

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERRI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

Master en Automatique

(Option Commande des Systèmes)

Thème

MODELISATION PAR SYSTEME MULTI-AGENT

ET RESEAU DE PETRI :

Application à un atelier flexible.

Dirigé par :

M^{me}. HEDJEM Ourida

Présenté par :

M^{elle}. OULTAF Ouardia

Promotion 2011



REMERCIEMENT

*Nous remercions dieu de nous avoir donné la force et
la volonté pour mener à bien notre travail.*

*Nous tenons à exprimer nos vifs remerciements pour Madame
HEDJEM Ourida, née MEGHERBI pour nous avoir proposé ce sujet,
pour son aide, conseils et son soutien constant
tout au long de la réalisation de ce mémoire.*

*Nous remercions aussi Monsieur KARA.R, Monsieur
GUERMAH.S, pour leur aide précieuse et pour la documentation
qu'ils ont mis à notre disposition.*

*Nous remercions également les membres de jury qui nous
feront l'honneur d'évaluer ce modeste travail.*

*Que tous ceux qui ont contribué de près ou de loin à
la réalisation de ce travail, trouvent ici, l'expression
de notre profonde gratitude.*



Dédicace



*Je dédie cet humble travail à ma très chère mère
qui n'as jamais manquée de me soutenir tout au long
de mes études, à mon chère père qui m'as éduqué et
soutenu.*

À mon frère et mes sœurs.

À mes neveux et ma nièce.

À mes chers(es) amis(es).

Ouardia



Sommaire

INTRODUCTION GENERALE	1
------------------------------------	---

Chapitre I : Généralités sur les Systèmes Multi-Agents.

INTRODUCTION	3
I.1 Notion d'agents	3
I.1.1 Définition d'agent.....	3
I.1.2 L'agent et l'environnement.....	6
I.1.3 Les architecture d'agents	7
a) Architectures d'agent réactif.....	7
b) Architectures d'agent cognitif.....	9
c) Architecture d'agent hybride	14
I.2 Sociétés et organisations	15
I.2.1 Définition d'un Systèmes Multi-Agent (SMA).....	15
I.2.2 Environnement.....	17
I.2.3 organisation	17
a) Organisation émergente.....	18
b) Organisation support d'activités.....	18
I.2.4 Capacités communication.....	18
a) Mécanisme du tableau noir.....	19
b) Mécanisme par message.....	21
I.2.5 Interactions	22
a) La coopération.....	22
b) La coordination.....	23
c) Résolution de conflit.....	25
CONCLUSION	26

Chapitre II : Généralités sur les Réseaux de Petri.

INTRODUCTION	27
II.1 Réseaux de Petri généralisé	27
II.1.1 Définition informelle.....	27
II.1.2 Définition formelle.....	28
II.1.3 Marquage.....	29
II.1.4 Dynamique d'un Réseau de Petri marqué.....	32
II.1.5 Propriétés comportementales des Réseaux de Petri	34
a) Marquage atteignable (accessible).....	35
b) RdP borné.....	37
c) RdP vivant.....	37
d) RdP quasi-vivant.....	38
e) RdP conforme.....	39
f) Etat d'accueil et RdP réinitialisable.....	39
II.1.6 Réseaux de Petri particuliers	39
a) Graphe d'événement (GEV).....	39
b) Graphe d'état (GET).....	40
c) RdP avec conflit vs RdP sans conflit.....	40
d) RdP autonome.....	42
e) RdP non autonome	44
f) RdP à choix libre.....	43
g) RdP simple.....	43
h) RdP pur.....	44
i) RdP à capacités.....	44
j) RdP à priorités.....	44
k) RdP à arc inhibiteur.....	45
II.1.7 Les invariants	46
a) Notion de P-invariant (composante conservative).....	46
b) Notion de T-invariant (composante répétitive).....	47
II.1.8 Méthodes d'analyse	49
a) L'analyse par énumération.....	49
b) L'analyse structurelle (P-invariant).....	49

II.1.9 Les réseaux de Petri de haut niveau	49
a) RdP coloré.....	50
b) RdP à temps discret	51
c) RdP hiérarchiques.....	52
d) RdP récursifs.....	53
CONCLUSION	53

Chapitre III : Modélisation des SMA par RdP.

INTRODUCTION	54
III.1 Notion générales sur les systèmes et modèles	55
III.2 Modélisation d'un Système Multi-Agent par un Réseau de Petri	56
III.2.1 Modélisation des agents	57
III.2.2 Modélisation des interactions entre agents	59
III.2.2.1 Modélisation par RdP ordinaires.....	60
III.2.2.2 Modélisation par RdP colorés	62
a) Représentation structurelle	62
b) Sémantique opérationnelle du modèle.....	63
c) Autre éléments de modélisation.....	63
III.2.2.3 Analyse des interactions	64
III.2.3 Modélisation des plans d'actions.....	64
III.2.3.1 Principe.....	64
III.2.3.2 Modélisation des plans d'actions pour l'allocation des tâches.....	66
a) Hypothèses.....	67
b) Architecture générique du système	67
c) Principe et modélisation par réseaux de Petri.....	69
d) Un RdP permet de visualiser	72
III.2.4 Suivi d'exécution.....	75
CONCLUSION	75

Chapitre IV : Etude de cas.

INTRODUCTION	76
IV.1 Modélisation et validation de protocole Winrdp	76
IV.1.1 Présentation du logiciel	77
a) Présentation de l'interface de saisie graphique.....	77
b) Présentation de l'interface de saisie textuelle.....	77
IV.2 Description du problème	78
IV.2.1 Application de la modélisation	78
IV.2.1.1 Description de la chaine de fabrication des armoires frigorifiques.....	78
IV.2.1.2 Modélisation par SMA	80
a) Premier identification des agents et des rôles.....	81
b) Deuxième identification des agents et des rôles.....	96
c) Scénario d'allocation.....	102
CONCLUSION	104
CONCLUSION GENERALE	105
BIBLIOGRAPHIE	107

Introduction

Générale

Introduction Générale

Les applications d'aujourd'hui deviennent de plus en plus complexes, dynamiques et imprévisibles compte tenu du grand nombre, des acteurs en jeu et de leurs diversités.

Pour répondre à l'accroissement de la compétitivité et à la globalisation des marchés, les entreprises doivent développer des systèmes de production dotés d'une flexibilité maximale. Dans de tels systèmes, l'avantage compétitif est déterminé par la rapidité d'arrivée sur le marché, par l'habilité à satisfaire les préférences des consommateurs et par la réactivité aux problèmes liés aux impacts environnementaux de l'industrie.

Cependant, l'ingénieur pose alors plusieurs problèmes importants : comment réussir à appréhender le comportement de ses systèmes afin de les concevoir, les réaliser et / ou les commander ?

Les Systèmes Multi-Agents (SMA) ont permis de résoudre le problème de l'explosion combinatoire liés à l'Intelligence Artificielle (IA) en évitant de travailler sur une seule entité et en répartissant la manipulation des connaissances sur plusieurs entités intelligentes. Cette répartition est ainsi destinée à réduire l'explosion combinatoire de la recherche. De plus, l'intégration de nouvelles entités intelligentes dans le même environnement a permis d'étendre les travaux de l'Intelligence Artificielle traditionnelle, puisque chaque entité doit intégrer les comportements des autres entités dans son environnement. D'autre part, les Systèmes Multi-Agents ont permis de décrire la coopération homme-ordinateur qui est le thème de nombreuses recherches cherchant à étudier les systèmes homme-machine. Enfin, l'essor des Systèmes Multi-Agents est également lié aux évolutions technologiques de l'informatique, au développement du commerce électronique (où de nombreuses interactions existent entre plusieurs entités intelligentes d'objectifs différents), à l'utilisation d'Internet et au nombre croissant de ses utilisateurs, à la modélisation et la simulation des systèmes complexes.

Finalement, les SMA sont compatibles pour représenter des problèmes possédant de multiples méthodes de résolution, de multiples perspectives. L'approche multi-agents est justifiée par : l'adaptation à la réalité ; la coopération ; la fiabilité ; et la réutilisation. Elle repose sur un ensemble de concepts tels que la communication, l'organisation, la coopération, la coordination entre entités plus au moins intelligentes et autonomes. L'ensemble des concepts de l'approche très riche et très puissante en termes de modélisation.

Il est vrai que l'ensemble des concepts de l'approche multi-agents font de celle-ci une approche très riche et très puissante en termes de modélisation. Néanmoins, bien qu'elle offre de nombreux avantages potentiels, elle doit aussi relever beaucoup de défis. La modélisation des SMA est un défi en elle-même. En effet la richesse de SMA fait qu'aucun formalisme ne peut couvrir à lui seul tous les aspects de ce domaine.

Plusieurs approches existent pour modéliser les SMA. Parmi ces approches il y a celles orientées conception et orientées analyse, dont, le Réseau de Petri (RdP), le langage Z, la logique temporelle, ..., etc.

On s'intéresse dans ce travail, à l'utilisation des Réseaux de Petri comme approche de modélisation orientée analyse pour la modélisation de certains aspects des SMA, car une modélisation totalement orientée conception n'est pas souvent appropriée pour l'analyse, et nécessite souvent une transformation du modèle d'un formalisme vers un autre. D'autre part, une modélisation totalement orientée analyse peut mener à la manipulation de modèles très complexes, et difficilement lisibles (RdP de taille très grande, de nombreuses formules mathématiques,..., etc.).

Ce travail est structuré en quatre chapitres :

- Dans le premier chapitre, nous présentons les notions de base des SMA.
- Le second chapitre est consacré aux RdP.
- Le troisième chapitre est consacré à modéliser les SAM par les RdP.
- Le dernier chapitre est consacré à l'étude de l'évolution des performances et de la validation d'un système de production. A fin d'illustrer cette étude, nous considérons la chaîne de fabrication des armoires frigorifiques de l'ENIEM.

Chapitre I.:

Généralités sur les Systèmes Multi-Agents

INTRODUCTION :

Les Systèmes Multi-Agents (SMA) se situent dans le prolongement de l'Intelligence Artificielle Distribuée (IAD) qui se définit comme l'étude et la conception d'organisations d'agents artificiels pour obtenir des systèmes intelligents. Les systèmes multi-agents se veulent de portée plus générale que l'IAD, puisqu'ils concernent la résolution de problèmes au sens large, la robotique distribuée, la simulation et la conception de mondes hypothétiques.

De la notion de système multi-agent, c'est-à-dire, un modèle électronique ou informatique composé d'entités (ou d'agents) artificielles qui communiquent entre elles et agissent dans un environnement, le concept d'agent reste le pivot de ce domaine et l'interprétation qu'on a de ce terme doit être le premier pas dans l'exploration des univers multi-agents. La façon dont les agents interagissent entre eux permettra ensuite de comprendre la construction des SMA.

I.1 Notion d'agent :

Comme tous les concepts fondamentaux, il est relativement vague. On peut distinguer plusieurs manières de concevoir et de comprendre la notion d'agent. Chacune de ces notions renvoie à un courant de recherche particulier dans le domaine de ce qui touche à la nébuleuse "agent". Ainsi la définition de l'agent est le fruit de la contribution de plusieurs auteurs.

I.1.1 Définition d'agent :

➤ Emazeau et Muller propose un modèle d'agent générique en exhibant les caractéristiques minimales qui permettront d'expliquer les diverses interactions dans un univers multi-agent. L'agent se trouve ainsi explicité en termes de but, de connaissances, de perceptions, de choix, de capacités de raisonnement et de décision, de communication. Ce modèle est résumé dans la figure (I-1) [1]:

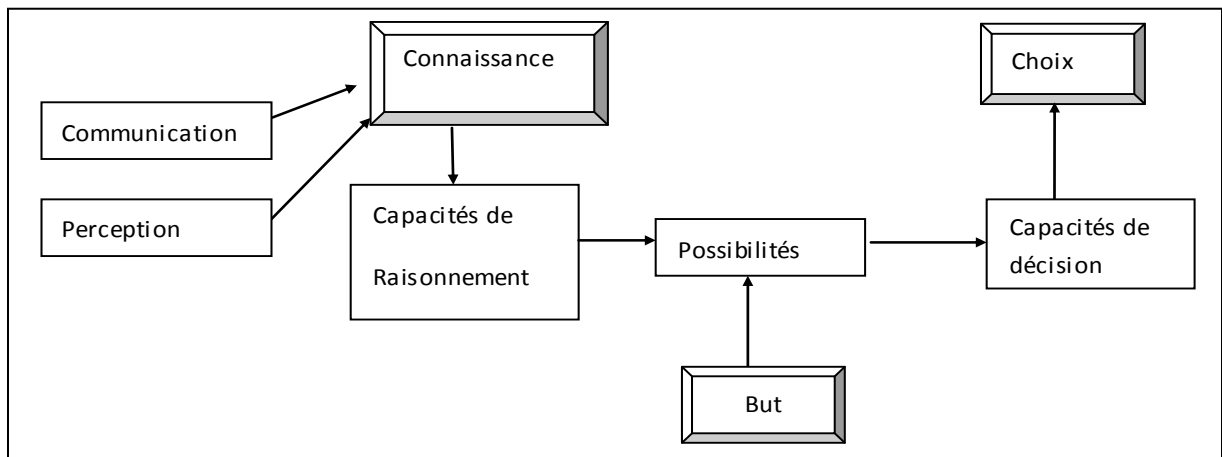


Figure I.1 : Modèle d'agent générique.

➤ Pour Jennings, Sycara et Wooldridge, un agent est un système informatique, situé dans un environnement. Il agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu. Les notions « situé », « autonomie » et « flexible » sont définies comme suit [14]:

✚ **Situé** : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement. Exemple : systèmes de contrôle de processus, systèmes embarqués, le monde réel et l'Internet etc.

✚ **Autonome** : l'agent est capable d'agir sans l'intervention directe d'un humain (ou d'un autre agent) et contrôle ses propres actions ainsi que son état interne.

✚ **Flexible** : l'agent dans ce cas est :

- *Capable de répondre à temps* : capable de percevoir son environnement et élaborer une réponse dans les temps requis.
- *Proactif* : il n'agit pas simplement en réponse à son environnement, il est également capable d'avoir un comportement opportuniste, dirigé par ses buts ou sa fonction d'utilité, et prendre des initiatives au moment approprié.
- *Social* : capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ces agents à accomplir les leurs.

➤ la définition la plus couramment admise de la notion d'agent est celle proposée par J.Ferber [17]:

On appelle agent, une entité physique (robot, avion) ou virtuelle (composant logiciel) :

- a. qui est capable d'agir dans un environnement ;
- b. qui peut communiquer directement avec d'autres agents ;
- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser) ;
- d. qui possède des ressources propres ;
- e. qui est capable de percevoir (mais de manière limitée) son environnement ;
- f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune) ;
- g. qui possède des compétences et offre des services ;
- h. qui peut éventuellement se reproduire ;
- i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

En effet, Un agent représente « une entité qui agit », c'est-à-dire une entité capable de modifier son environnement d'une manière intelligente.

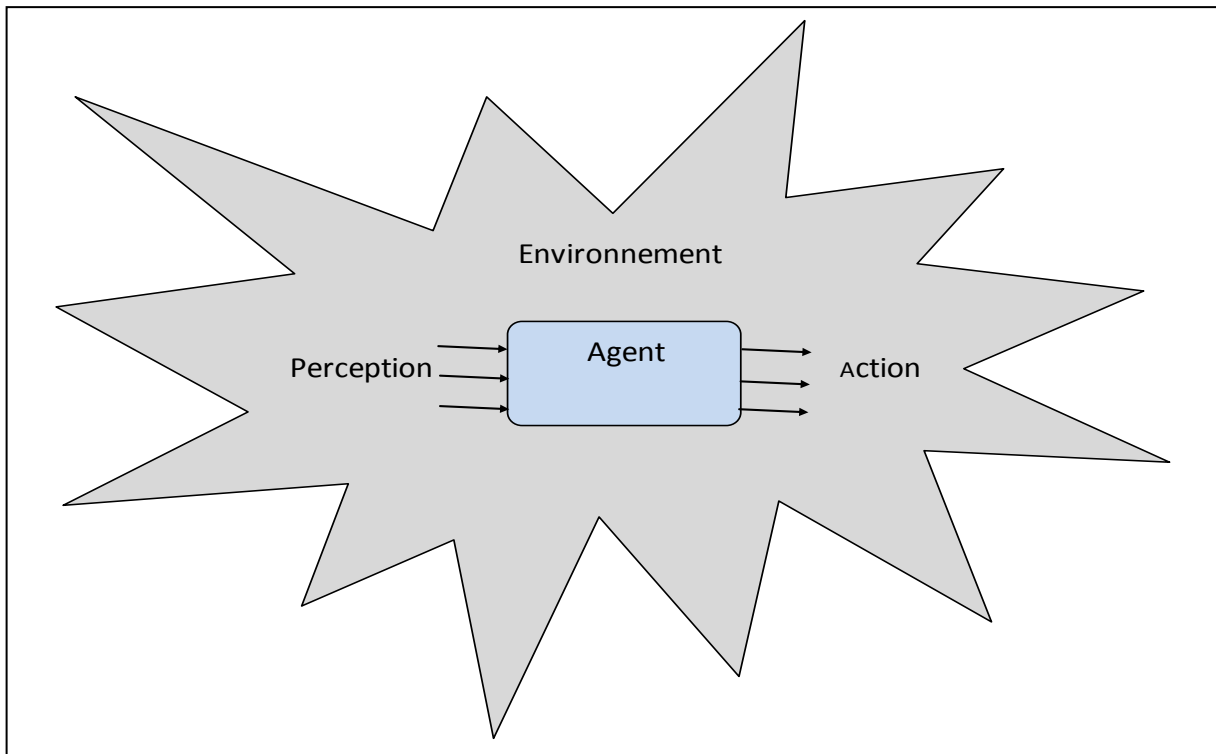


Figure I.2 Aperçu externe et général d'un agent.

Les capacités de l'agent sont de trois ordres qui répondent aux trois phases générales de réalisation d'une tâche (figure I.3) [22] :

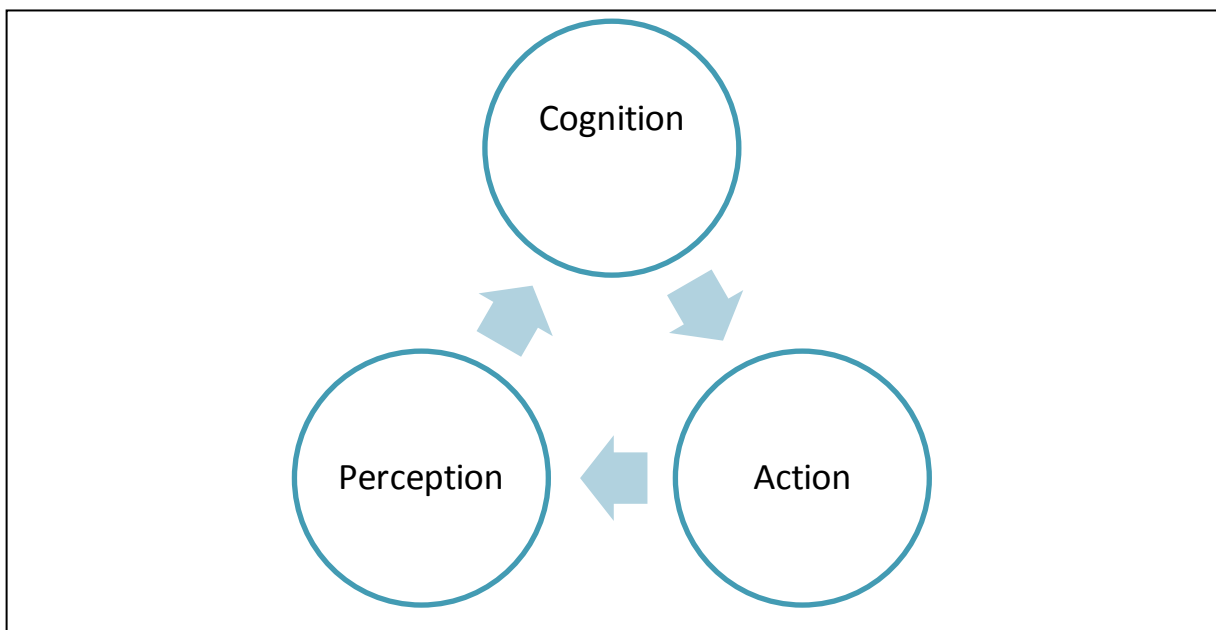


Figure I.3 : Les trois phases générales de réalisation d'une tâche par un agent.

✚ **une phase de perception**: l'agent perçoit son environnement et met à jour ses représentations internes de l'environnement et des autres agents.

✚ **une phase de cognition** : l'agent détermine ce qui est à faire (sa tâche) et décide quand et comment le faire. Une large palette de types de raisonnement est utilisée et implémentée, les extrêmes étant les raisonnements réactifs (réflexe) et cognitifs (de haut niveau).

✚ **une phase d'action** : la réalisation effective des actions qui ont été décidées.

