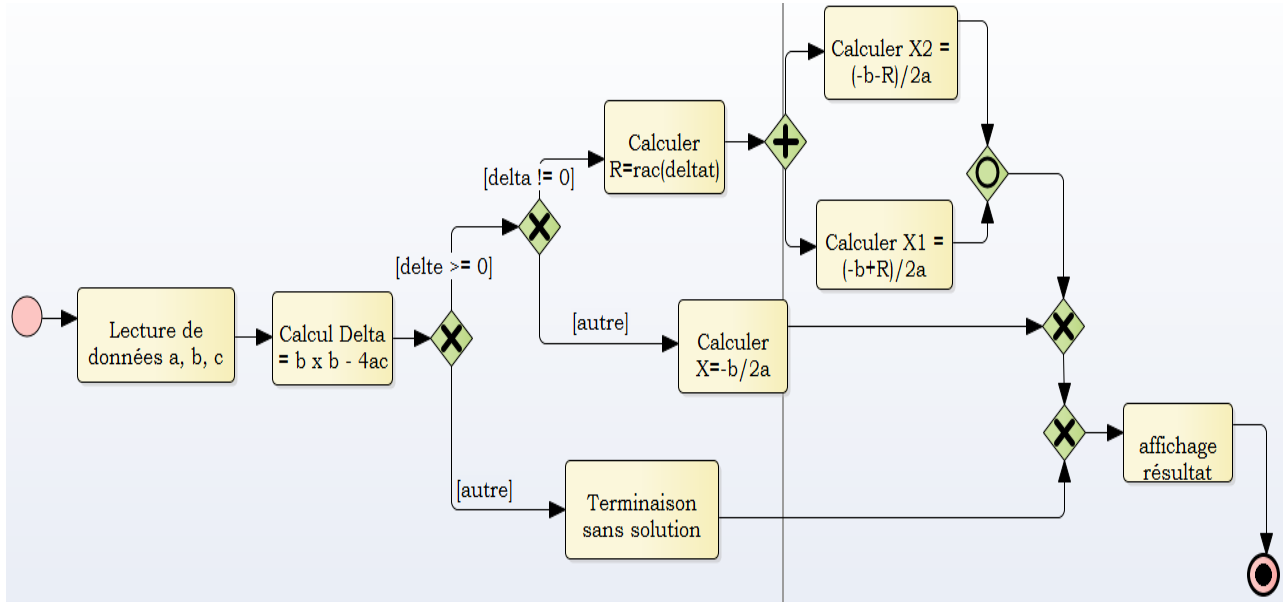


Figure 6: Exemple du Diagramme de Processus

Donc la transformation de ce digramme de Processus en suivant les règles de transformation qu'on a présenté dans la section précédente est présentée dans la figure 7 suivante :