Ces trois phases forment un cycle par le bouclage de la phase action sur la phase perception. En effet, l'environnement évolue continûment et offre donc, à tout moment, au système de nouvelles données à percevoir et de nouvelles tâches à réaliser.

Ces trois phases se retrouvent également au niveau supérieur, à savoir au niveau du système multi-agent.

1.1.2 L'agent et l'environnement :

Un agent se définit généralement par rapport à un environnement. Un agent peut ainsi modéliser une voiture dans un système d'étude du trafic routier, un individu ou un groupe d'individus dans un système représentant une société, un jeton dans un jeu de dames.

L'environnement est assimilé à l'univers dans lequel évoluent les agents. Il peut être simplement un espace métrique pour des agents situés ou un environnement d'exécution (variables d'environnement) pour des agents représentant des processus.

Wooldridge propose un cadre formel dans lequel les notions d'agent et d'environnement sont définies sans aucune ambiguïté. Ainsi un environnement est caractérisé par un ensemble d'état (de l'environnement)

$$S = \{s_1, s_2, \dots\} \quad (\text{I.1})$$

Etant donné un ensemble d'actions possibles

$$A = \{a_1, a_2, \dots\} \quad (\text{I.2})$$

Un agent se définit comme des suites d'états de l'environnement

Action : $S^* \rightarrow A$

$$\text{Où} \quad S^* = \coprod_{i=0}^{+\infty} S^i \quad (\text{I.3})$$

A chacune de ces suites, la fonction action associe une action particulière de l'agent. Cette approche fonctionnelle de l'agent offre des outils appropriés pour caractériser formellement les types d'agents et d'environnements [1].

1.1.3 Les architectures d'agent :

L'architecture d'un agent définit la manière dont il est conçu, c'est-à-dire son organisation interne. Les applications utilisant l'approche multi-agent étant de plus en plus nombreuses, on voit émerger dans la communauté des architectures d'agents, plus au moins dédiées à des systèmes particuliers. Bien que ce ne soit pas encore le cas, l'enjeu est d'aboutir à terme à une véritable méthodologie normalisée de conception et d'implémentation des systèmes à base d'agents. Nous présentons quelques unes des architectures les plus courantes.

a) Architectures d'agent réactif :

L'agent réactif ne fait que réagir aux changements qui surviennent dans l'environnement (suivant un cycle perception/action ou stimulus/réponse). Autrement dit, un tel agent ne fait ni délibération ni planification, il se contente simplement d'acquiescer des perceptions de celles-ci en appliquant certaines règles prédéfinies. Etant donné qu'il n'y a pratiquement pas de raisonnement, l'agent réactif peut agir et réagir très rapidement, il n'est pas capable de tenir compte de ses actions passées.

Les deux sous-sections suivantes décrivent deux modèles qui peuvent servir à la conception d'agents réactifs.

❖ Agent à réflexe simple :

Ce type d'agent agit uniquement en se basant sur ses perceptions courantes. Il utilise un ensemble de règles prédéfinies, du type *Si condition alors action*, pour choisir ses actions. Par exemple, pour un agent en charge du contrôle de la défense d'une frégate, on pourrait avoir la règle suivante :

Si missile-en-direction-de-la-frégate alors lancer-missile-d'interception.

Comme on peut le constater, ces règles permettent d'avoir un lien direct entre les perceptions de l'agent et ses actions. Le comportement de l'agent est donc très rapide, mais peu réfléchi.

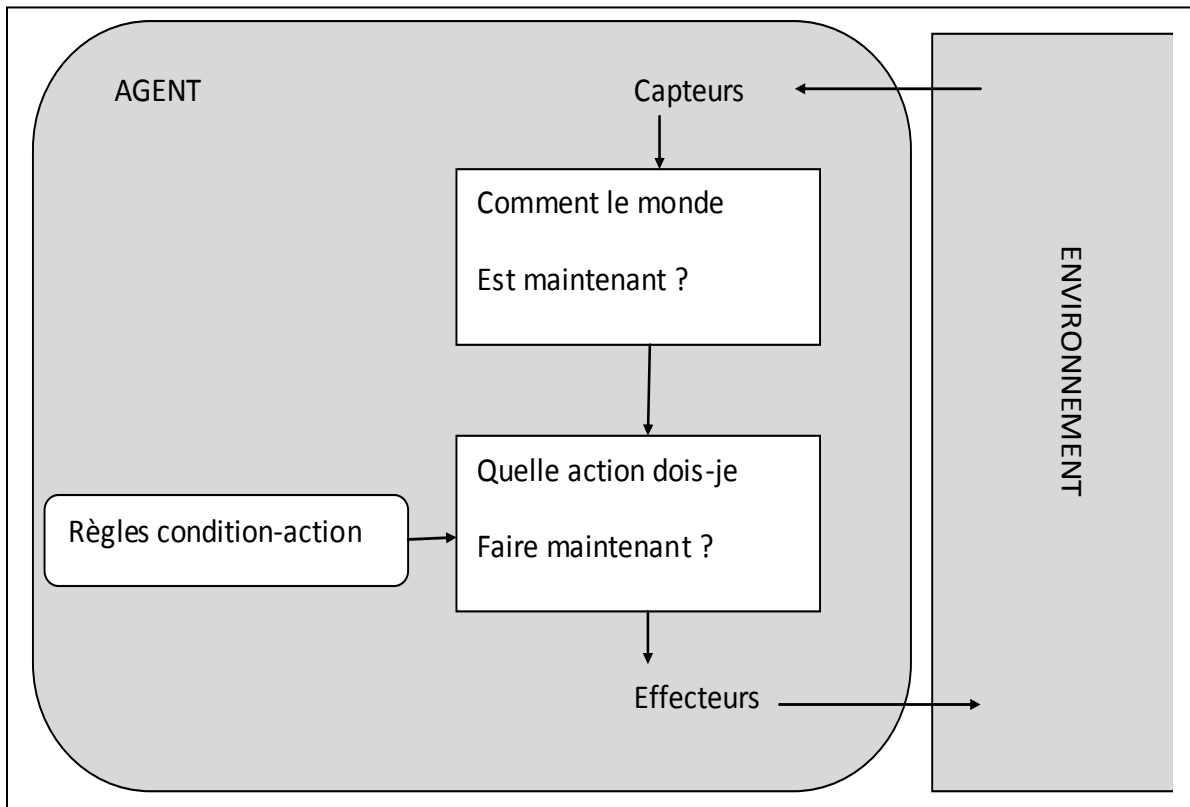


Figure I.4 : Schéma d'un agent à réflexes simples [27].

La figure (I.4) montre l'architecture d'un agent à réflexes simples. Les rectangles représentent l'état interne de l'agent dans son processus de décision et les ovales représentent les informations qui sont utilisées dans le processus. L'agent se bâtit une représentation du monde à l'aide de ses perceptions lui venant de ses capteurs. Par la suite, il utilise ses règles pour choisir l'action qu'il doit effectuer selon ce qu'il perçoit de l'environnement.

❖ **Agent conservant une trace du monde :**

Le type d'agent qui a été décrit précédemment, ne peut fonctionner que si un tel agent peut choisir ses actions en se basant uniquement sur sa perception actuelle. Par exemple, si le radar de la frégate détecte un missile et que l'instant d'après, il le perd de vue, dû à un obstacle, cela ne signifie nullement qu'il n'y a plus de missile. Dès lors, l'agent en charge du contrôle de la frégate doit tenir compte de ce missile dans le choix de ses actions et ce, même si le radar ne détecte plus le missile.

Le problème que nous venons de mentionner survient parce que les capteurs de l'agent ne fournissent pas une vue complète du monde. Pour régler ce problème, l'agent doit maintenir des informations internes sur l'état du monde dans le but d'être capable de distinguer deux situations qui ont des perceptions identiques, mais qui, en réalité, sont différentes. L'agent doit pouvoir faire la différence entre un état où il n'y a pas de missile et un état où le missile est caché, même si ses capteurs lui fournissent exactement les mêmes informations dans les deux cas.

Pour que l'agent puisse faire évoluer ses informations internes sur l'état du monde, il a besoin de deux types d'information. Tout d'abord, il doit avoir des informations sur la manière dont le monde évolue, indépendamment de l'agent. Par exemple, il doit savoir que si un missile avance à une vitesse de 300 m/s, alors 5 secondes plus tard, il aura parcouru 1500 mètres. L'agent doit avoir ensuite des informations sur la manière dont ses propres actions affectent le monde autour de lui. Si la frégate tourne, l'agent doit savoir que tout ce qui l'entoure tourne aussi. Il doit donc mettre à jour la position relative de tous les objets autour de la frégate.

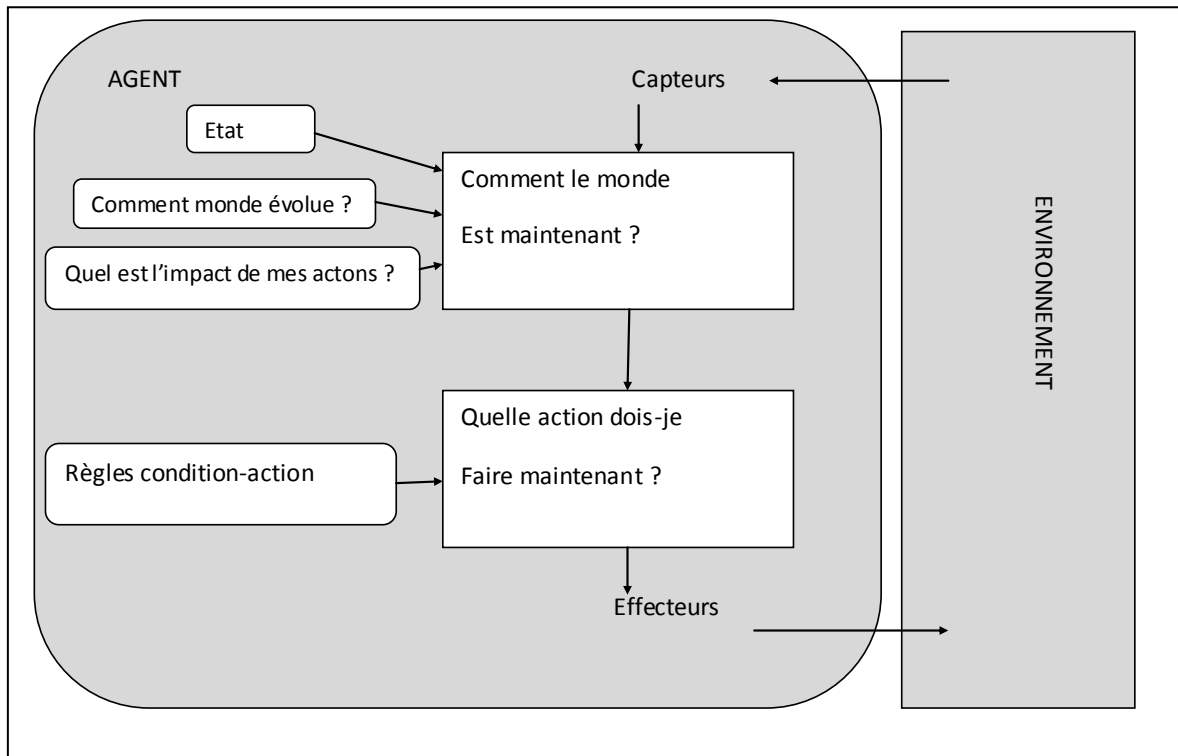


Figure I.5 : Schéma d'un agent conservant une trace du monde [27].

On peut voir, sur la figure (I.5), la structure d'un agent conservant une trace du monde. Il utilise ses informations internes (état précédent du monde, l'évolution du monde et l'impact de ses actions) pour mettre à jour ses perceptions actuelles. Par la suite, il choisit ses actions en se basant sur cette perception « améliorée » du monde qui l'entoure.

b) Architectures d'agent cognitif :

Selon Labidi et Lejouad, les systèmes d'agents cognitifs sont fondés sur la coopération d'agents capables à eux seuls d'effectuer des opérations complexes [3]. Les agents sont donc dotés d'une grande capacité de perception, de raisonnement et d'action. Les agents cognitifs ont une représentation du monde sur laquelle ils peuvent raisonner. Elle regroupe l'ensemble des états mentaux de l'agent.

Nous présentons trois architectures des agents cognitifs :

❖ *Agents ayant des buts :*

Contrairement aux agents réactifs, les agents cognitifs, qui raisonnent sur les buts, tiennent compte d'une certaine projection dans le futur. Ils se posent des questions comme « Qu'est-ce qui va arriver si je fais telle ou telle action ? » et « Est-ce que je serai satisfait si cela se produit ? ». Bien entendu, l'agent raisonnant sur ses buts prend, en général, beaucoup plus de temps à agir qu'un agent réactif. Il offre en revanche beaucoup plus de flexibilité. Par exemple, si nous voulions changer de destination, il faudrait changer toutes les règles de l'agent réactif, tandis que pour l'agent ayant des buts, nous ne changerions que le but.

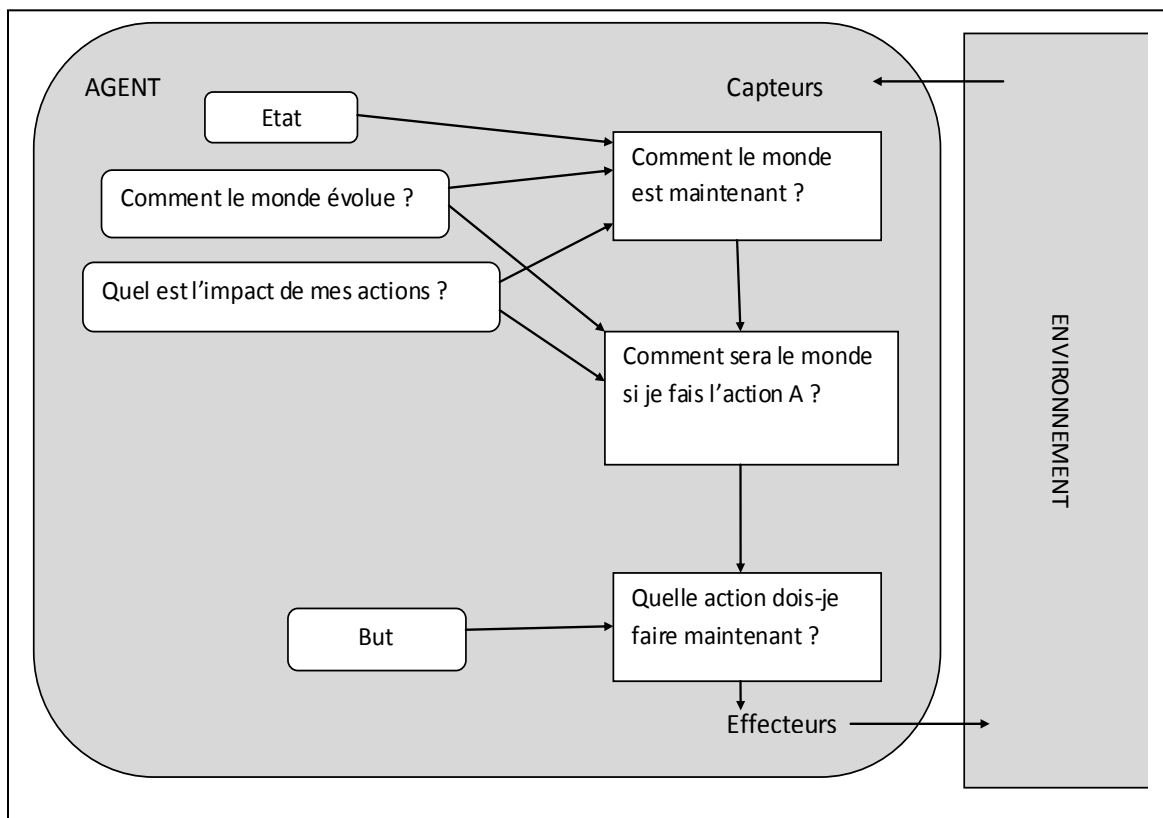


Figure I.6 : Schéma d'un agent ayant des buts [27].

La figure (I.6) montre la structure d'un agent basé sur les buts. Comme on peut le constater, il est identique à l'agent réactif gardant une trace de l'environnement, sauf qu'il se projette dans le futur pour voir l'impact de ses actions et qu'il choisit ses actions en se basant sur ses buts.

❖ *Agents utilisant une fonction d'utilité :*

Dans plusieurs situations, les buts ne sont pas suffisants pour générer un comportement de haute qualité. Par exemple, s'il y a plusieurs chemins possibles pour atteindre un endroit précis, certains seront plus rapides et d'autres plus dangereux. Dans cette situation, l'agent raisonnant uniquement sur ses buts n'a pas de moyens pour choisir le meilleur chemin, son seul but étant de se rendre à destination. Cela se produit car les buts ne procurent qu'une simple distinction entre les états où l'agent est satisfait ou non. En fait, l'agent doit plutôt s'appuyer sur une manière plus fine d'évaluer les états pour être en mesure de reconnaître pour chacun des états son degré de satisfaction. Pour cela, on dit que l'agent va préférer un état à un autre si son utilité est plus grande dans le premier état que dans le deuxième.

Généralement, l'utilité est une fonction qui attribue une valeur numérique à chacun des états. Plus l'état a une grande valeur, plus il est désirable pour l'agent. Dès lors, la spécification d'une fonction d'utilité permet à l'agent de prendre des décisions rationnelles dans deux types de situations où le raisonnement sur les buts échoue. Ainsi, par exemple, lorsqu'il y a des buts conflictuels qui ne peuvent pas être satisfaits en même temps (par exemple, la vitesse et la sécurité), la fonction d'utilité spécifie le compromis approprié entre les différents buts. De même, lorsqu'il y a plusieurs buts possibles, mais qu'aucun d'eux ne peut être atteint avec certitude, la fonction d'utilité permet de pondérer la chance de succès avec l'importance de chacun des buts.

La figure (I.7) montre le schéma d'un agent basé sur l'utilité. On peut voir que l'agent utilise la fonction d'utilité pour évaluer la pertinence d'une action. Il choisit donc les actions qui l'amèneront dans les états ayant la plus grande valeur d'utilité pour lui.

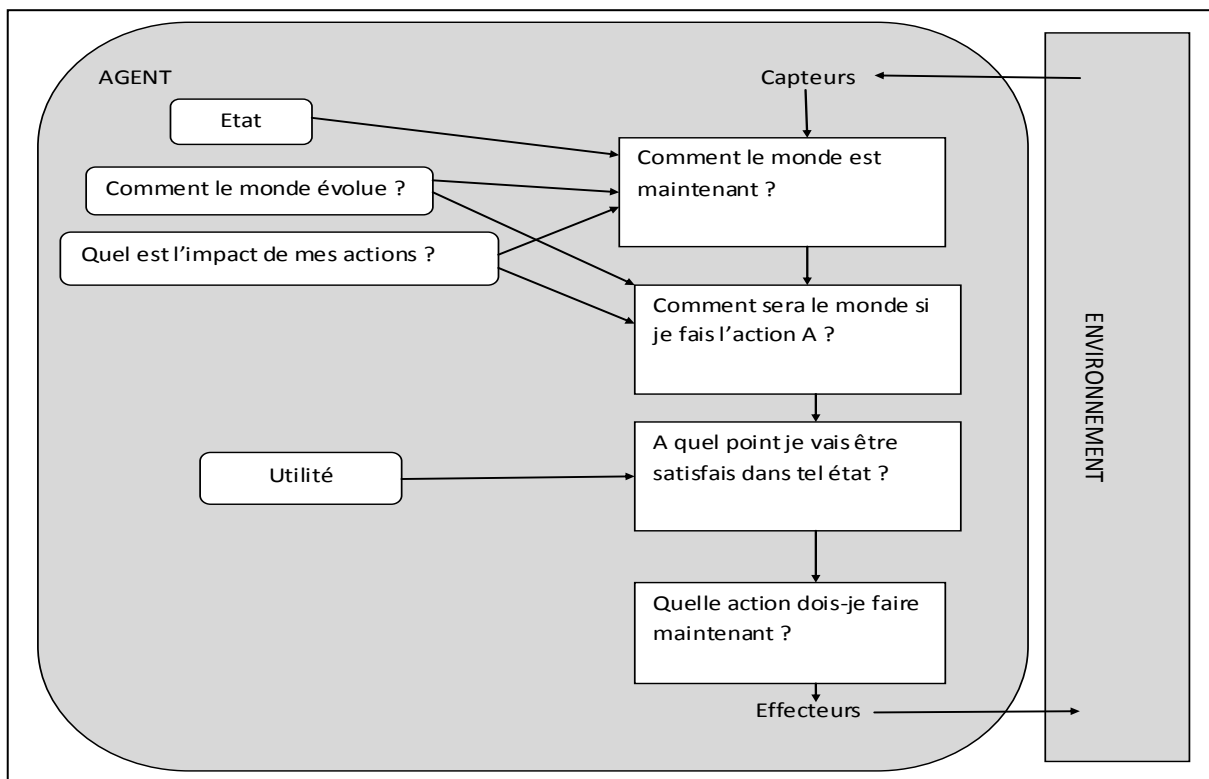


Figure I.7 : Schéma d'agent basé sur l'utilité [27].

❖ **ASIC (Architecture for Social and Individual Control) [14]:**

Cette architecture proposée par Boissier se présente comme un modèle de contrôle vertical en trois couches (figure (I.8)) et chaque couche respecte la décomposition classique Perception-Raisonnement-Action. L'architecture ASIC est sous la forme de contrôle hiérarchique, la couche de décision définit l'objectif de l'agent, la couche d'adaptation adapte la loi de commande en fonction de l'évolution de l'environnement pour un objectif donné et enfin pour la couche commande qui détermine la commande à appliquer au processus par l'application de la loi de commande définit en vue de satisfaire le but défini.

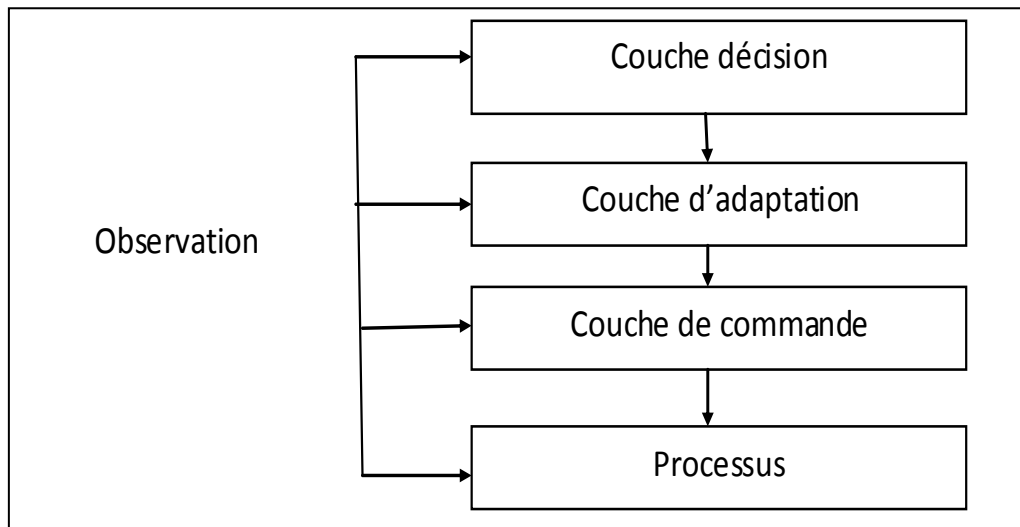


Figure I.8 : Architecture d'une couche de contrôle d'après.

La présence des trois couches n'est pas nécessaire, car un agent peut avoir un but fixe et il n'aura pas de besoin de la couche de décision, et un agent n'adaptant pas sa manière de satisfaire ses buts selon le contexte, n'aura pas besoin de la couche adaptation. Ces différents modules agissent en fonction de l'état interne de l'agent.

❖ **Agents BDI (Beliefs, Desires, Intention) [14] :**

L'architecture BDI est une autre approche utilisée dans la conception des agents délibératifs. BDI est un acronyme qui signifie, en anglais, Belief, Desire, Intentions. Ce qui se traduit en français par croyances, désirs et intentions. Les agents se basent donc sur ces trois aspects pour choisir leurs actions. Dans ce cadre, Wooldrige propose une architecture ayant sept composantes, telles que présentées sur la figure (I.9).

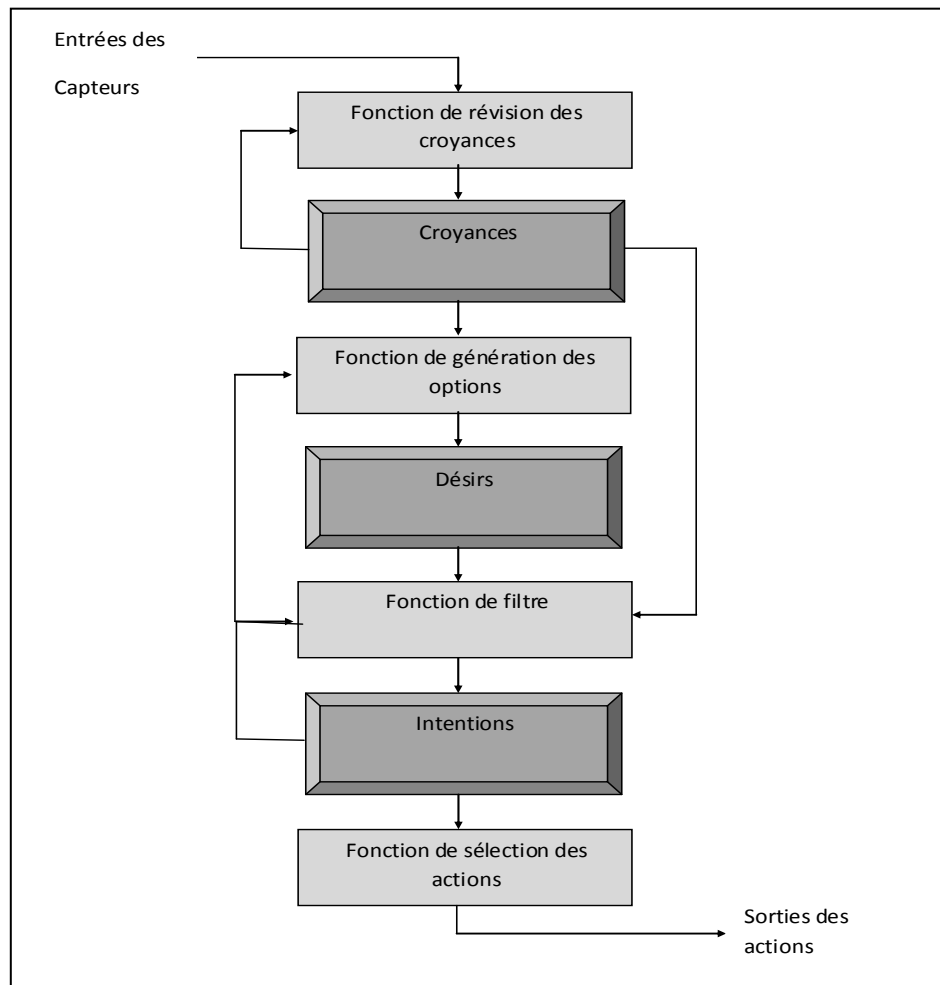


Figure I.9 : Diagramme d'une architecture BDI.

✚ Un ensemble de croyances courantes, représentant les informations que l'agent possède à propos de son environnement courant ;

✚ Une fonction de révision des croyances, qui prend les entrées des capteurs et les croyances actuelles de l'agent et qui détermine un nouvel ensemble de croyances ;

✚ Une fonction de génération des options, qui détermine les options disponibles pour l'agent (i.e. ses désirs), en se basant sur les croyances courantes de l'agent à propos de son environnement et sur ses intentions courantes ;

✚ Un ensemble de désirs, représentant les options disponibles à l'agent ;

✚ Une fonction de filtre, qui représente le processus de délibération de l'agent et qui détermine les intentions de l'agent en se basant sur ses croyances, ses désirs et ses intentions courantes ;

Un ensemble d'intentions courantes, représentant le centre d'attention actuel de l'agent, c'est-à-dire les buts envers lesquels il s'est engagé et envers lesquels il a engagé des ressources ;

✚ Une fonction de sélection des actions, qui détermine l'action à effectuer en se basant sur les intentions courantes de l'agent.

En résumé, un agent BDI doit donc mettre à jour ses croyances avec les informations qui lui proviennent de son environnement ; décider quelles options lui sont offertes, filtrer ces options afin de déterminer de nouvelles intentions et poser ses actions en se basant sur ses intentions.

c) Architectures d'Agent Hybride :

Les sections précédentes ont présenté deux types d'architectures : réactive et cognitive. Chacune de ces architectures est appropriée pour un certain type de problème. Pour la majorité des problèmes cependant, ni une architecture complètement réactive, ni une architecture complètement cognitive n'est appropriée. Comme pour les humains, les agents doivent pouvoir réagir très rapidement dans certaines situations (comportement réflexe), tandis que dans d'autres, ils doivent avoir un comportement plus réfléchi. Dans ce cas, une architecture conciliant à la fois des aspects réactifs et cognitifs est requise. On parle alors d'architecture hybride, dans laquelle on retrouve généralement plusieurs couches logicielles. Les couches peuvent être arrangées verticalement (seulement une couche a accès aux capteurs et aux effecteurs) ou horizontalement (toutes les couches ont accès aux entrées et aux sorties).

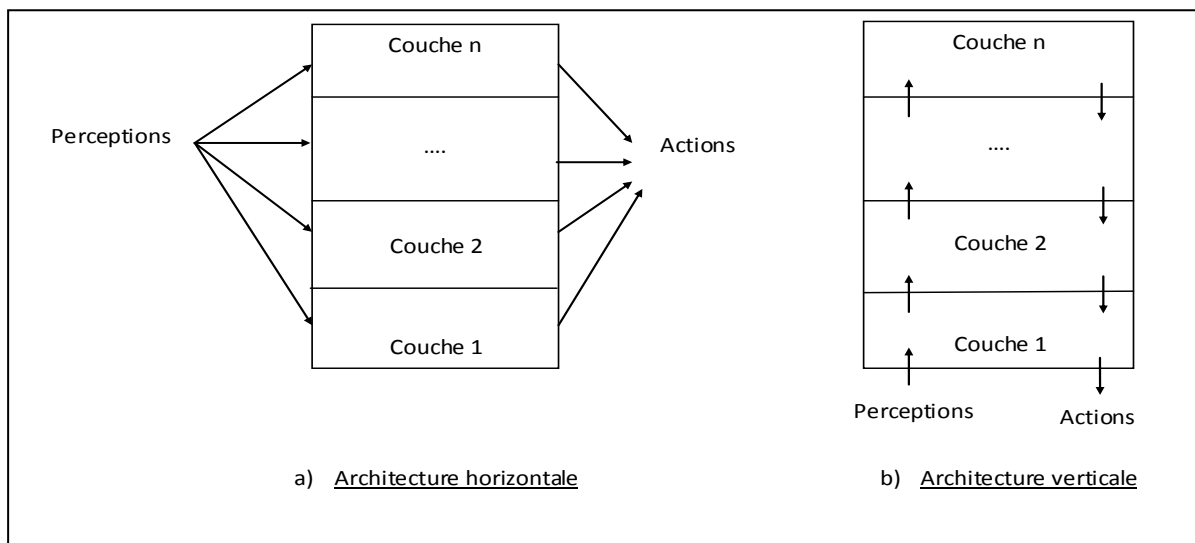


Figure I.10 : Architectures d'agents en couches.

Dans ce type d'architecture, les couches sont arrangées de manière hiérarchique. Les différents niveaux de la hiérarchie traitent les informations provenant de l'environnement à différents niveaux d'abstraction. La plupart des architectures considèrent que trois couches suffisent amplement [27]. Ainsi, au plus bas niveau de l'architecture, on retrouve habituellement une couche purement réactive, qui prend ses décisions en se basant sur des données brutes en provenance des capteurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision qui se situe au niveau des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement. Dans cette dernière couche, on retrouve généralement une représentation des autres agents (leurs buts, leurs croyances, etc.). Pour produire le comportement global de l'agent, ces trois couches interagissent ensemble. Ces

interactions varient beaucoup d'une implémentation à une autre, et c'est la raison pour laquelle elles ne sont pas décrites ici.

I.2 Sociétés et Organisations:

Après avoir évoqué les différentes natures et structures d'agents, nous allons analyser la façon dont ils se constituent en société, de plus, les agents sont en relation dans l'objectif d'interagir et forment ainsi un système multi-agent.

I.2.1 Définition d'un Système Multi-Agent (SMA) :

La définition la plus couramment admise de la notion d'un système multi-agent (SMA) est celle proposé par J.Ferber [17]:

On appelle système multi-agents, un système composé des éléments suivants :

- a. Un environnement E , c'est-à-dire un espace disposant généralement d'une métrique.
- b. Un ensemble d'objets O . Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
- c. Un ensemble A d'agents, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.
- d. Un ensemble de relation R qui unissent des objets (et donc des agents) entre eux.
- e. Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O .
- f. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Un système multi-agent est « un ensemble d'entités qui coordonnent leurs connaissances, buts, expériences et plans pour agir ou résoudre des problèmes, incluant le problème de la coordination lui-même » ou encore « Un monde artificiel peuplé de processus interagissant est appelé système multi-agent ».

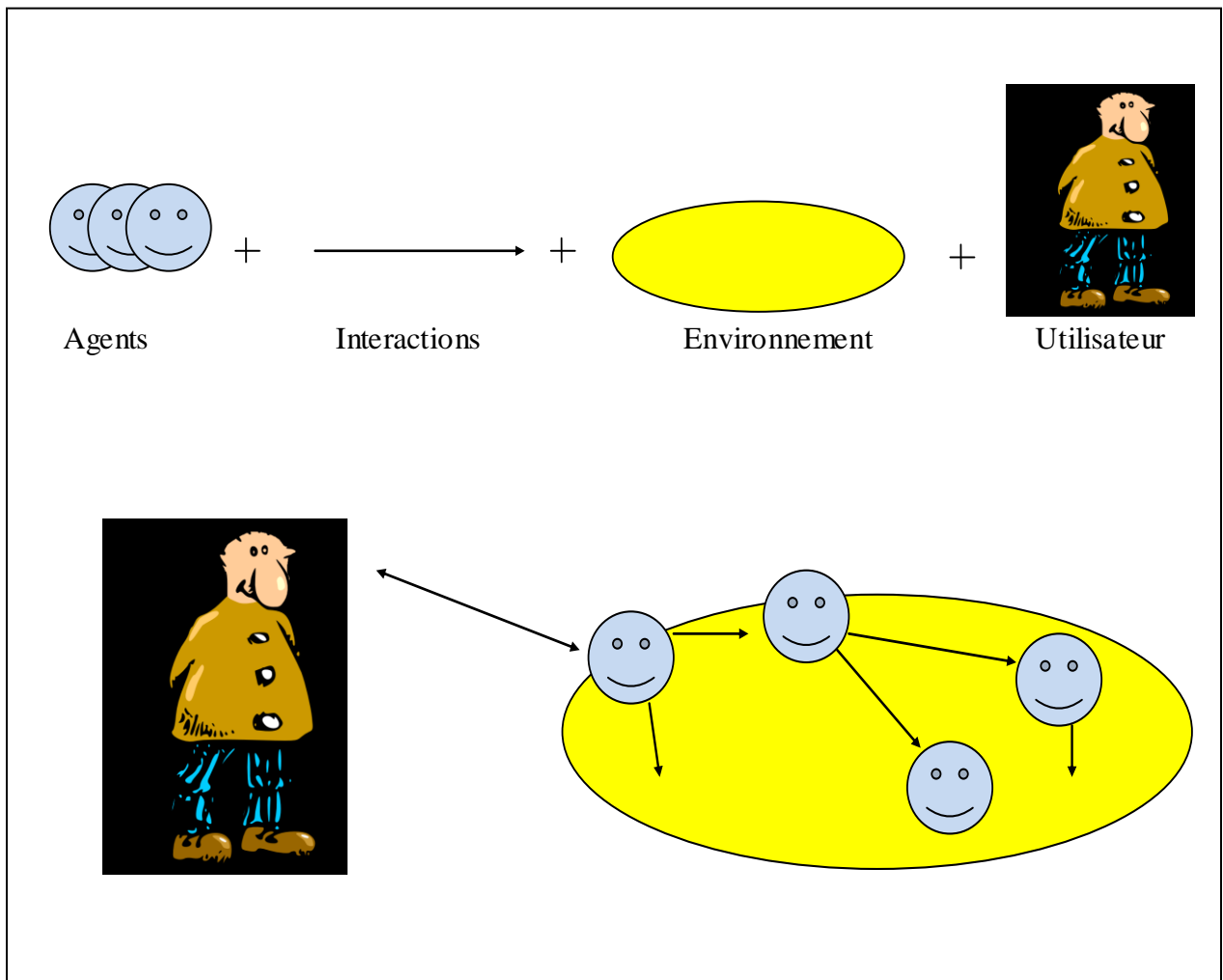


Figure I.11 : SMA [30].

✚ Caractéristiques:

Un système multi-agent est caractérisé par :

- ✓ chaque agent a des informations ou des capacités de résolution de problème limitée. Ainsi, chaque agent à un point de vue partiel.
- ✓ Il n'y a aucun contrôle global du système multi agents.
- ✓ les données sont décentralisées.
- ✓ le calcul est asynchrone.

Un système multi-agent est un système composé d'agent qui communique et collaborent pour accomplir des objectifs spécifiques individuels ou collectifs. La communication implique l'existence d'un espace partagé support de cette communication. Cet espace est généralement qualifié d'environnement.

1.2.2 Environnement:

L'environnement est un élément très important dans le processus dynamique d'un SMA.

Selon Weyns, il peut être défini comme étant tout ce qui entoure les agents, permettant à ces derniers la communication entre eux, représentant le support de leurs actions en définissant les règles et en renforçant ces actions [2]. Il est observable par les agents et enfin se charge de l'activité propre des objets et des ressources présents en son sein.

Russell et Norvig, proposent une caractérisation des différents types d'environnement [2] :

✓ **accessible versus inaccessible** : un environnement est dit accessible lorsque les agents ont accès à l'intégralité de son état. L'accessibilité de l'environnement n'est pas une propriété souhaitable pour un SMA où la portée des actions et la perception est locale.

✓ **déterministe versus indéterministe** : un environnement est déterministe si un changement de son état est déterminé uniquement par son état courant et les actions des agents. Un environnement indéterministe peut produire deux résultats différents pour une même action.

✓ **statique versus dynamique** : un environnement est dit statique s'il n'évolue que sous l'effet des actions des agents. Au contraire, un environnement dynamique possède ses propres processus d'évolution qui peuvent le faire changer d'état sans intervention des agents.

✓ **discret versus continu** : on parle d'environnement discret lorsque la fonction de perception est discrète.

Exemple de quelques systèmes et de leur environnement :

Une fourmilière et son territoire ; une base de données distribuée et l'ensemble de ses périphériques d'utilisation ; l'équipe de surveillance d'une centrale nucléaire et les pupitres de commande des dispositifs de supervision ; une entreprise et la situation du marché économique. Les objets sur lesquels agissent les différents systèmes sont alors : la nourriture aux alentours de la fourmilière ; les primitives de saisie et d'affichage de données ; les vannes du circuit de refroidissement ; le prix de vente des produits ; etc.

1.2.3 Organisation :

L'organisation est l'un des concepts de base des systèmes multi-agents. Il s'agit du cadre organisationnel qui fixe les rôles des agents dans le groupe et donc les règles d'interaction entre eux. Les modèles de sociétés humaines et animales ont largement influé sur les études dans ce domaine.

Deux modèles d'organisation fondamentalement différents sont souvent distingués dans la littérature.

a) Organisation Emergente :

Un premier modèle concerne l'émergence d'organisations. En effet, graduellement, une forme d'organisation se met en place, ou plus précisément, un observateur à l'impression de voir apparaître (émerger) un comportement structuré parmi les agents [22].

Exemple :

La fourmi ouvrière vaque à sa tâche sans se préoccuper de l'activité des fourmis qui l'entourent mais la fourmilière évolue et survit, en donnant l'impression d'être très organisée. Chaque entité, par son action locale, participe au développement de l'ensemble sans qu'aucune d'entre elles ne dirige les autres ; il suffit que les conditions environnementales varient pour que chaque individu réagisse différemment et que, par conséquent, des changements soient observables au niveau du groupe.

b) Organisation Support d'Activités :

Cette structure est plus ou moins flexible et auto-adaptative, mais les comportements des agents vis-à-vis de leurs congénères sont délimités par leurs rôles respectifs : « Une organisation fournit un schéma pour l'activité et l'interaction par la définition des rôles, comportements et relations d'autorité (contrôle). » [22]. Ce modèle se rapproche plus des organisations humaines car chaque agent est autonome tout en ayant connaissance de ce qu'il doit faire et comment se comporter avec tel ou tel agent en fonction de leurs rôles respectifs.

Exemple :

Un agent se comporte différemment avec un agent pair (de même niveau hiérarchique) qu'avec un agent manager (supérieur hiérarchique).

Ces modes de comportement induisent des organisations moins flexibles que celles proposées par le modèle d'émergence. En revanche, ils ont l'avantage de représenter de manière explicite les liens entre les agents ou les rôles joués par ceux-ci.

1.2.4 Capacités Communicatives :

La communication agrandit les capacités perceptives des agents en leur permettant de bénéficier des informations et du savoir-faire des autres agents. Les communications sont indispensables à la coopération et il est difficile de concevoir un système d'agents coopérant s'il n'existe pas un système permettant aux agents d'échanger des informations ou de véhiculer des requêtes. Les recherches développent deux mécanismes de support de la communication, à savoir des communications à l'aide de tableau noir et des communications à l'aide de messages.

a) Mécanisme du Tableau Noir [22]:

✚ *Un tableau noir est une base de données structurée en niveaux d'abstractions. Son rôle consiste à stocker tous les éléments générés pendant la résolution d'un problème. Une combinaison de ces différents éléments permet de représenter les résultats partiels du problème.*

Un système utilisant un tableau noir comporte au minimum les trois composants de la figure (1-12).