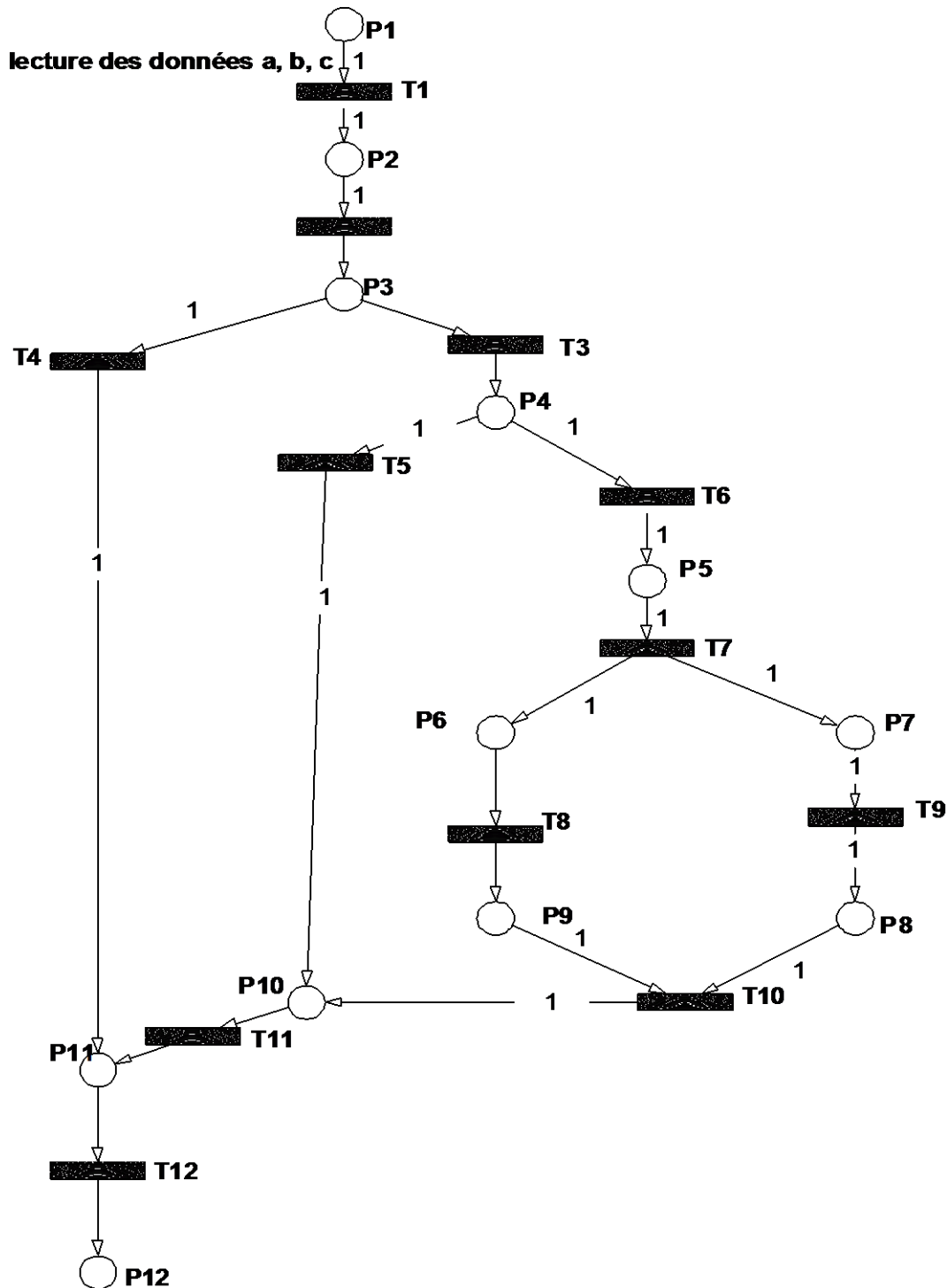


Figure 7 : Résultat de Transformation du Diagramme de Processus

4.2.8. Matrices incidences

La présentation faite des Réseaux de Pétri est une représentation **graphique**, alors qu'il est possible d'en faire une représentation mathématique grâce à l'algèbre linéaire [12]. Cette notation permet de représenter un RdP de manière plus formelle mais moins lisible que la représentation graphique [21]. Ainsi, Pré et Post sont représentés par des matrices à q lignes (nombre de places), n colonnes (nombre de transition) :