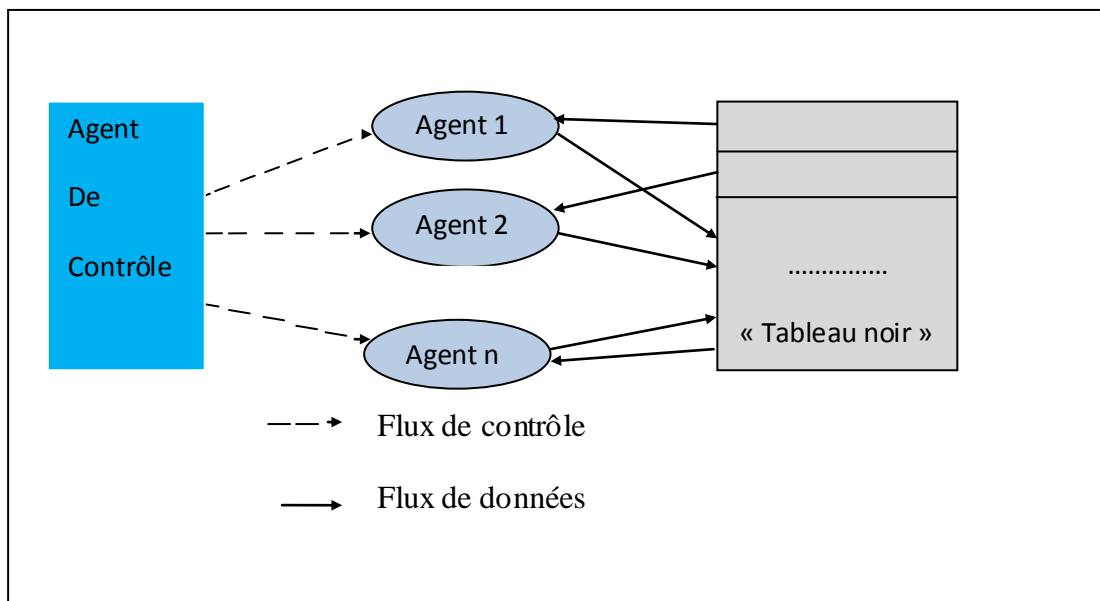


Figure I.12 : Trois composants d'un système de type « tableau noir ».

✚ *Les agents stockent leurs hypothèses dans un tableau noir. Ces agents représentent des modules indépendants. La coopération entre ces agents s'effectue par des ajouts, des suppressions ou des modifications des éléments du tableau noir. Sous certaines conditions, les agents peuvent reconnaître les situations dans lesquelles ils peuvent intervenir et ainsi faire progresser la solution. En revanche, lorsque l'agent est sélectionné, il peut modifier une certaine zone du tableau.*

✚ *Le mécanisme de contrôle (ou agent de contrôle) constitue le cœur d'une architecture de type tableau noir. Le contrôle est généralement basé sur un cycle répétitif en trois phases différentes (figure I.13) :*

- dans une première phase de *détection*, les éléments les plus récents du tableau, appelés *événements*, intéressent un certain nombre d'agents désirant contribuer à la résolution de problème. Les agents intéressés, appelés « *agents instanciés* » sont enregistrés par le mécanisme de contrôle ;
- dans une seconde phase de *sélection*, le mécanisme de contrôle choisit parmi les agents instanciés, des « *agents exécutables* ». Ce choix tient compte de l'état de la solution

courante, de la rapidité d'exécution, des critères de priorité et des travaux potentiels de chaque agent. Dans cette phase, le mécanisme de contrôle doit aussi résoudre les conflits potentiels entre différents agents ;

- dans une troisième phase, le tableau noir est mis à jour par *l'exécution* des actions des agents. Cette mise à jour génère de nouveaux événements.

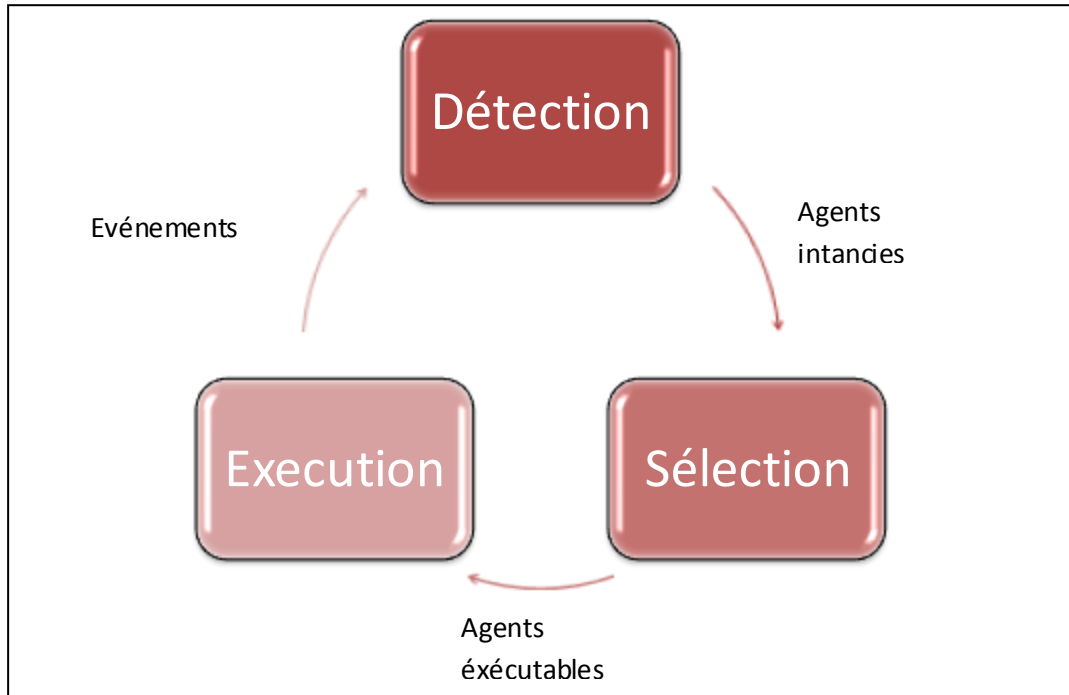


Figure I.13 : Cycle de contrôle d'un « tableau noir ».

Un système basé sur un tableau noir peut être considéré comme dirigé par les événements représentant les changements dans le tableau. Se contentant d'inscrire leurs résultats dans le tableau, les agents interagissent de manière indirecte.

En général, les avantages reconnus des systèmes basés sur un tableau noir sont essentiellement la possibilité de gérer une très grande quantité d'informations et de décomposer les problèmes, lors de leurs différentes étapes de résolution, en niveaux d'abstractions pour une application donnée. Les recherches entreprises montrent aussi deux autres avantages des tableaux noirs, à savoir le parallélisme des agents et le temps réel.

Les systèmes basés sur un tableau noir sont caractérisés par une communication indirecte puisque les différents agents communiquent par l'intermédiaire du module de contrôle. Ils sont aussi caractérisés par une « communication centralisée », puisque ce module de contrôle active et désactive les différents agents. Cependant, des recherches récentes ont souligné des goulets d'étranglement liés à ce type de mécanisme.

b) Mécanisme par Messages :

Le mécanisme par transmission de messages suppose une communication directe entre les agents, puisqu'il n'existe pas de modules intermédiaires entre eux. Chaque agent désirant participer au système, par exemple, de négociation doit s'inscrire auprès du serveur de noms en donnant son nom, son adresse, le type d'application de négociation qu'il veut rejoindre (par exemple, des enchères hollandaises) et ses propres ressources qu'il veut négocier.

Chaque agent désirant envoyer un message à un ensemble de participants contacte le serveur de noms qui se charge d'envoyer le message. En effet, les participants connaissent seulement le nom des autres participants, pas leur adresse. Ils doivent donc contacter le serveur de noms pour envoyer leurs messages car il est le seul à connaître les adresses des participants. Lorsqu'un agent est déconnecté, le serveur de noms stocke tous les messages à son intention dans sa boîte aux lettres créée à son inscription.

L'agent reçoit tous les messages de sa boîte aux lettres lors de sa reconnexion au système.

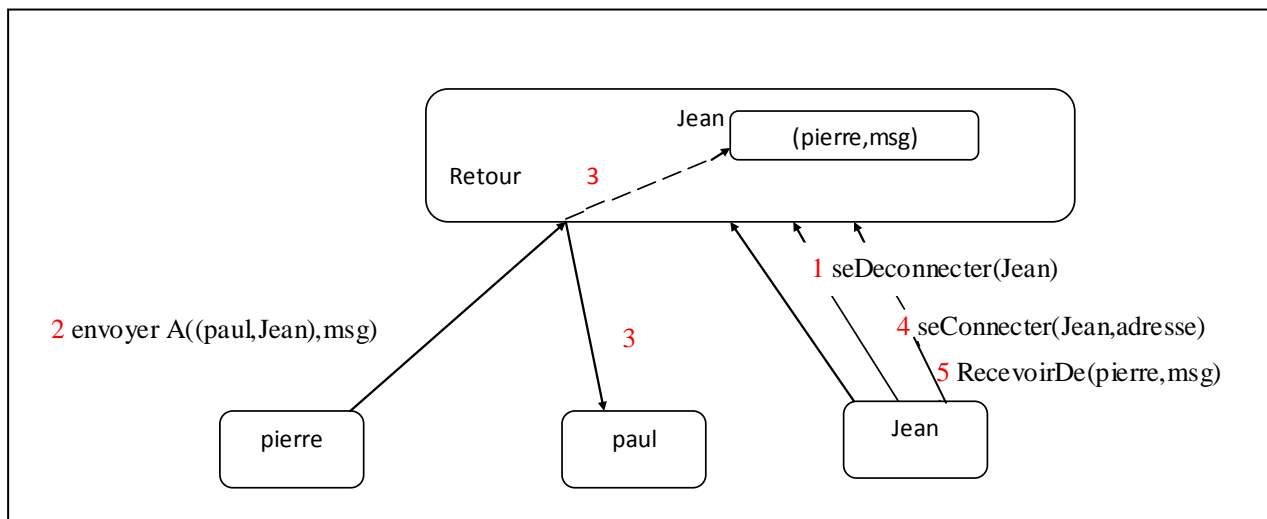


Figure I.14 : Exemple de mécanisme par message.

La figure (I.14) représente le mécanisme d'envoi d'un message de Pierre à Paul et à Jean alors que Jean est déconnecté. Pierre envoie au routeur le message msg et la liste des destinataires, Paul et Jean. Le routeur consulte la liste des destinataires et envoie le message à Paul qui est connecté et place le message pour Jean dans sa boîte aux lettres comme il est déconnecté. Lorsque Jean se connecte à nouveau, le routeur lui envoie le message de Pierre et vide sa boîte aux lettres [17].

Le mécanisme de tableau noir suppose une communication centralisée et indirecte. Le mécanisme par transmissions de message, en revanche, a l'avantage de définir une communication décentralisée et directe.

I.2.5 Interactions :

C'est parce qu'ils coopèrent que les agents peuvent accomplir plus que la somme de leurs actions, mais c'est aussi à cause de leur multitude qu'ils doivent coordonner leurs actions et résoudre des conflits [17].

D'où la nécessité de définir des schémas d'interactions tels que la coopération, la coordination et la résolution de conflit.

a) La Coopération [14]:

La coopération est la contribution des agents à satisfaire un but commun ou individuel. Il existe de multiples manières d'envisager la coopération dans l'approche multi agent.

Trois formes de coopération peuvent être distinguées :

- *la coopération confortative*, selon laquelle une tâche est exécutée par plusieurs agents de spécialités différentes œuvrant de manière concurrente sur le même ensemble de données, le résultat étant obtenu par « fusion ».
- *la coopération augmentative*, selon laquelle une tâche est répartie sur une collection d'agents similaires, œuvrant de manière concurrente sur des sous-ensembles disjoints de données, la solution étant obtenue sous la forme d'un ensemble de solutions locales.
- *la coopération intégrative*, selon laquelle une tâche est décomposée en sous-tâches et ses agents de spécialités différentes et œuvrant de manière coordonnée, la solution étant obtenue au terme de leur exécution.

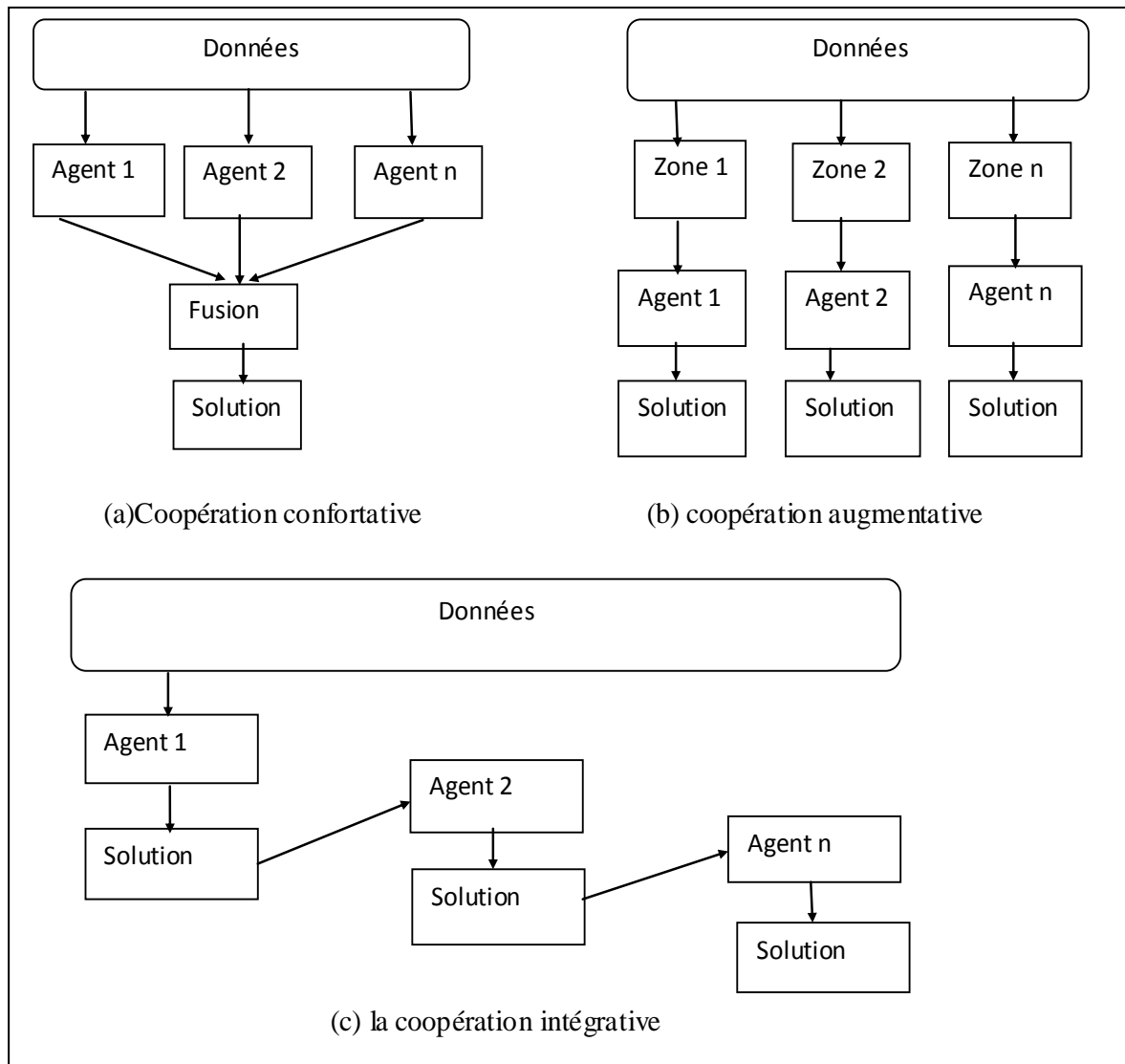


Figure I.15 : Les trois formes de la coopération.

b) La Coordination [14] :

Cette étape correspond au contrôle social au collectif du système. Le but de la coordination est d'organiser l'activité des agents (coopération) dans le temps et dans l'espace, on distingue généralement deux étapes :

➤ *la coordination centralisée* : Constituée d'une seule entité, le coordinateur est responsable de la bonne exécution, donc il va agir suivant son plan à envoyer des ordres d'actions aux agents esclaves et attendre un accusé de réception ou bien le résultat d'exécution d'une action peut ordonner l'action suivante. Cette méthode de coordination facilite la conception du programme, mais réduit les performances du système (temps).

➤ *la coordination distribuée* : Dans le cas où plusieurs agents décident de coopérer, chaque agent va choisir un protocole d'interaction adapté à la coopération qu'il souhaite mettre en œuvre. Ces protocoles sont sous la forme d'un automate à états finis ou sous la forme d'un

réseau de pétri. Cette approche vise à augmenter l'autonomie décisionnelle des agents, ce qui améliore la flexibilité et l'ouverture du système.

c) Résolution de Conflit :

La collaboration des deux premières formes d'interactions provoque des conflits. Autrement dit, Le conflit apparaît lorsqu'il existe une relation de dépendance entre les actions des agents. Une action A1 (agent A1) dépend d'une action A2 (agent A2) si la partie commune à leurs environnements d'application est non nulle.

Du fait de l'autonomie des agents, c'est-à-dire de leur capacité à déterminer leur propre comportement, ils sont amenés à se trouver dans des situations où leurs intérêts peuvent être contradictoires, les agents sont alors en situation objective de conflits. Ces situations proviennent essentiellement d'un problème d'accès à des ressources (tous élément environnementaux et matériel utiles à la réalisation d'une action) limitées : deux agents désirent au même moment quelque chose qui ne peut être partagé ou pour lequel toute répartition diminue ce que l'un des agents aurait pu obtenir si l'autre n'avait pas été présent. Par exemple, des professeurs qui veulent avoir le même cours au même moment dans la même classe, ils se trouvent en situation conflictuelle.

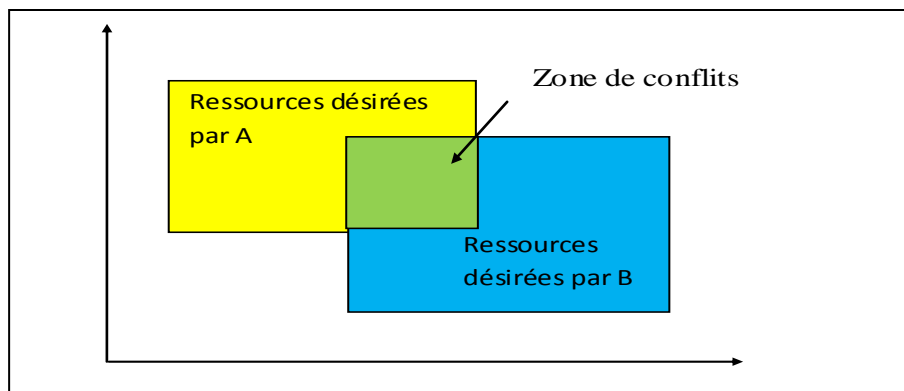


Figure I.16 L'agent A désire des ressources qui recoupent celles qui sont désirées par B, ce qui engendre un conflit.

Les moyens de résoudre un conflit sont fournis par l'organisation, les mécanismes de coordination et les capacités sociales individuelles des agents.

✚ Solution apportée par l'organisation :

Les règles d'interaction issues de l'organisation permettent de résoudre les conflits dans lesquels les rôles respectifs des intervenants dictent les comportements à adopter [22].

Exemple :

La relation hiérarchique indique clairement quel est le décideur entre deux agents.

+ Solution apportée par la coordination :

Les mécanismes de coordination que les agents sont capables de mettre en œuvre fournissent également des solutions aux situations de conflit [22].

Exemple :

Le vote (dans le cas de la coordination distribuée), qui donne raison à la majorité, est un moyen de prendre une décision dans un groupe. L'appel d'offre et les enchères sont également des mécanismes classiquement utilisés.

+ Solution apportée par le raisonnement [14] :

Les techniques de négociation servent ainsi à satisfaire les parties impliquées en établissant des compromis ou en dépassant la nature du conflit.

Nous avons deux catégories de négociation :

➤ *négociation par compromis :*

- Chacune des parties relâche les contraintes les moins importantes.
- Il y a accord lorsque toutes les contraintes sont satisfaites.

➤ *négociation intégrante :*

- Cherche à identifier les buts profonds (changement de but)
- Trouver une solution qui satisfasse complètement ces buts et non les propositions de « surface ».

CONCLUSION :

Ce chapitre a été consacré à une présentation des systèmes multi-agents. Il a été possible de constater la richesse et la diversité des approches, issues initialement d'une branche de l'IA qui consistait à distribuer l'intelligence : l'intelligence artificielle distribuée. En raison de l'essor des nouvelles technologies logicielles ainsi que de l'analyse et de la programmation orientées objet, ce domaine s'est fortement développé.

Cette synthèse sur les systèmes multi-agents comporte les éléments fondamentaux à connaître et à utiliser par les ingénieurs, explicitement dans la conception de tels systèmes ou, implicitement, dans la manipulation d'un noyau déjà développé.

La conception d'un tel système est basée sur deux aspects. Le premier aspect concerne les spécifications d'un agent. Elles ont souligné l'importance des propriétés liées aux informations manipulées, à l'identité de cet agent qui a un comportement autonome et, enfin, à son comportement intelligent. Ce comportement intelligent de l'agent peut être implémenté par différents moyens de description des raisonnements. Le second aspect concerne, ensuite, les spécifications liées aux interactions inter agents. Ces interactions mettent en œuvre des capacités communicatives et sociales.

À partir des différents éléments essentiels, nous pouvons alors concevoir un système multi-agent « minimal ». Lorsque cela est développé, nous pouvons ensuite l'utiliser pour des applications quelconques, à condition bien sûr de modéliser et de représenter l'expertise correspondante ceci sera exposé dans la suite du travail.

Chapitre II:

Généralités sur les Réseaux de Petri

INTRODUCTION :

Les Réseaux de Petri (RdP) ont été introduits par le mathématicien Allemand Carl Adam Petri dans sa thèse "*Communication avec des Automates*" en Allemagne à Bonn en 1962, d'où leur appellation Réseaux de Petri (ou *Petri Nets* en anglais). Ces derniers méritent bien leur appellation car la thèse de C.A Petri a présenté un certain nombre d'idées fondamentales du modèle. Mais la théorie des RdP en sa totalité, telle que nous la connaissons actuellement est le résultat de contribution directe ou indirecte des travaux de plusieurs chercheurs de différentes universités et de différents laboratoires.

Les RdP sont des outils à la fois graphiques et mathématiques permettant de modéliser le comportement dynamique des systèmes à événements discrets. Leur représentation graphique permet de visualiser d'une manière naturelle le parallélisme, la synchronisation, le partage de ressources, les choix (conflits), ...etc. Leur représentation mathématique permet d'analyser le modèle pour étudier ses propriétés et de les comparer avec le comportement du système réel.

Dans le présent chapitre, nous nous intéressons au formalisme RdP. Nous présentons dans un premier lieu les RdP généralisés. Ensuite, nous rappelons certaines de leurs méthodes d'analyse. À la fin de ce chapitre nous citons quelque extension des RdP.

II.1 Réseaux de Petri généralisé :

Dans cette section, nous nous intéressons aux Réseaux de Petri dits, Réseaux de Petri généralisés.

II.1.1 Définition informelle :

Informellement un réseau de Petri (RdP) est un graphe orienté bipartite comprenant deux sortes de nœuds: des places et des transitions. Des arcs relient certaines places à certaines transitions ou certaines transitions à certaines places.

On représente graphiquement les places par des cercles et les transitions par des barres (Figure II.1). Les places servent à représenter les états du système modélisé, tandis que les transitions représentent les changements d'état ou les événements. Quelques interprétations typiques des transitions et leurs places d'entrées et de sorties sont présentées dans La table (II.1) [12].

Places d'entrée	Transition	Places de sortie
Pré-conditions	Evénement	Post-conditions
Données d'entées	Traitement	Données de sorties
Signaux d'entées	Processeur	Signaux de sorties
Ressources demandées	Tâche	Ressources libérées
Conditions	Clauses en logique	Conclusions
Buffers	Processeur	Buffers

Table II.1 : Quelques interprétations typiques de transitions et de places.

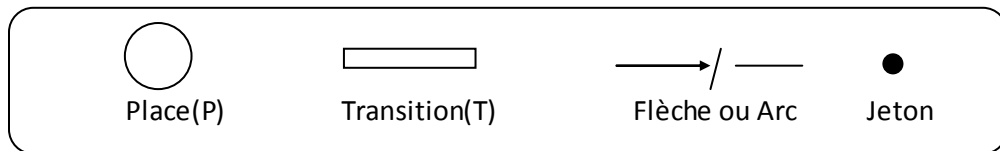


Figure II.1 : Représentation graphique des éléments de RdP.

Chaque arc reliant P_i à T_j (respectivement T_j à P_k) un nombre naturel est ainsi associé: $Pre (P_i; T_j)$ (respectivement $Post (P_k, T_j)$), Ce nombre est appelé « poids » et peut être différent de 1 [29].

Le RdP dont tous ses arcs sont de poids "1" est appelé *RdP ordinaire*. Dans le cas où les arcs peuvent avoir des poids supérieurs à "1", il s'agit de *RdP généralisé*.

II.1.2 Définition formelle :

D'après Murata, un réseau de petri (RdP) *généralisé non marqué* est un quadruplet $Q = \langle P; T; Pre; Post \rangle$ [28].

Sans aucune spécification de marquage initiale M_0 , tel que:

- $P = \{P_1, \dots, P_n\}$ est un ensemble fini non vide de places ;
- $T = \{T_1, \dots, T_m\}$ est un ensemble fini non vide de transitions ;
- $F \subseteq (P \times T) \cup (T \times P)$ Ensemble d'arcs;
- Pre** est une application d'incidence avant

$$Pre : P \times T \rightarrow N,$$

$$Pre (P_i, T_j)$$

Post est une application d'incidence arrière.

$$Post : P \times T \rightarrow N$$

$$Post (P_k, T_j)$$

$$P \cap T = \emptyset \quad \text{Et} \quad P \cup T \neq \emptyset$$

Un *RdP généralisé marqué* est un quintuplet $R = \langle P; T; Pre; Post; M_0 \rangle$ (ou $R = \langle Q; M_0 \rangle$) avec une spécification de marquage initiale M_0 .

Exemple N°1:

Exemple d'un RdP généralisé marqué.

$$P = \{P_1, P_2\}$$

$$T = \{T_1, T_2\}$$

$$F = \{(P_1, T_1), (T_1, P_2), (P_2, T_2), (T_2, P_1)\}$$

$$Pre (P_1, T_1) = 1, Pre (P_2, T_2) = 2$$

$$Post (P_2, T_1) = 1, Post (P_1, T_2) = 3$$

$$M_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

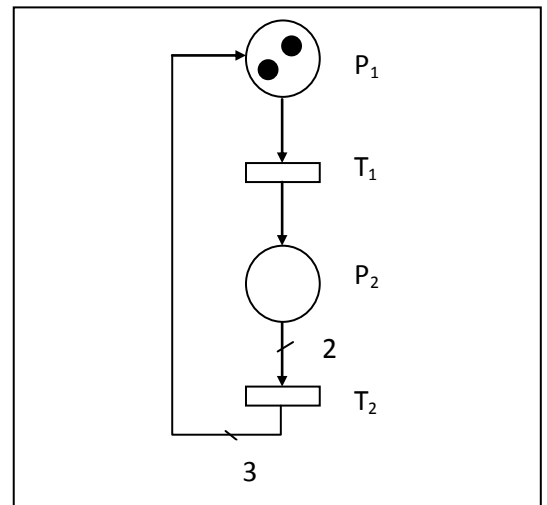


Figure II.2 : Exemple d'un RdP généralisé.

II.1.3 Marquage :

On appelle *marquage* une distribution de jetons sur les places. Le marquage initial noté M_0 est la distribution initiale des jetons dans le réseau à l'instant initial. Un marquage définit l'état du système [28].

Le marquage d'un RdP est une application vectorielle $M : P \rightarrow N$ donnant pour chaque place le nombre de jetons qu'elle contient, la dimension du vecteur de marquage $M(p)$ est égale au nombre de places. Le marquage de la place P_i est noté par $M (P_i)$ qui est un nombre entier.

Notations :

A l'aide des applications *Pre* et *Post*, on peut définir les ensembles des places (transitions) d'entrée (de sortie) d'une transition (place) [29] :

✚ Ensemble des places d'entrée de la transition T_j :

$${}^oT_j = \{P_i \in P \mid Pre(P_i, T_j) > 0\} \quad (II.1)$$

✚ Ensemble des places de sortie de la transition T_j :

$$T_j^o = \{P_i \in P \mid Post(P_i, T_j) > 0\} \quad (II.2)$$

✚ Ensemble des transitions d'entrée de la place P_i :

$${}^oP_i = \{T_j \in T \mid Post(P_i, T_j) > 0\} \quad (II.3)$$

✚ Ensemble des transitions de sortie de la place P_i :

$$P_i^o = \{T_j \in T \mid Pre(P_i, T_j) > 0\} \quad (II.4)$$

Franchissement de la transition T_1 ou de la transition T_2	$T_1 + T_2 = T_2 + T_1$
Franchissement de la transition T_1 puis la transition T_2	$T_1 T_2 \neq T_2 T_1$
Franchissement de la transition T_1 puis la transition T_1	$T_1^2 = T_1 T_1$
Séquence de longueur 2	$T_1 T_2$
Séquence de longueur nulle	λ
Répétition du franchissement de T_1 un nombre quelconque de fois	$T_1^* = (\lambda + T_1 + T_1^2 + \dots)$

Table II.2 : Notation des séquences de franchissement.

Afin de pouvoir les décrire de façon plus compacte, on associe aux applications *Pre* et *Post* les matrices suivantes :

Matrice d'incidence avant associée à l'application *Pre*, elle est définie par :

$$W = [w^-_{ij}] \quad \text{avec:} \quad w^-_{ij} = Pre(P_i, T_j) \quad (II.5)$$

Matrice d'incidence arrière associée à l'application *Post*, elle est définie par :

$$W^+ = [w^+_{ij}] \quad \text{avec:} \quad w^+_{ij} = Post(P_i, T_j) \quad (II.6)$$

Matrice d'incidence : on définit la matrice d'incidence comme suit :

$$W = W^+ - W^- \quad (II.7)$$

Alors, la structure de RdP sera donnée par le quadruplet $Q = (P, T, F, W)$, et le RdP marqué sera noté par $R = (Q, M_0)$.

Remarque :

- ✓ La matrice d'incidence W est indépendante du marquage.
- ✓ La matrice d'incidence W est liée à la structure du RdP. Si le réseau est pur alors il est possible de reconstruire le RdP à partir de sa matrice d'incidence.

Exemple N°2 :

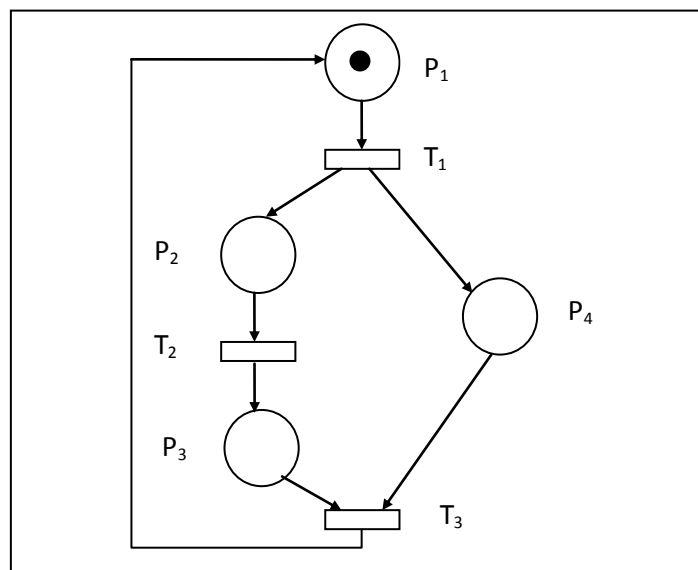


Figure II. 3 : Exemple N°2.

$$\begin{aligned} {}^{\circ}T_1 &= \{P_1\} ; {}^{\circ}T_2 = \{P_2\} ; {}^{\circ}T_3 = \{P_3, P_4\} ; \\ T_1^{\circ} &= \{P_2, P_4\} ; T_2^{\circ} = \{P_3\} ; T_3^{\circ} = \{P_1\} ; \\ {}^{\circ}P_1 &= \{T_3\} ; {}^{\circ}P_2 = \{T_1\} ; {}^{\circ}P_3 = \{T_2\} ; {}^{\circ}P_4 = \{T_1\} \end{aligned}$$

$$P_1^{\circ} = \{T_1\} ; P_2^{\circ} = \{T_2\} ; P_3^{\circ} = \{T_3\} ; P_4^{\circ} = \{T_3\} ;$$

$$w^- = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad w^+ = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

La matrice d'incidence est :

$$w = w^+ - w^- = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

III.1.4 Dynamique d'un RdP marqué :

Un RdP est un graphe muni d'une sémantique opérationnelle, c'est-à-dire qu'un comportement est associé au graphe, ce qui permet de décrire la dynamique du système représenté. L'évolution d'un RdP correspond à l'évolution de son marquage au cours du temps, cette évolution simule le comportement dynamique du système modélisé. Le marquage du réseau change selon les règles suivantes [29]:

- a) Une transition T_j est dite *validée* (ou *sensibilisée*) si chaque place d'entrée P_i (oT_j) contient au moins un nombre de marque égal au poids de l'arc reliant P_i à T_j , donc :

$$T_j \text{ est validée} \Leftrightarrow M(P_i) \geq \text{Pre}(P_i, T_j) \quad \forall P_i \in {}^oT_j$$

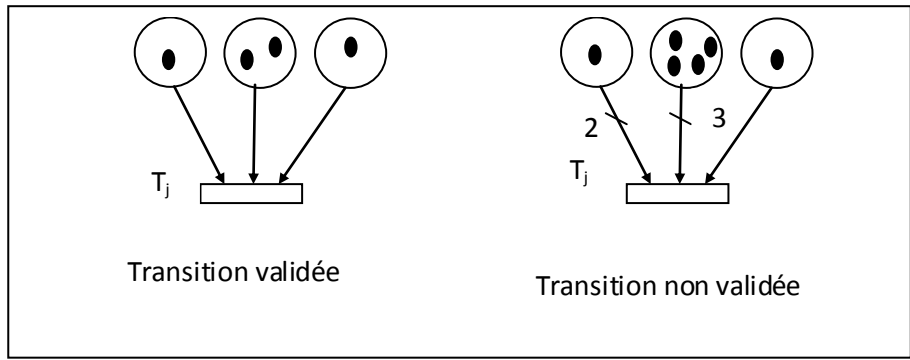


Figure II.4 : Exemples de transitions validées et non validées.

- b) Le franchissement d'une transition validée T_j retranche de chaque place d'entrée P_i un nombre de marques égal au poids de l'arc reliant P_i à T_j et ajoute dans chaque place de sortie P_k un nombre de marques égal au poids de l'arc T_j à P_k .

Après franchissement, on obtient un nouveau marquage M' donné par :

$$M'(P_i) = M(P_i) - \text{Pre}(P_i, T_j) + \text{Post}(P_k, T_j) \quad \forall P ; i, k, j \in \mathbb{N}$$

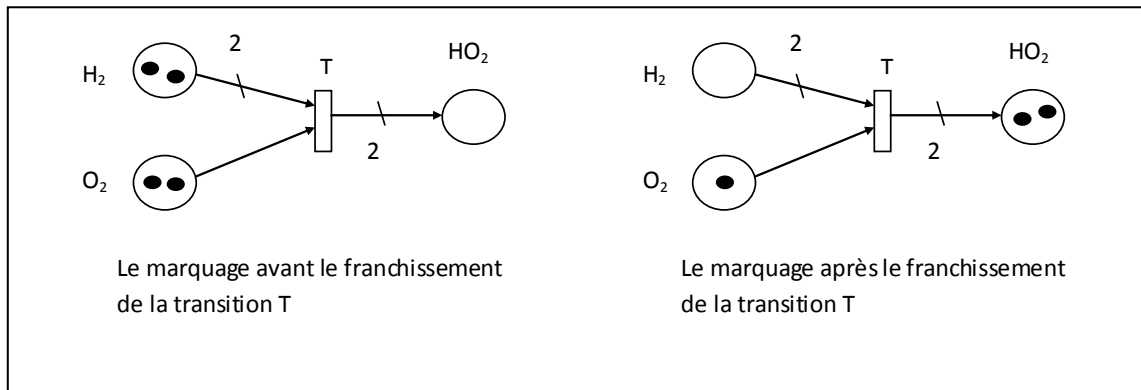


Figure II.5 : La composition d'eau ($2H_2+O_2\rightarrow 2H_2O$) sous forme d'un RdP.

c) Lorsqu' une transition est validée, cela n'implique pas qu'elle sera immédiatement franchie ; cela ne représente qu'une possibilité de franchissement, dans un RdP, même si plusieurs transitions sont validées par un même marquage une et seulement une transition peut être franchie.

Transition Source, Transition Puits :

- Transition source est une transition qui n'a pas de place en entrée de la transition.
- Transition puits est une transition qui n'a pas de place en sortie de la transition.

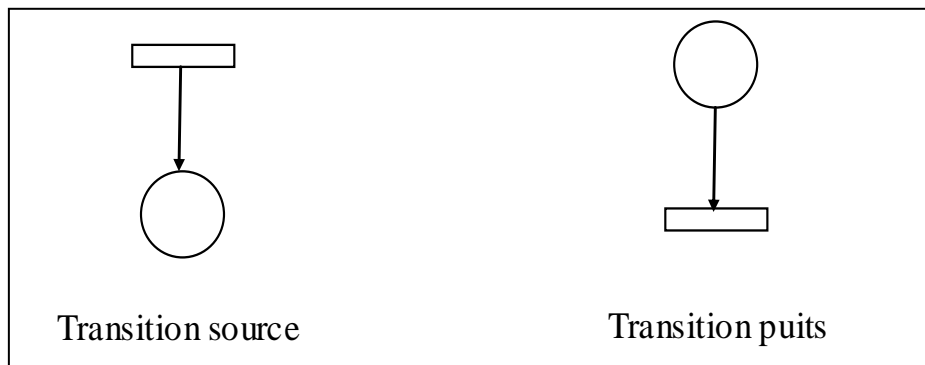
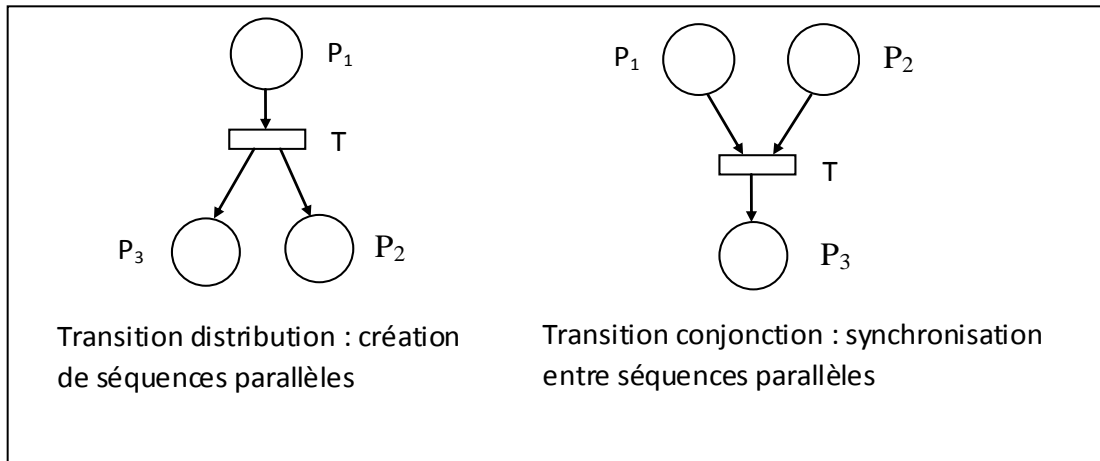


Figure II.6 : Transition Source, Transition Puits.

Transition Distribution, Transition Conjonction [20]:**Figure II.7** : Transition Distribution, Transition Conjonction.**II.1.5 Propriétés comportementales des réseaux de Petri :**

La construction de la structure du RdP qui modélise le système réel est une tâche très importante. La tâche suivante qui est de même importance est l'étude ou l'analyse des propriétés de ce réseau. Il existe deux classes de propriétés. Celles qui sont dépendantes d'un marquage initial M_0 dites comportementales, si l'on change ce marquage, rien ne garantit que ces propriétés tiennent encore. D'autres qui sont indépendantes du marquage initial dites structurelles. Dans ce qui suit nous présentons les propriétés comportementales d'un RdP.

La méthode la plus utilisée dans l'étude des propriétés d'un RdP est la méthode par construction du graphe des marquages accessibles, alors, qu'est-ce que un graphe de marquage accessible ? Le *graphe de marquage accessible* est composé de nœuds qui représentent les marquages accessibles, et d'arcs correspondant aux franchissements de transitions faisant d'un marquage à un autre.

Exemple N°3 :

Pour l'exemple de la figure (II.5), on aura le graphe de marquage accessible suivant :

$$[2 \ 2 \ 0]^T \xrightarrow{T} [0 \ 1 \ 2]^T$$

a) Marquage atteignable (Accessible) :

Le franchissement d'une transition validée dans un RdP apporte une modification au marquage initial M_0 . Un marquage M_n est dit *accessible* à partir du marquage initial M_0 , s'il existe une séquence de franchissement $S = T_1 T_2 \dots T_n$ qui transforme M_0 en M_n [29].

Dans ce cas on écrit :

$$M_0 [S > M_n$$

Qui signifie que M_n est accessible à partir de M_0 par S .