➤ **Matrice d'Incidences**

$W (T_i, P_i)$ si $P_i \in S (T_i)$

$W (P_i, T_j)$ si $P_i \in E (T_j)$

0 sinon

Tableau 4 : Matrice d'incidences

Matrice d'incidence avant (Pré) E												Matrice d'incidence arrière (Post) S												Matrice d'Incidence W = S-E														
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
p_1	1	0	0	0	0	0	0	0	0	0	0	0	p_1	0	0	0	0	0	0	0	0	0	0	0	0	p_1	-1	0	0	0	0	0	0	0	0	0	0	0
p_2	0	1	0	0	0	0	0	0	0	0	0	0	p_2	1	0	0	0	0	0	0	0	0	0	0	0	p_2	1	-1	0	0	0	0	0	0	0	0	0	0
p_3	0	0	1	1	0	0	0	0	0	0	0	0	p_3	0	1	0	0	0	0	0	0	0	0	0	0	p_3	0	1	-1	-1	0	0	0	0	0	0	0	0
p_4	0	0	0	0	1	1	0	0	0	0	0	0	p_4	0	0	1	0	0	0	0	0	0	0	0	0	p_4	0	0	1	0	-1	-1	0	0	0	0	0	0
p_5	0	0	0	0	0	0	1	0	0	0	0	0	p_5	0	0	0	0	0	1	0	0	0	0	0	0	p_5	0	0	0	0	0	1	-1	0	0	0	0	0
p_6	0	0	0	0	0	0	0	1	0	0	0	0	p_6	0	0	0	0	0	0	1	0	0	0	0	0	p_6	0	0	0	0	0	0	1	-1	0	0	0	0
p_7	0	0	0	0	0	0	0	0	1	0	0	0	p_7	0	0	0	0	0	0	1	0	0	0	0	0	p_7	0	0	0	0	0	0	1	0	-1	0	0	0
p_8	0	0	0	0	0	0	0	0	0	1	0	0	p_8	0	0	0	0	0	0	0	1	0	0	0	0	p_8	0	0	0	0	0	0	0	0	1	-1	0	0
p_9	0	0	0	0	0	0	0	0	0	1	0	0	p_9	0	0	0	0	0	0	0	1	0	0	0	0	p_9	0	0	0	0	0	0	0	1	0	-1	0	0
p_{10}	0	0	0	0	0	0	0	0	0	0	1	0	p_{10}	0	0	0	0	1	0	0	0	0	1	0	0	p_{10}	0	0	0	0	1	0	0	0	0	1	-1	0
p_{11}	0	0	0	0	0	0	0	0	0	0	0	1	p_{11}	0	0	0	1	0	0	0	0	0	1	0	0	p_{11}	0	0	0	1	0	0	0	0	0	0	1	-1
p_{12}	0	0	0	0	0	0	0	0	0	0	0	0	p_{12}	0	0	0	0	0	0	0	0	0	1	0	0	p_{12}	0	0	0	0	0	0	0	0	0	0	1	0

4.2.9. Interprétation

1°) Dans l'exemple précédant (figure 5), T_3 et T_4 sont en conflit structurel potentiel pour le partage des jetons de la place P_3 . Quand la place P_3 contient un jeton, les transitions T_3 et T_4 sont franchissables. Seule une des deux transitions peut être franchie : il est nécessaire de prendre une décision pour savoir laquelle des deux le sera effectivement. Ce conflit structurel doit être arbitré par une règle de priorité quelconque lorsque le conflit est effectif, c'est-à-dire lorsque les transitions aval en compétition pourraient être activées.

2°) Le franchissement de la transition T_7 met un jeton dans la place P_6 (ce qui marque le déclenchement du processus de calcul X_1) et un jeton dans la place P_7 (ce qui marque le déclenchement du processus $2 X_2$). Cette remarque traduit la notion de synchronisation.

3°) Nous remarquons que notre réseau est pur car il n'existe pas de transition ayant une place d'entrée qui soit à la fois place de sortie de cette transition (boucle élémentaire). Il est alors possible de reconstruire le RdP à partir de sa matrice d'incidence.

4°) Nous remarquons que ce réseau démontre que le réseau est vivant car il est sans blocages.

4.3. Critique de la transformation

La transformation définie par *Shraideh* [6] permet de transformer un diagramme de Processus vers les Réseaux de Pétri. Cependant par manque des règles dans l'approche et vu l'existence de certains éléments de la notation BPMN qui n'ont pas d'équivalent dans les réseaux de Pétri (pas des règles de transformation).

L'outil AToM³ exploité par EIMENSOURI [3, p. 115] transforme sans conteste le diagramme de processus vers les réseaux de Pétri de manière automatique en utilisant une combinaison entre la méta-modélisation et les transformations de graphes ce qui va permettre l'automatisation des transformations des modèles, BPMN et l'élargissement de l'application de ces transformations à la totalité des modèles associés à un méta modèle donné. Cette transformation automatique souffre d'une absence de règles de transformation pour d'autres éléments de la notation BPMN (pool, ruelle, groupe, les objets de données).

Conclusion

Le but de cet article était de donner une idée sur la vérification et la validation formelle des processus métier. Nous avons mis l'accent sur la notion de processus métier selon l'approche BPMN vu le rôle crucial que jouent les processus métiers dans la prospérité des entreprises d'aujourd'hui. Ceci nécessite une attention particulière lors de leur modélisation afin d'éviter d'introduire des erreurs.

Cependant, comme nous pouvons le constater, la plupart des modèles UML ou BPMN des processus métiers, souffrent de l'absence d'approches effectives d'analyse, et vu la place qu'occupe le processus métier dans le développement des systèmes d'informations, c'est un vrai danger de laisser passer des anomalies.