- ✚ L'ensemble de tous les marques accessibles à partir de M_0 dans un RdP R est noté par $R(M_0)$.
- ✚ L'ensemble de toutes les séquences de franchissement possibles à partir de M_0 est noté $L(M_0)$.
- ✚ Le vecteur de franchissement (ou caractéristique) d'une séquence de franchissement S , notée σ , est le vecteur de \mathbf{N}^m tel que σ_j est le nombre de franchissements de la transition T_j dans la séquence S .

Equation fondamentale : Si S est une séquence franchissable à partir du marquage M_i qui mène au marquage M_K :

$$M_i [S > M_K$$

De là on aura l'équation d'état suivante [20]:

$$M_K = M_i + W \sigma$$

Exemple N° 4 :

A partir du RdP représenté par la figure suivante on aura:

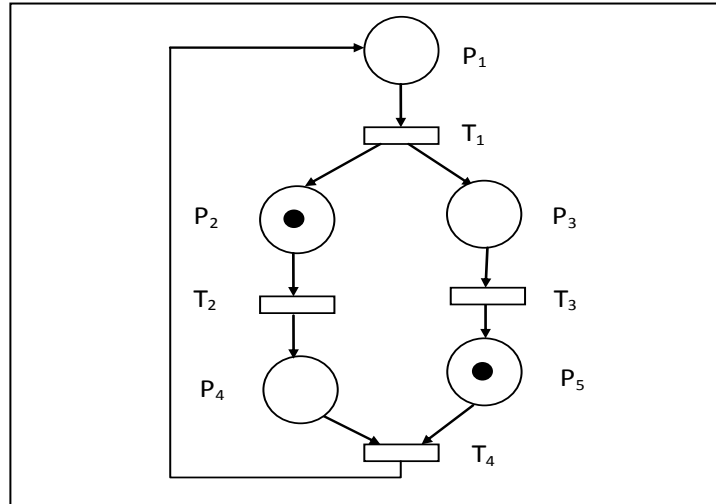


Figure II.8 : Exemple d'un RdP.

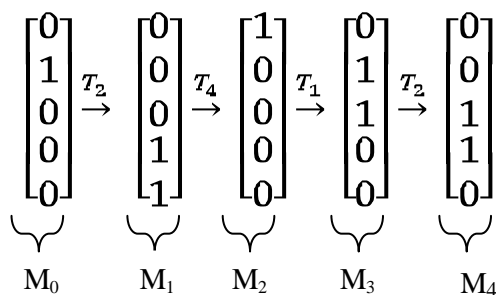
- le marquage initial est :

$$M_0 = M(p_2) = [0 \ 1 \ 0 \ 0 \ 1]^T$$

- L'ensemble de toutes les marques accessibles à partir de M_0 :

$$R(M_0) = \{ M_0, M_1, M_2, M_3, M_4 \}$$

- Le graphe des marquages accessible par La séquence de franchissement S est :



- La séquence de franchissement est : $S = T_2 T_4 T_1 T_2$
- Le vecteur de franchissement (σ) associé à la séquence de franchissement S est :

$$\sigma = [1 \ 2 \ 0 \ 1]^T$$

b) RdP borné :

Un RdP marqué est *k-borné* si toutes ses places sont *k-bornées*, c'est-à-dire, quel que soit le marquage accessible M à partir du marquage initial M_0 , et quel que soit la place P considérée le nombre de jetons contenus dans cette place est inférieur ou égale à une *borne* k [29]:

$$R(Q, M_0) \text{ est } k\text{-borné} \Leftrightarrow \forall M \in R(M_0), \forall p \in P : M(p) \leq k$$

Un RdP marqué est *sauf ou binaire* pour un marquage initial M_0 s'il est *1-borné*.

Cette propriété de bornitude est nécessaire lorsqu'une place représente par exemple une ressource dont le nombre est limité, ou un stock pour lequel la capacité est limitée.

c) RdP vivant :

Un RdP est *vivant* si quelque soit le marquage M appartenant à l'ensemble de tous les marques accessibles à partir de M_0 ($R(M_0)$), il est possible de franchir toute transition du réseau en progressant à travers une séquence de franchissement. Donc, un RdP est vivant pour un marquage initial M_0 , si et seulement si [28]:

- ❖ Toutes les transitions du RdP apparaissent dans le graphe des marquages ;
- ❖ Le graphe des marquages est fortement connexe, c'est-à-dire, si et seulement si, en prenant deux nœuds (i, j) quelconque du graphe alors, il existe une séquence d'arcs de i vers j et un autre de j vers i .

$$RdP \text{ vivant} \Leftrightarrow \forall T_j \in T, \forall M \in R(M_0), \exists S/T_j \in S, \exists M' \in R(M_0) / M [S > M'$$

Un RdP *vivant* est un RdP *sans blocage*, un *blocage* est un état tel qu'aucune transition n'est validée.

Exemple N°5:

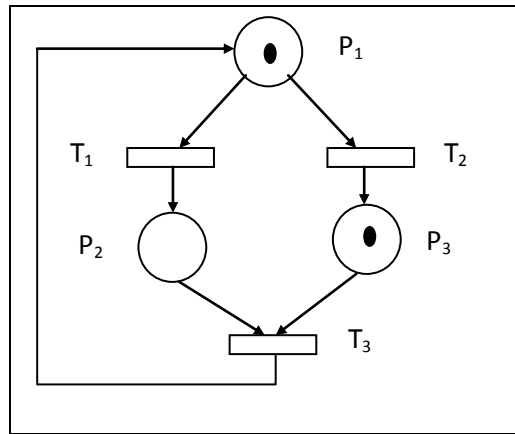
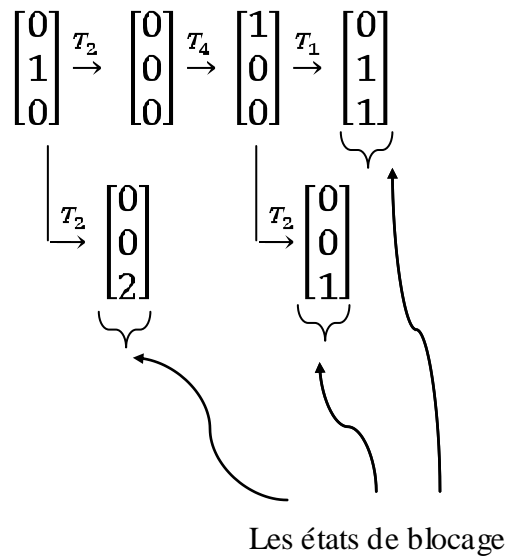


Figure II.9 : Exemple d'un RdP avec blocage.

- Le graphe des marquages :



d) RdP quasi-vivant :

Une transition T_j est quasi-vivante pour un marquage initial M_0 , s'il existe une séquence de franchissement S qui contient la transition T_j à partir de M_0 [29].

Un RdP est quasi-vivant si toutes ses transitions sont quasi-vivantes.

$$RdP \text{ quasi-vivant} \Leftrightarrow \forall P_i \in P, \forall M \in R(M_0), \exists T_j \in S : M(P_i) \geq Pre(P_i, T_j)$$

Interprétation :

Cette propriété traduit l'absence de blocage totale dans le graphe donc dans le système spécifié.

Remarque :

Si une transition T_j est quasi-vivante pour un marquage initial M_0 , elle est quasi-vivante pour un marquage $M'_0 \geq M_0$ [28].

Si une transition T_j est vivante pour un marquage initial M_0 , elle n'est pas nécessairement vivante pour un marquage $M'_0 \geq M_0$.

Si un RdP est sans blocage pour un marquage initial M_0 , il n'est pas nécessairement sans blocage pour un marquage $M'_0 \geq M_0$.

e) RdP conforme :

Est un RdP vivant et sauf : 1 jeton, sans blocage.

f) États d'accueil et RdP réinitialisable :

Un RdP possède un état d'accueil M_a pour un marquage initial M_0 si pour tout marquage accessible M il existe une séquence de franchissement permettant d'atteindre le marquage M_a [29].

$$M_a \text{ état d'accueil} \Leftrightarrow \forall M \in R(M_0), \exists S/M [S > M_a$$

Un RdP est réinitialisable (ou réversible) pour un marquage initial M_0 si M_0 est un état d'accueil.

II.1.6 RdP particuliers :**a) Graphe d'événement (GEV):**

Un graphe d'événement est un RdP où chaque place possède exactement une transition d'entrée et une transition de sortie [29].

$$\forall P_i \in P, \quad |^{\circ} P_i| = |P_i^{\circ}| = 1$$

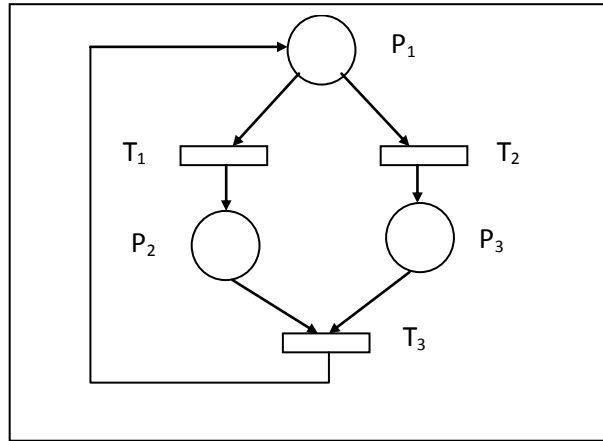


Figure II.10 : Exemple d'un graphe d'événement (GEV).

b) Graphe d'état (GET):

Un *graphe d'état* est un RdP où chaque transition possède exactement une place d'entrée et une place de sortie [29].

$$\forall T_j \in \mathcal{T}, \quad |{}^o T_j| = |T_j^o| = 1$$

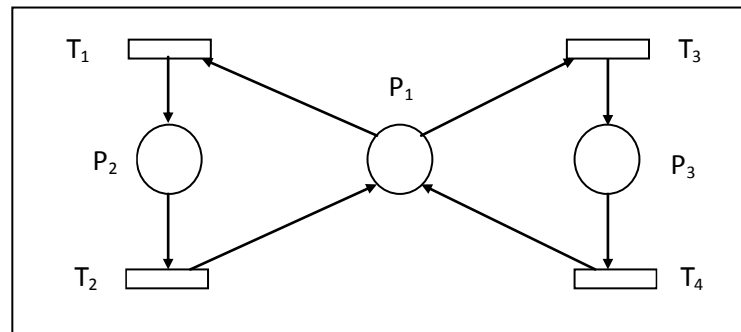


Figure II.11 : Exemple d'un graphe d'état (GET).

c) RdP avec conflit vs RdP sans conflit :

Un conflit (ou un *conflit structurel*) correspond à l'existence d'une place P_i qui a au moins deux transitions de sortie T_i, T_k [28]

On notera un conflit par : $K_s = \langle P_i, \{ T_i, T_k, \dots \} \rangle$

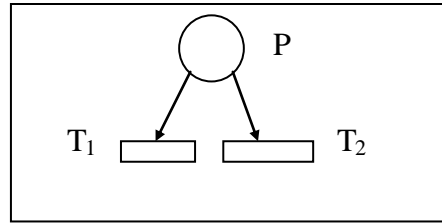


Figure II.12 : Exemple d'un conflit.

Un *conflit effectif* est l'existence d'un conflit structurel $\langle P_i, \{ T_i, T_k, \dots \} \rangle$ et d'un marquage M tels que le nombre de marques dans la place P_i est strictement inférieur au nombre des transitions de sortie de P_i validées par le marquage M (voir Figure II.13).

Lors qu'un conflit effectif se produit, il est nécessaire de choisir la transition qui va être effectivement franchie.

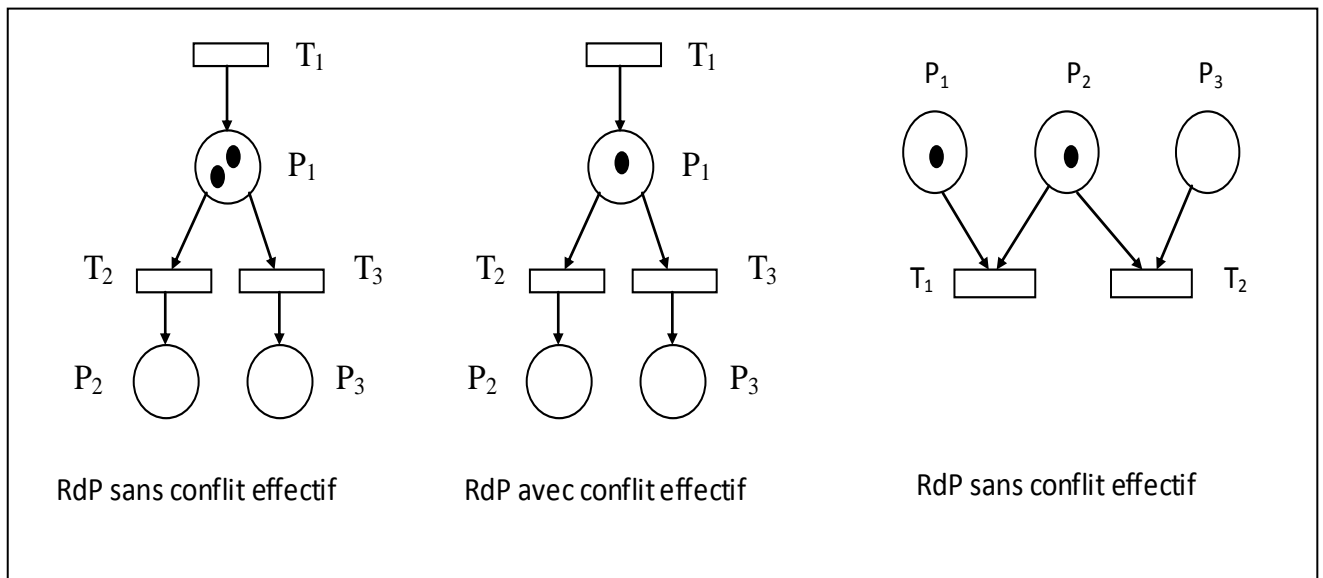


Figure II.13 : Exemples des RdP avec et sans conflit effectif.

d) RdP autonome [31] :

Un RdP *autonome* décrit le fonctionnement d'un système dont les instants de franchissement ne sont pas connus ou indiqués.

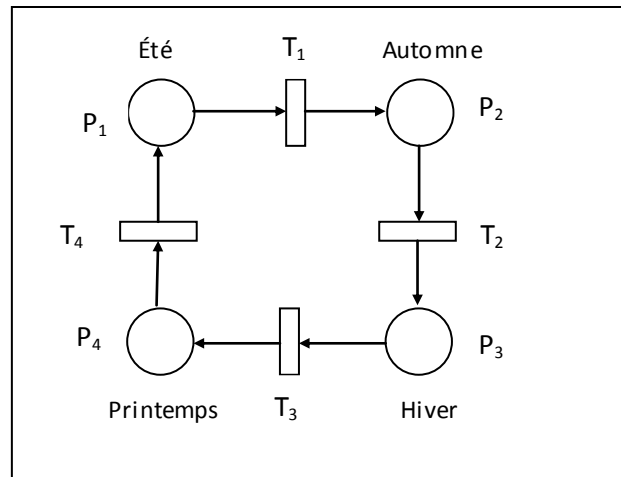


Figure II.14 : Exemple d'un RdP autonome (RdP de quatre saisons).

Le moment de passage de l'été à l'automne est inconnu.

e) RdP non autonome :

Un RdP *non autonome* décrit le fonctionnement d'un système dont l'évolution est conditionnée par des événements externes ou par le temps. Un RdP non autonome est synchronisé et/ou temporisé [31].

Exemple N°06:

Soit à commander l'allumage/extinction d'une lampe par un bouton poussoir (BP). L'appui sur le BP allume ou éteint la lampe selon qu'elle est déjà éteinte ou allumée.

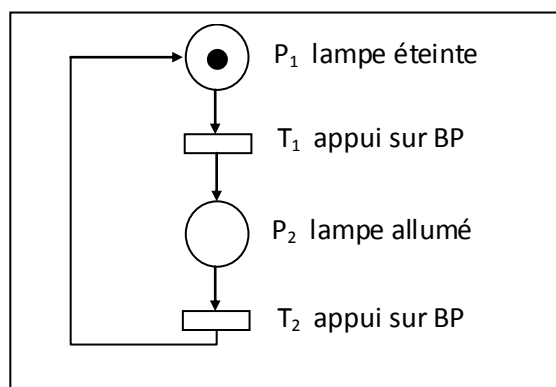


Figure II.15 : Exemple d'un RdP non autonome.

f) RdP à choix libre :

Un RdP à *choix libre* est un réseau dans lequel pour tout conflit $\langle P_i, \{ T_1, T_2, \dots, T_n \} \rangle$ aucune des transitions $\{ T_1, T_2, \dots, T_n \}$ ne possède une autre place d'entrée que P_i [31].

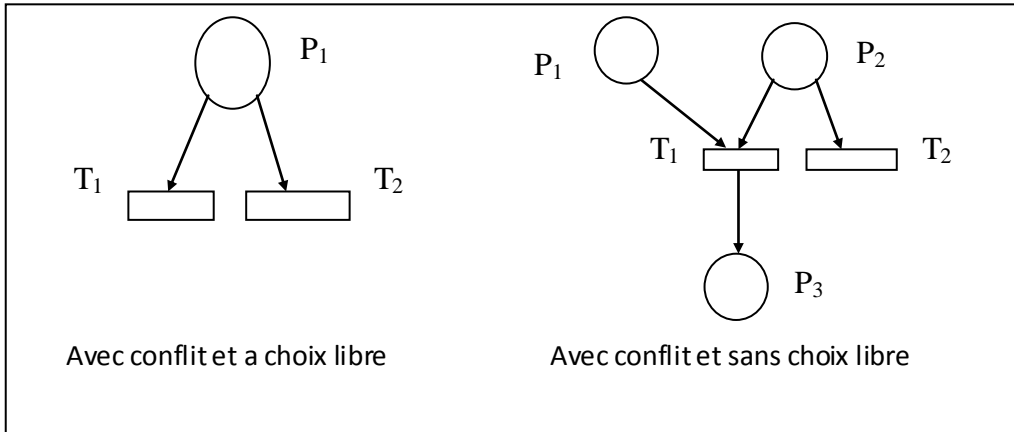


Figure II.16 : Exemple d'un RdP à choix libre.

g) RdP simple:

Un Réseau de Pétri *simple* est un RdP dans lequel chaque transition ne peut être concernée que par un conflit au plus [28].

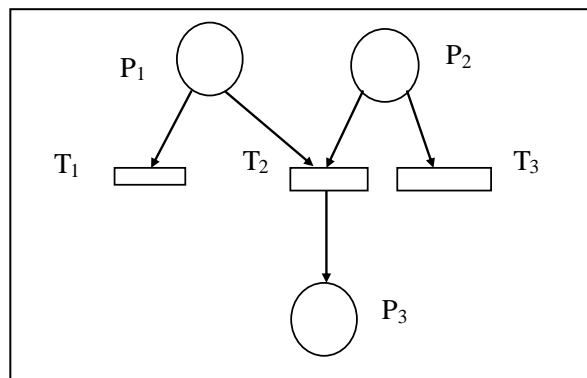


Figure II.17 : Exemple d'un RdP non simple.

h) 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 [19].

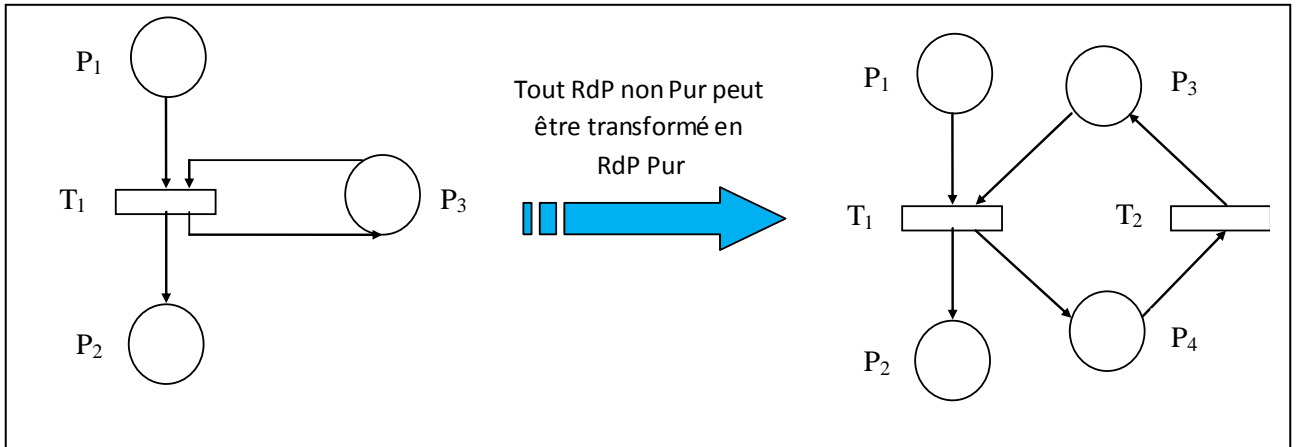


Figure II.18 : Exemple d'un RdP non pur et pur.

i) RdP à capacités :

Un RdP à capacités est un RdP dans lequel des capacités (nombres entiers strictement positifs) sont associées aux places. Le franchissement d'une transition d'entrée d'une place P_i dont la capacité est $Cap(P_i)$ n'est possible que si le franchissement ne conduit pas à un nombre de jetons dans P_i qui est plus grand que $Cap(P_i)$ [31].

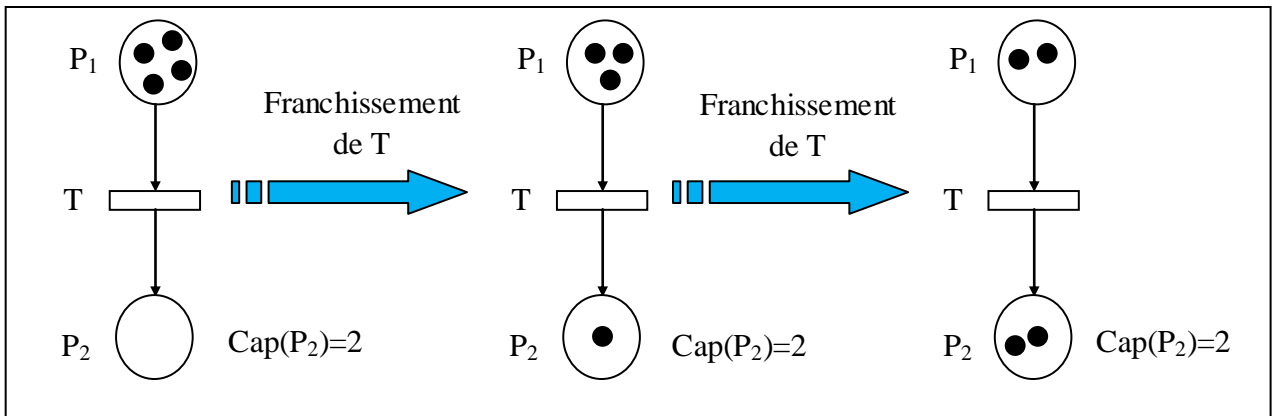


Figure II.19 : Exemple d'un RdP à capacité.

Le franchissement de T conduit à 3 jetons dans P_2 d'où T ne peut plus être franchi.

j) RdP à priorités :

Dans un tel réseau si on atteint un marquage tel que plusieurs transitions sont franchissables, on doit franchir la transition qui a la plus grande priorité [31].

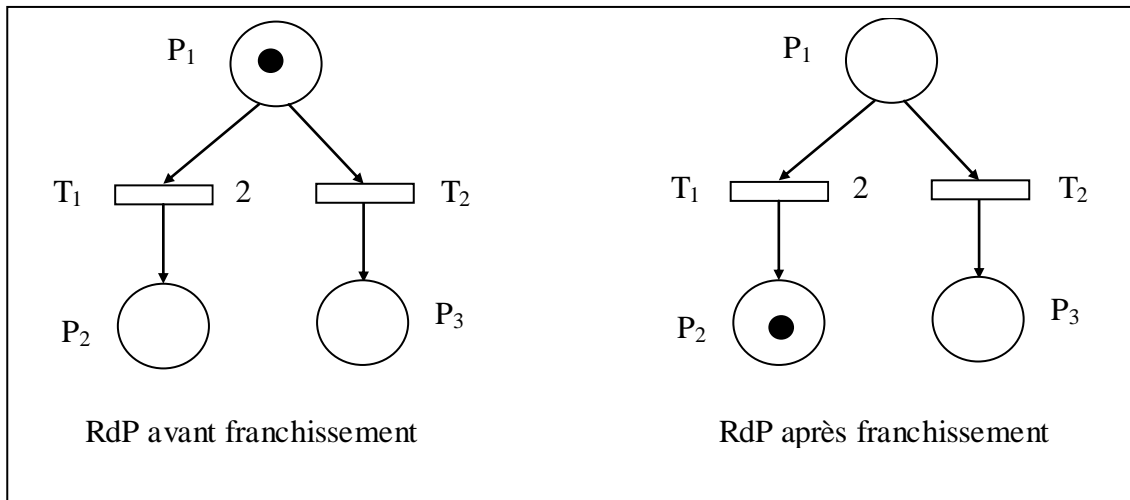


Figure II.20 : Exemple d'un RdP à priorité.

k) *RdP à arc inhibiteur:*

Un RdP à *arc inhibiteur* est un RdP dans lequel le franchissement d'une transition (arc inhibiteur entrant dans cette transition) est conditionné par l'état vide (pas de marque) de l'une de ces places d'entrée. Les RdP à arcs inhibiteurs permettent de réaliser le test de zéro. L'arc inhibiteur est représenté par une flèche avec un cercle à l'extrémité [19].

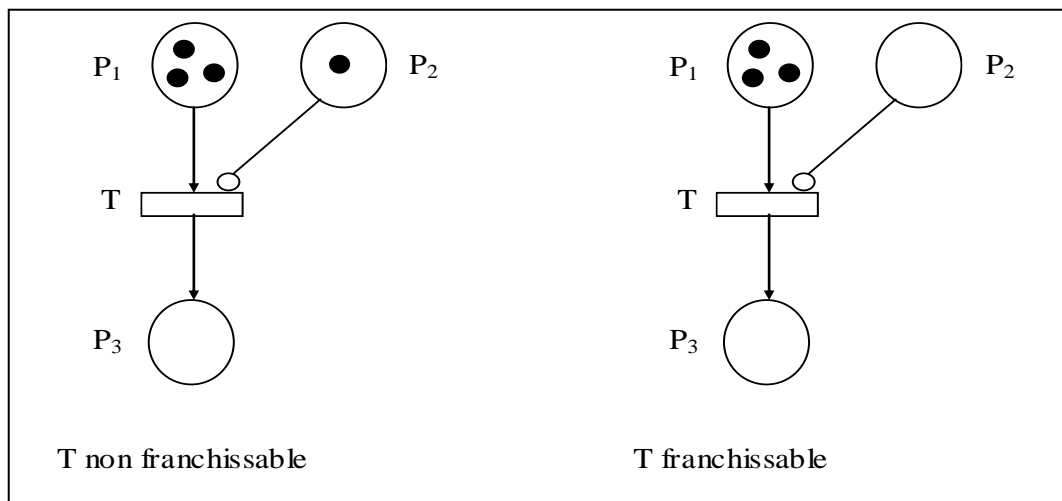


Figure II.21 : Exemple d'un RdP à arc inhibiteur.

II.1.7 Les invariants:

A partir d'un marquage initial, le marquage d'un RdP évolue par franchissements de transitions. En l'absence de blocage, le nombre de franchissements de transitions et le nombre de marquages effectivement réalisées sont illimités. Il est donc difficile d'étudier les séquences de transition et les marquages accessibles simplement par exemple en entreprenant une énumération. On est donc amené à définir des invariants caractérisant certaines propriétés des séquences de transitions et des marquages accessibles quelque soit l'évolution.

a) Notion de P -invariant (Composante conservative) [29]:

Définition :

Un P -invariant est un vecteur Y composé d'entiers non négatif solution de $Y^T W = 0$

Où W est la matrice d'incidence.

On appelle *support de Y* , noté $P(Y)$, l'ensemble des places P qui correspondent aux composantes non nulles de Y . le vecteur Y est dit *minimale*, si et seulement si, $P(Y)$ ne contient d'autre support $P(Y')$ que lui-même et l'ensemble vide.

Propriété :

Si Y est solution de $Y^T W = 0$, alors : $\forall M \in R(M_0) : Y^T M = Y^T M_0 = \text{constante}$.

Cette relation exprime la conservation du marquage des places appartenant à $P(Y)$ indépendamment de l'évolution du réseau.

Remarque :

La propriété « *composante conservative* » est indépendante du marquage initial M_0 , Par contre, la valeur de la constante dépend de M_0 .

Définition :

On appelle *composante conservative* d'un RdP le rapport du P -invariant. De plus, si l'ensemble de toutes les places P est lui-même une composante conservative, on dira que le réseau est conservatif.

b) Notion de T -invariant (Composante répétitive) [29]:

Il s'agit ici d'étudier le comportement cyclique de l'évolution d'un RdP.

Définition :

Un T -invariant est un vecteur X à composantes entiers non négatives solution de : $W X=0$.

Le support d'un T -invariant est noté $T(X)$, l'ensemble des transitions qui correspondent aux composantes non nulles de X .

Propriété :

Considérons une séquence S telle que son vecteur caractéristique U :

$$T\text{-invariant} \Leftrightarrow W U = 0 \Rightarrow M = M_0 + W U = M_0$$

Le franchissement de S nous ramène à M_0 , dans ce cas, S est une *séquence répétitive stationnaire*.

Définition :

Le support d'un T -invariant est appelé *composante répétitive stationnaire*, si l'ensemble de toutes les transitions est lui-même une composante répétitive stationnaire, on dira que le réseau est répétitif stationnaire.

$$\text{Le réseau répétitive stationnaire} \Rightarrow \exists S / M_0 [S > M_0$$

Définition :

S est une *composante répétitive* si et seulement si il existe une séquence de franchissement S tel que;

$$T(U) = S \quad \text{et} \quad W U \geq 0$$

U vecteur caractéristique de S .

La séquence S est dite *complète* si elle contient toutes les transitions du RdP.

On appelle *séquence répétitive croissante*, une séquence de franchissements S telle que

$$M_0 [S > M_0' \text{ avec } M_0' > M_0$$

On appelle *séquence répétitive décroissante*, une séquence de franchissements S telle que

$$M_0 [S > M_0'' \text{ avec } M_0 > M_0''$$

➤ **Algorithme de recherche des P-invariants (T-invariants) [29]:**

Pas N°1 :

Soit $A = I_n$

Construire la matrice $[A / B]$

N : nombre de places ; $B = W$.

Pas N°2 :

Pour chaque indice j de transition T_j :

Pas N°2-1 :

Ajoute à la matrice $[A / B]$ autant de ligne i qu'il y a de combinaisons linéaires de deux lignes, à coefficients entiers positifs, annulant l'élément (i, j) .

Pas N°2-2 :

Eliminer de la matrice $[A / B]$ les lignes i dont l'élément (i, j) n'est pas nul.

Pas N°3 :

Les P -invariants correspondent aux lignes non nulles de A .

II.2.8 méthode d'analyse :

La mise en œuvre d'un système ne peut être envisagée qu'après avoir démontré qu'il ne subsiste aucune erreur, aucune ambiguïté dans la spécification des fonctions qui le composent. Toute erreur de conception entraîne un surcroît de coût important si elle n'est détectée qu'au moment de la mise en service, lors des tests de bon fonctionnement. C'est pourquoi toute implémentation doit être précédée par une validation du projet qui passe par une *validation du modèle RdP*.

La validation est l'opération qui consiste à mettre en évidence les propriétés du système ; différentes méthodes d'analyse ont été développées pour permettre une vérification des propriétés des réseaux autonomes. Parmi les méthodes d'analyse les plus utilisées nous citons :

a) L'analyse par énumération :

L'analyse par énumération utilise un algorithme dit *graphe de couverture* et consiste, pour un RdP marqué, à construire un graphe des marquages accessibles. Si l'ensemble des marquages accessibles est fini, on pourra vérifier sur le graphe certaines propriétés telles que l'absence de blocage (RdP vivant) [20].

La méthode par énumération, fortement combinatoire, s'avère très difficile à mettre en œuvre. On assiste souvent à une explosion combinatoire du graphe des marquages, même si le nombre des places et des transitions reste faible dans RdP.

b) L'analyse structurelle (P-invariant) :

Elle s'effectue indépendamment du marquage initial et évite la construction du graphe des marquages. Nous pouvons mettre en évidence à partir des invariants les propriétés de base d'un RdP ainsi que ses propriétés spécifiques. La décomposition se fera sans énumération des marquages accessibles, puisque la résolution du système d'équation se ramène à un problème de programmation en nombre entiers [20].

Si chacune des places du réseau apparaît dans au moins un invariant et si les composantes du Y solution de $Y^T M = Y^T M_0$ sont positifs non nulles, le réseau est borné.

II.1.9 Les réseaux de Petri de haut niveau :

Un RdP ordinaires ou généralisés peut modéliser des comportements très divers ; parallélisme, partage de ressources, etc. Cependant cette modélisation souffre d'une limitation. En effet, la modélisation des systèmes réels mène parfois à des RdP de taille ingérable, ce qui diminue leur lisibilité et rend leur manipulation et même leur analyse difficile (problème d'explosion combinatoire).

Pour apporter des solutions à cette limitation et augmenter la puissance de modélisation, des extensions du modèle RdP dites des RdP de haut niveau ont été proposées dans lesquelles d'autres aspects ont été pris en compte, tel que, la modularité, la colorisation, la temporisation, et autres.

a) Réseaux de Petri colorés :

D'après K.JENSON [12], les *RdP colorés* sont des RdP dans lesquels les jetons portent des couleurs. Chaque couleur correspond à une information bien définie associée au jeton. Deux principales raisons justifient l'appel aux RdP colorés :

- Il ne nécessite pas une redéfinition chaque fois qu'une modification est apportée au système étudié.
- Il s'utilise pour éviter le problème de taille importante des RdP *ordinaires* dans le cas où des entités différentes présentent des comportements similaires ce qui condense le modèle.

L'intérêt des réseaux colorés est de pouvoir décrire des dynamiques relativement complexes avec un petit nombre de places et de transitions. En effet la notion de couleur et les expressions d'arcs permettent de rendre plus compacte la structure du réseau en réduisant de manière importante le nombre de places et de transitions nécessaires à la modélisation du système.

Exemple N°07 :

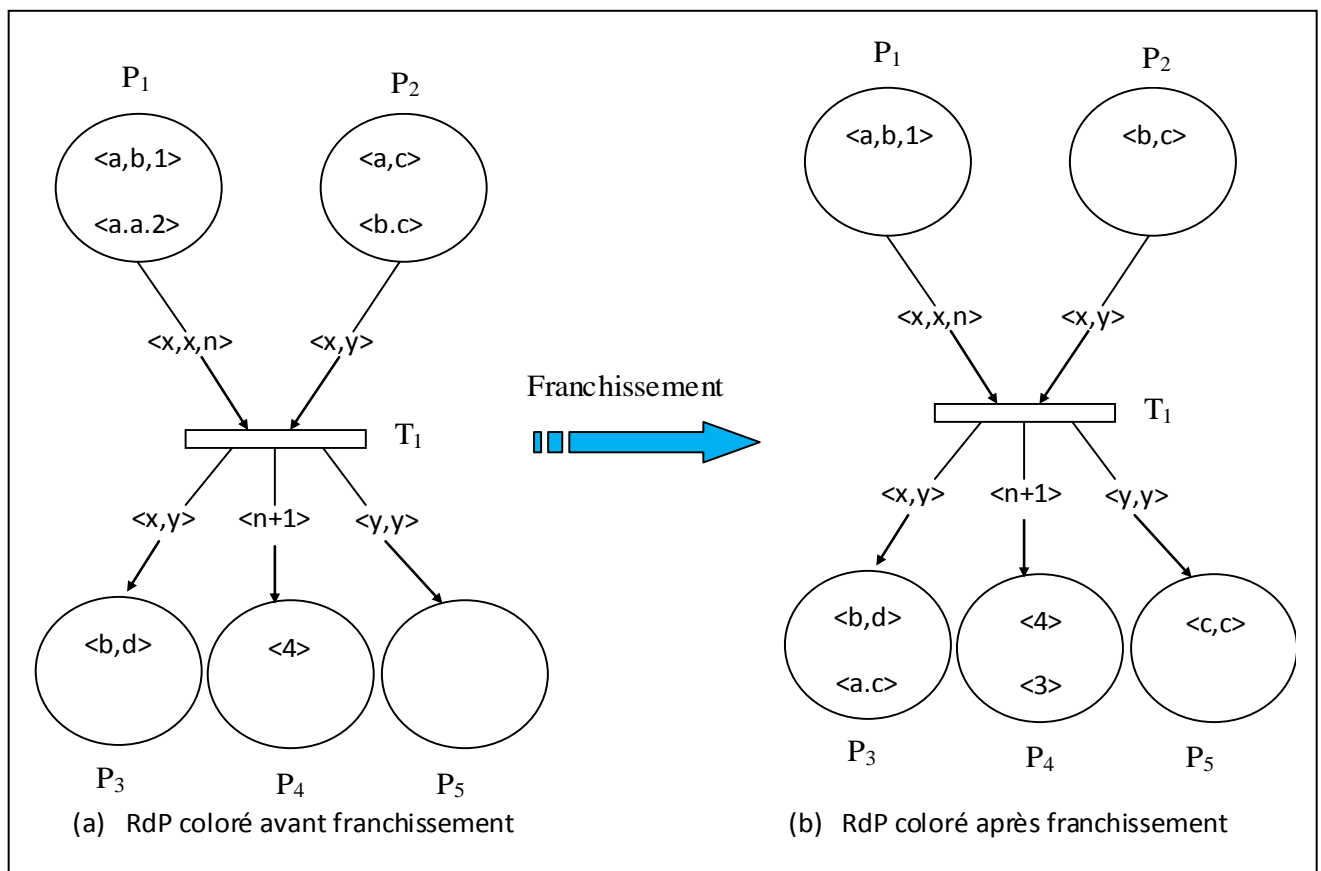


Figure II.22 : franchissement d'une transition dans un RdP coloré [17].

Cet exemple montre le mécanisme de franchissement dans un RdP coloré. La transition T_1 ne peut être franchie que s'il existe une marque de la forme $\langle x, x, n \rangle$ sur la place P_1 et une marque de forme $\langle x, y \rangle$ sur la place P_2 . dans l'exemple, cela est possible puisqu'il existe une marque $\langle a, a, 2 \rangle$ sur P_1 et une marque $\langle a, c \rangle$ sur P_2 à la situation (a). Ces marques sont alors supprimées des places d'entrées, et de marques correspondant aux indications des arcs sortants sont produites sur les places de sorties. Dans l'exemple, les marques $\langle a, c \rangle$, $\langle 3 \rangle$ et $\langle c, c \rangle$ sont respectivement ajoutées à la liste des marques des places P_3 , P_4 et P_5 (b).

b) Réseaux de Petri à temps discret :

Les RdP à *temps discret* peuvent être classés en deux familles, les RdP *temporisés* qui permettent une modélisation du temps sur un espace entier, Leur comportement est déterministe, et RdP *stochastiques* qui ont un comportement non déterministe, mais stochastique, ils permettent de modéliser les processus aléatoires.

Réseaux de Petri temporisés :

Les *RdP temporisés* (RdPT) introduits par C.RAMCHANDANI et P.M.MERLIN [20] étendent les RdP avec des intervalles temporels associés aux transitions, spécifiant les bornes de délai de tir des transitions. Ils permettent la modélisation des systèmes avec des contraintes temporelles tel que : les protocoles de communication, certains cas des systèmes à temps réels, etc.

L'extension temporelle peut s'envisager sur les places, sur les arcs ou sur les transitions.

➤ Merlin définit le *RdP T-temporisé* comme un RdP avec deux valeurs du temps a et b associées aux transitions ; avec $(0 \leq a \leq b)$ et b peut être illimité, spécifiant les bornes de délai de franchissement des transitions. En considérant que la transition T est devenue sensibilisée pour la dernière fois à l'instant θ , alors T ne peut être franchie plus tôt qu'à l'instant $\theta + a$ et ne doit pas être franchie plus tard qu'avant (ou exactement à) $\theta + b$. L'intervalle $[a, b]$ est le temps pendant lequel les jetons des places en entrée ne sont plus présents (ils sont réservés), mais pendant lequel les jetons produits ne sont pas encore visibles dans les places de sortie.

➤ Sifakis propose le *RdP P-temporisé* où toute place P_i se voit associé une constante $r_i \geq 0$ appelée *délai*. L'arrivée d'une marque, à l'instant t , dans une place P_i commande l'action associée à cette place, la durée de l'action est alors égal à r_i . on dira que la marque reste indisponible pendant r_i , durée de l'action, ce qui signifie qu'elle ne peut quitter cette place avant l'instant $t+r_i$. lorsque le délai r_i est écoulé, la marque devient disponible et peut intervenir dans la sensibilisation des transitions aval [20].

On distinguera donc, dans tout marquage, un marquage M_d (constitue des marquages disponibles) et un marquage M_i (constitue des marquages indisponibles), on aura :

$$M = M_d + M_i \quad (II.8)$$

Lorsqu'une transition est franchie, les marques M_d dans les places amont sont retirées selon les règles de franchissement définies pour les RdP généralisés ; les places aval reçoivent des M_i selon les mêmes règles de franchissement. Le tir d'une transition à une durée nulle.

➤ Walter a introduit le concept de *RdP arc-temporisé* dont la définition inclut celle des RdP *P-temporisés* et des RdP *T-temporisés*. Dans un RdP arc-temporisé, le délai r_{ij} est associé à chaque arc reliant une transition T_j à une place P_i . le jeton qui arrive en P_i reste indispensable pendant une durée de temps qui dépend de la transition amont tirée. Les RdP arc-temporisés permettent de réduire la complexité des modèles.