C'est à juste titre que la vérification et la validation formelle qu'apporte le réseau de Pétri servent à déterminer à l'avance si un modèle de processus respect ses spécifications.

En effectuant cette vérification au moment de la conception, il est possible d'identifier des problèmes potentiels et, le cas échéant, le modèle peut être modifié avant son exécution.

Une analyse des modèles de processus lors de la conception peut grandement améliorer la fiabilité des systèmes.

Dans un cadre spécifique, nous avons présenté les règles de transformation des modèles BPMN (Diagramme de processus) vers les réseaux de Pétri colorés. Dans le souci de clarté sur cette transformation, nous avons démontré par un exemple concret comment appliquer ces règles.

Références

- [1] G. FAYCAL, «Une Approche de Transformation des Diagrammes d'Activités d'UML Mobile 2.0 vers les Réseaux de Petri,» 2009. [En ligne]. Available: <http://eprints.univ-batna2.dz/174/1/Face%20GUERROUF.pdf>. [Accès le 27 01 2018].
- [2] A. (2006)., «Home page:,» 2006. [En ligne]. Available: <http://atom3.cs.mcgill.ca/>. [Accès le 27 01 2018].
- [3] R. EIMENSOURI, «Modélisation et Vérification des processus métiers dans les entreprises virtuelles : Une approche basée sur la transformation de graphes,» [En ligne]. Available: <https://bu.umc.edu.dz/theses/informatique/ELM5432.pdf>. [Accès le 15 09 2017].
- [4] Mahmoud Nassar, Analyse/Conception par points de vue : le vUML, ENSIA, 2005.
- [5] O. R. Maria, Vérification à la volée de contrainte OCL étendues sur des modèles UML, montreal, 2005.
- [6] S. Ahmad, «Integrating a business process analysis and optimization step using BPMN model in a general process design and development approach : application to a paperless mail flow process,» 24 03 2009. [En ligne]. Available: <https://tel.archives-ouvertes.fr/tel-00579520>. [Accès le 27 21 2018].
- [7] Aissam Belghiat., Transformation des modèles UML vers des ontologies OWL, 2017.
- [8] B. MOUNA, Une approche basée transformation de graphes pour la génération de modèles de réseaux de Pétri analysables à partir de Diagramme UML, Avril 2013.
- [9] C. K. Ehrig, Towards Model Transformation in Generated Eclipse Editor plug-ins, Berlin, 2005.
- [10] Meena. Alexander, «Modeling and Simulation of High Impact,» *Computer Engineerig Journal*, 2004.
- [11] R. David and H. Alla, Discrete, Continuous, and Hybrid Petri Nets. Springer, 1 ed., Springer, 2004.
- [12] T. Murata, "Petri nets : Properties, analysis and applications," Proceedings of the IEEE,, vol. 77,, Avril 1989.
- [13] K. Jensen, "An introduction to the practical use of coloured petri nets.," Lecture Notes in Computer Science : Lectures on Petri Nets II : Applications, vol. 1492,, 1998..
- [14] J. Lamp, "Using petri nets to model weltanschauung alternatives in soft systems methodology," Oct. 27 1998.
- [15] Y. CASEAU, Processus et Entreprise 2.0, Dunod, 2011.
- [16] T.Allweyer, BPMN 2.0 : Introduction to the Standard for Business Modeling, 3e ed. Dunod, Février 2010.
- [17] T. S. Staines, "Intuitive mapping of UML 2 activity diagrams into fundamental modeling concept petri net diagrams and colored petri nets," , 2008..
- [18] J. EL. HADDAD, «Travail Coopératif et Nouvelles Technologies,» [En ligne]. Available: <http://docplayer.fr/4519310-Travail-cooperatif-et-nouvelles-technologies.html>. [Accès le 25 01 2018].
- [19] Objecteering, « Guide Pratique des Processus Métiers,» [En ligne]. Available: http://support.objecteering.com/objecteering6.1/tutorials/fr/guides/business_process_guide_fr.pdf. [Accès le 25 01 2018].
- [20] M. D. C. O. Remco M. Dijkman, «Formal Semantics and Analysis of BPMN Process Models using Petri Nets».
- [21] F. C. J. M. J. Tricas, «Using the incidence matrix in an evolutionary algorithm for Computing minimal

siphons in Petri net models. 18th International Conference on System Theory, Control and Computing (ICSTCC 2014).» *Google Scholar*, 2014..