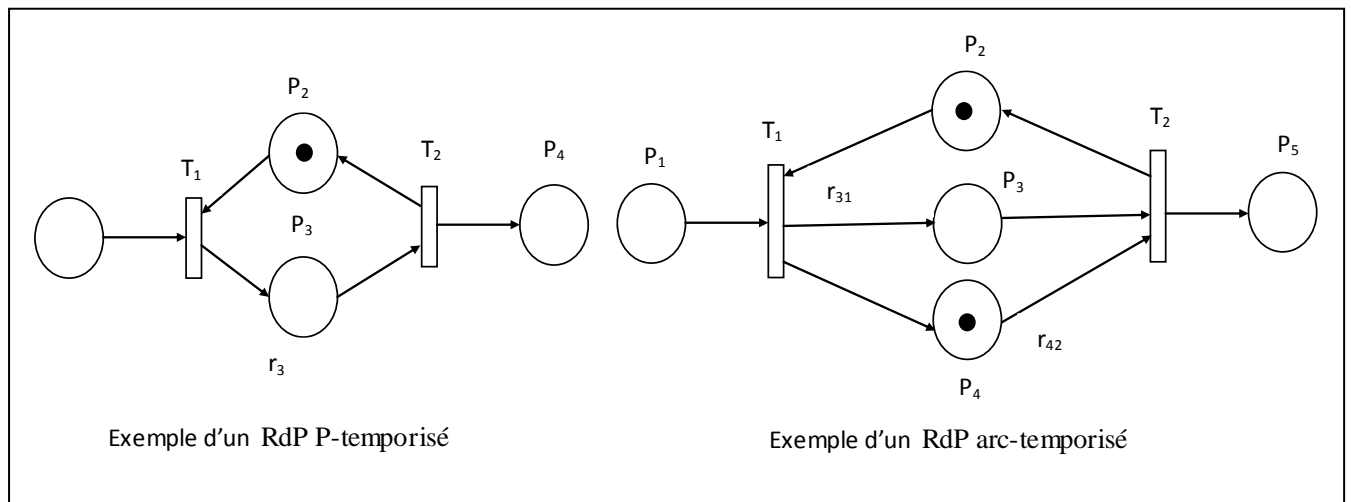


Figure II.23 : Exemples des RdP temporisés.

🚦 Réseaux de Petri stochastiques :

Ce type de modèle s'utilise pour traiter le problème des événements en conflit dans les systèmes discrets. Ce problème se pose lorsqu'il y a plusieurs transitions qui peuvent être franchies en même temps. Dans les RdP stochastiques, des poids ou des probabilités sont associés aux transitions pour régler le problème.

c) Réseaux de Petri hiérarchiques :

D'après Hudert, le principe de RdP hiérarchiques, est de construire un modèle en combinant un certain nombre des réseaux appelés '*page*' en un seul réseau. L'objectif est de réduire la complexité d'un modèle en permettant la modélisation modulaire [26].

Un *RdP hiérarchique* se compose de plusieurs modules '*pages*', où chaque module représente un sous-réseau de Pétri. Un module dit principal référence tous les autres modules secondaires. Le référencement d'un module se fait au moyen d'une transition ou une place de substitution portant le

nom du module en question. L'utilisation des RdP hiérarchiques est préconisée pour la modélisation des problèmes de grande taille.

d) Réseaux de Petri récursifs :

D'après Haddad, les RdP récursifs fournissent un outil pour couvrir certaines limites des RdP ordinaires, colorés hiérarchiques. Ce type de RdP a été introduit la première fois pour modéliser la dynamique dans les environnements multi-agents. Le point fort de ce formalisme est qu'il permet de construire des modèles compacts. Ceci permet de modéliser le fait d'alterner planification et exécution, une alternative naturelle dans les SMA [12].

Les auteurs de ce formalisme suggèrent que ce formalisme couvre les limites suivantes :

- Dans les RDP ordinaires, le franchissement des transitions est instantané, alors qu'une action peut prendre un certain temps.
- Les RdP de haut niveau, les transitions sont des actions élémentaires, alors qu'une action peut être comme une abstraction d'une activité plus complexe (un plan).
- Dans les RdP hiérarchiques ; quand une transition est une abstraction, il n'y a pas de fin bien définie de son franchissement.

Un RdP récursif a une même structure qu'un RDP ordinaire. Cependant les transitions sont divisées en trois catégories : *i)* des transitions élémentaires, *ii)* des transitions abstraites et *iii)* des transitions finales. À chaque transition abstraite est associé un marquage initial.

CONCLUSION :

Nous avons présenté dans ce chapitre le formalisme de RdP. Les RdP sont particulièrement bien adaptés à la description des aspects dynamiques ou comportementaux d'un système. Des concepts tels que la concurrence ou la synchronisation entre interactions s'expriment aisément dans le cadre de ce formalisme.

L'avantage des RdP par rapport à la grande majorité des autres représentations, est leur aptitude à donner des informations concernant l'exécution du système. De la théorie mathématique, on peut déduire un certain nombre de propriétés (invariants, vivacité, existence d'un état d'accueil, ...etc.) qui sont autant d'informations à la disposition du concepteur pour se faire une idée du comportement de son système.

Chapitre III.

*Modélisation
des Systèmes Multi-Agents
par
les Réseaux de Petri*

INTRODUCTION :

La modélisation consiste à construire des modèles pour des phénomènes ou des systèmes. Un modèle est vu comme une représentation abstraite d'une portion ou de tout le système. Cette représentation doit être explicite et simple à manipuler par rapport au système réel. Le mieux est de pouvoir analyser les modèles obtenus pour vérifier les propriétés du système.

Un des intérêts majeurs de la modélisation multi-agent est qu'elle s'inscrit dans un courant méthodologique très actuel en résolution distribuée et concurrente de problèmes : la modélisation est basée sur les entités. Cependant, les simulations d'un système multi-agent (SMA) correspondent à des évolutions particulières du modèle, c'est-à-dire à des chemins particuliers dans l'espace des états accessibles du système. A elles seules, elles ne renseignent pas sur la non existence de situations de blocage que nous cherchons à éviter, et ne permettent pas de répondre à certaines questions que l'on se pose sur la gestion de ressources renouvelable, notamment les questions sur la viabilité du système en fonction des conditions initiales et les hypothèses d'évolution. Il s'est donc avéré nécessaire de procéder à une étude formelle du système.

Les réseaux de Petri (RdP) sont à la fois un formalisme mathématique et un langage graphique pour modéliser les aspects dynamiques ou comportementaux d'un système complexe. Ils s'adaptent bien à la description des systèmes dans lesquels des problèmes de concurrence, de synchronisme et de parallélisme se présentent ; constituent des représentations faciles à comprendre et donc à manipuler, à la fois pour la création des modèles et pour leur analyse ; Ils sont particulièrement adaptés pour décrire les relations qui existent entre des circonstances (ressources, données et contraintes) et les actions, en fin, les RdP possèdent, des méthodes de validation basées sur un grand nombre de résultats théoriques et d'outils de support, utilisables pour étudier les comportements fonctionnels, temporels et stochastiques.

Vue l'intérêt que possèdent les RdP, et les caractéristiques des systèmes multi-agents (distribution, parallélisme, communication,...etc.), il semble très logique que les RdP constituent un bon candidat comme formalisme pour modéliser des systèmes modélisés au préalable par les SMA.

Dans le présent chapitre, nous présentons une modélisation des systèmes multi-agents à l'aide des réseaux de Petri. En premier lieu, on s'intéresse à la modélisation d'un agent, en suite ces interactions avec d'autres agents, a la fin de ce chapitre, nous présentons un mécanisme d'allocation de tâches dans un contexte multi-agent à base des modèles RdP des plans d'actions.

Avant d'entamer cette présentation il est important de parler de la notion d'un système et de son modèle.

III.1 Notions générales sur les systèmes et modèles:

Un *système* est une portion de la réalité définie par une frontière organisée en fonction d'un but. En général, un système est constitué par un ensemble d'éléments en interaction dynamique. Il possède des *entrées* et des *sorties*. Elles sont caractérisées par des *variables* de flux.

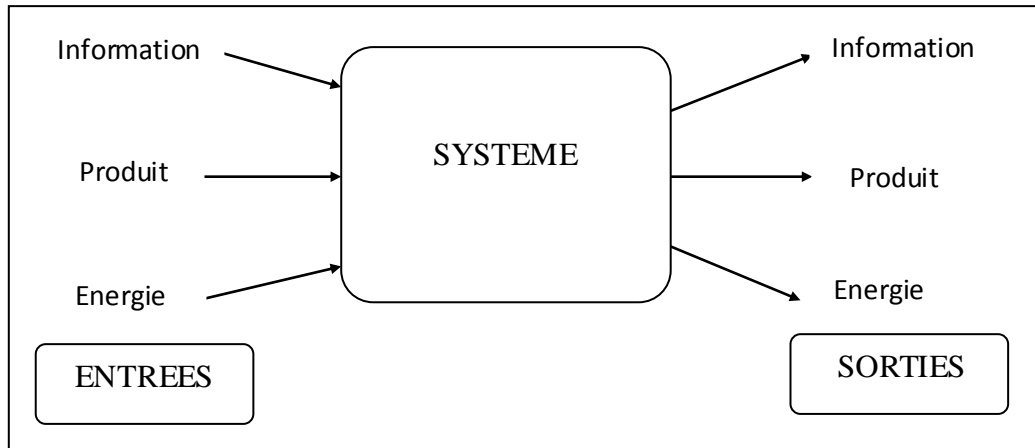


Figure III.1 : Système.

Un *modèle* est une représentation, souvent en termes mathématiques, des relations qui existent entre les variables de sortie et les variables d'entrée d'un système. Cette représentation nécessite souvent l'introduction de variables supplémentaires appelées *variables d'état*. A un instant donné, la valeur des variables d'état caractérise un système : leur connaissance avec celle des valeurs dans le futur des variables d'entrée permet à l'aide du modèle de déterminer les valeurs des variables de sortie dans le futur [28].

Un modèle serait *idéal* s'il est capable de reproduire de façon exacte le comportement du système. Cela signifie que si, à un instant donné, on connaît la valeur des variables d'état du système et que si on connaît la valeur des variables d'entrée dans le futur (et le présent), les relations mathématiques qui constituent le modèle permettent de déterminer exactement les valeurs des variables de sortie qui peuvent être observées expérimentalement. Un des principes fondamentaux des Sciences de l'Ingénieur est qu'il est impossible d'obtenir le modèle idéal d'un système. Au mieux, on peut obtenir un "bon" modèle, c'est-à-dire, un modèle qui permette de déterminer de façon approchée les valeurs des variables de sortie qui peuvent être observées expérimentalement.

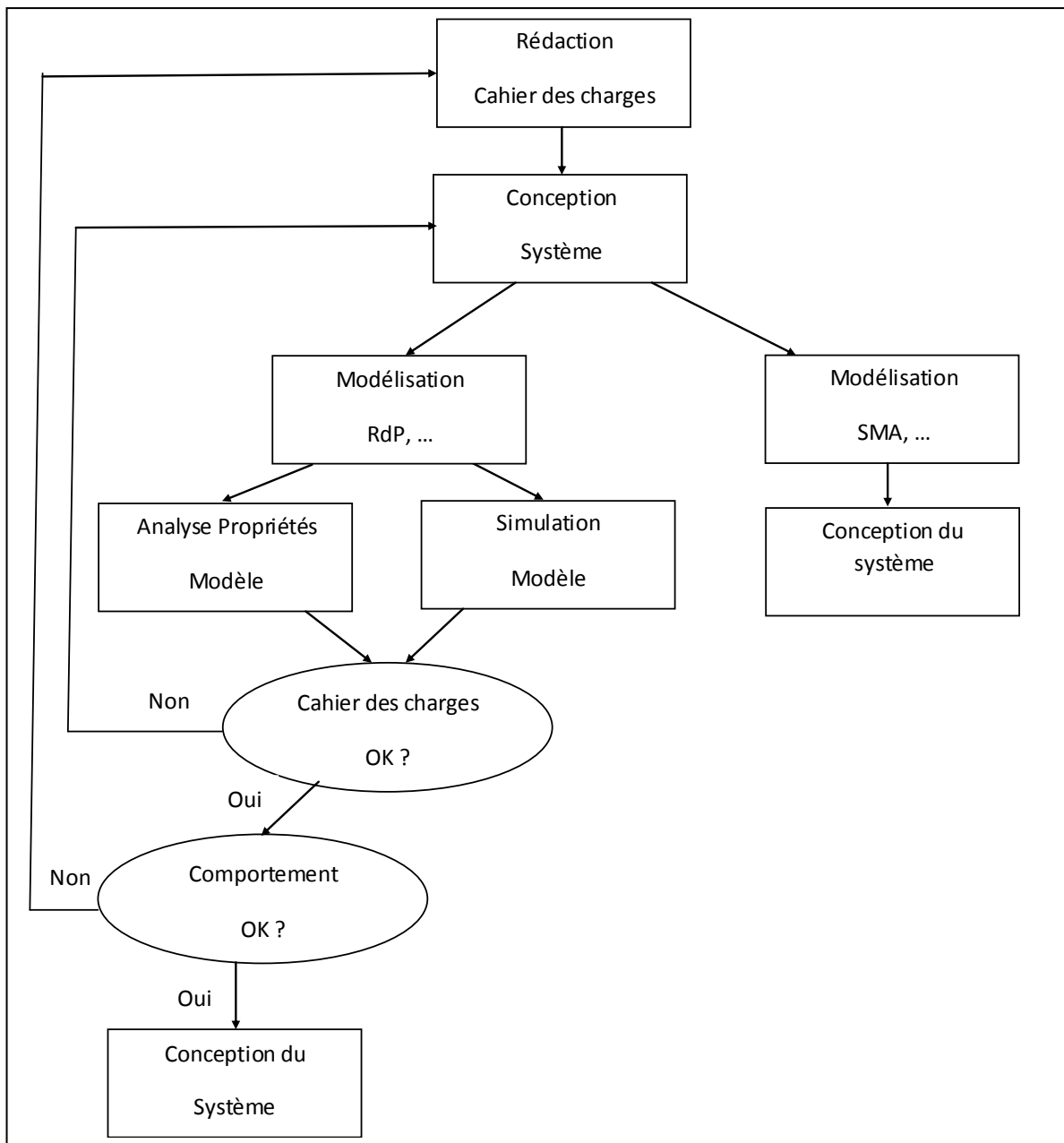


Figure III.2 : Rôle de la modélisation dans la conception d'un système.

III.2 Modélisation d'un Système Multi-Agent par un Réseau de Petri :

La modélisation des SMA par réseaux de Petri (RdP) dans ce contexte se fait en trois étapes: premièrement on écrit le système en termes de réseau, pour obtenir un modèle en réseau ; deuxièmement on analyse le modèle obtenu, pour en déduire des propriétés comme l'absence de blocage, existence d'une solution, etc. Finalement, on fait la révision des propriétés obtenues pour montrer si le système est bon (ou pour valider le système).

Les RdP possèdent plusieurs extensions (RdP colorés, RdP temporisés, RdP récursives...etc.). Le choix du type de RdP à utiliser dépend principalement de ce qu'on va modéliser et analyser (quels aspects, quelles propriétés). En effet, plusieurs solutions existent, donnent ainsi au concepteur une grande flexibilité de choix.

En analysant le paradigme SMA, nous pouvons envisager différents usages possibles des RdP. En effet, on peut modéliser par RdP :

- Le concept agent et les interactions intra-agent.
- Le SMA (les agents par les transitions, ou par les places).
- Les plans d'action des agents.
- Les interactions entre agents.
- ...etc.

Dans ce travail, on s'intéresse à la modélisation par RdP :

- De l'entité agent.
- Des interactions entre agents.
- Des plans d'actions des agents.

III.2.1 Modélisation des agents :

Les systèmes multi-agents prennent comme point de départ non pas le raisonnement mais le comportement des agents, les actions qu'ils accomplissent dans le monde et les interactions qui en découlent. Cependant, le réseau de Petri est le modèle le plus développé qui permette de préciser et de décrire le comportement des agents.

➤ Modélisation de l'environnement :

Avant de passer à la modélisation d'agents, il faut d'abord passer par un problème clé, l'environnement. Il est important de présenter une modélisation précise de l'environnement car celui-ci constitue en effet une partie essentielle des SMA.

L'environnement produit des influences, qui peuvent être exprimé par des demandes d'action à l'agent, et celui-ci répond en renvoyant des éléments consommés ou perçus. Il est possible de modéliser l'environnement par un composant qui réunit toutes les perceptions de cet environnement, qu'on peut exprimer par une transition dans la théorie des RdP. $T_{initiale}$ correspond à l'ensemble des perceptions et T_{finale} correspond à l'ensemble des actions.

Selon Ferber [17], un agent est caractérisé par son architecture et son comportement. Plusieurs types (cognitif, réactif, hybride), plusieurs architectures, et plusieurs vues de l'agent existent (agent comme une continuité de l'objet, vue modulaire,...etc.). Par conséquent, on peut envisager différentes façons d'utiliser les RdP.

En s'intéressant aux agents réactifs, l'architecture réactive la plus connue et la plus influente est l'architecture de subsomption proposée par Brooks [12] qui décompose un agent en modules verticaux $\{M1, M2, \dots, Mn\}$, chacun d'eux n'étant responsable que d'un type de comportement très limité.

Le fonctionnement d'un agent réactif est décrit par un ensemble de règles de comportement (comme nous l'avons déjà vu dans la section (I.1.3-a)). Une règle de comportement est semblable à

une règle de production : *Si condition alors action*. La condition correspond à une perception de l'environnement, et l'action à une action possible d'un module de compétence. Chaque module génère des réponses en fonction des stimuli reçus. C'est un système conditions-événements qui peut être représenté par places, transitions et jetons.

La Figure III.3 montre le modèle RdP d'un module de comportement. Le modèle RdP utilisé est de type RdP colorés. Les jetons qui peuvent être dans la place stimuli portent des informations sur l'extérieur. Une condition d'activation est associée à la transition T_1 , L'arc qui relie la place stimuli à la transition T_1 porte une condition sous forme d'une description de marque. Lorsque la place stimuli contient une marque correspondant à cette description, l'arc sera validé. La transition T_1 ne sera franchie que si l'arc à son entrée est validé et sa condition est vraie. Les deux arcs reliant la place P_2 et la transition T_2 peuvent être remplacés par un seul arc dit arc de test (Figure III.4).

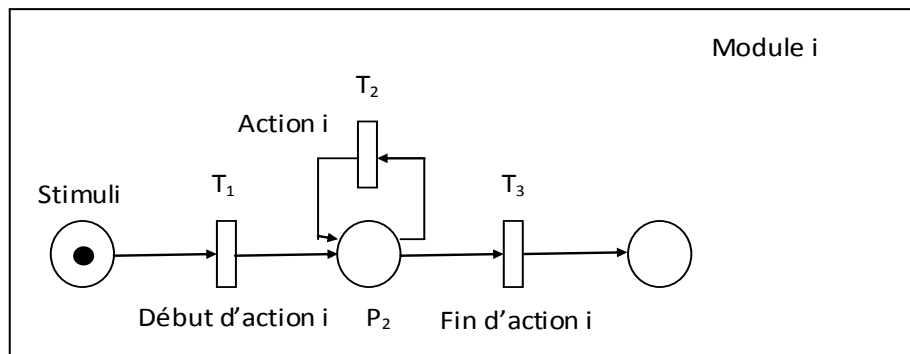


Figure III.3 : Modèle RdP d'un Module.

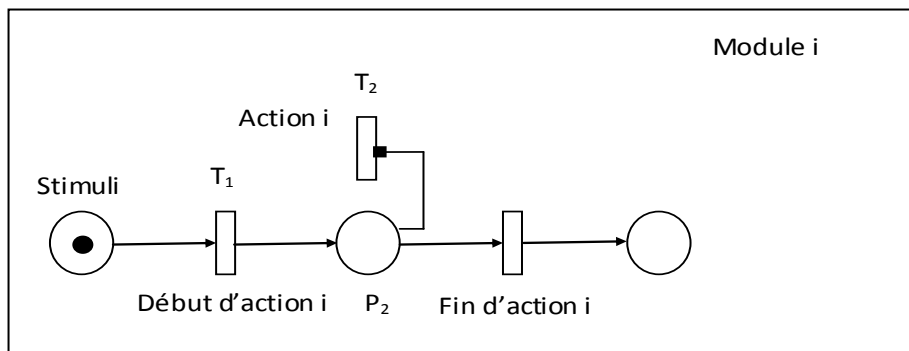


Figure III.4 Modèle RdP d'un Module avec un arc de test.

Les modules $\{M1, M2, \dots, Mn\}$ effectuent leurs tâches en parallèles. La coordination de ces comportements élémentaires repose sur le mécanisme d'inhibition. En effet, l'activation d'un module supérieur empêche le fonctionnement des modules inférieurs. Cela peut se faire par l'introduction d'arcs inhibiteurs qui testent l'absence de marques dans les places P_{2i} des modules supérieurs, comme le montre la Figure III.5.

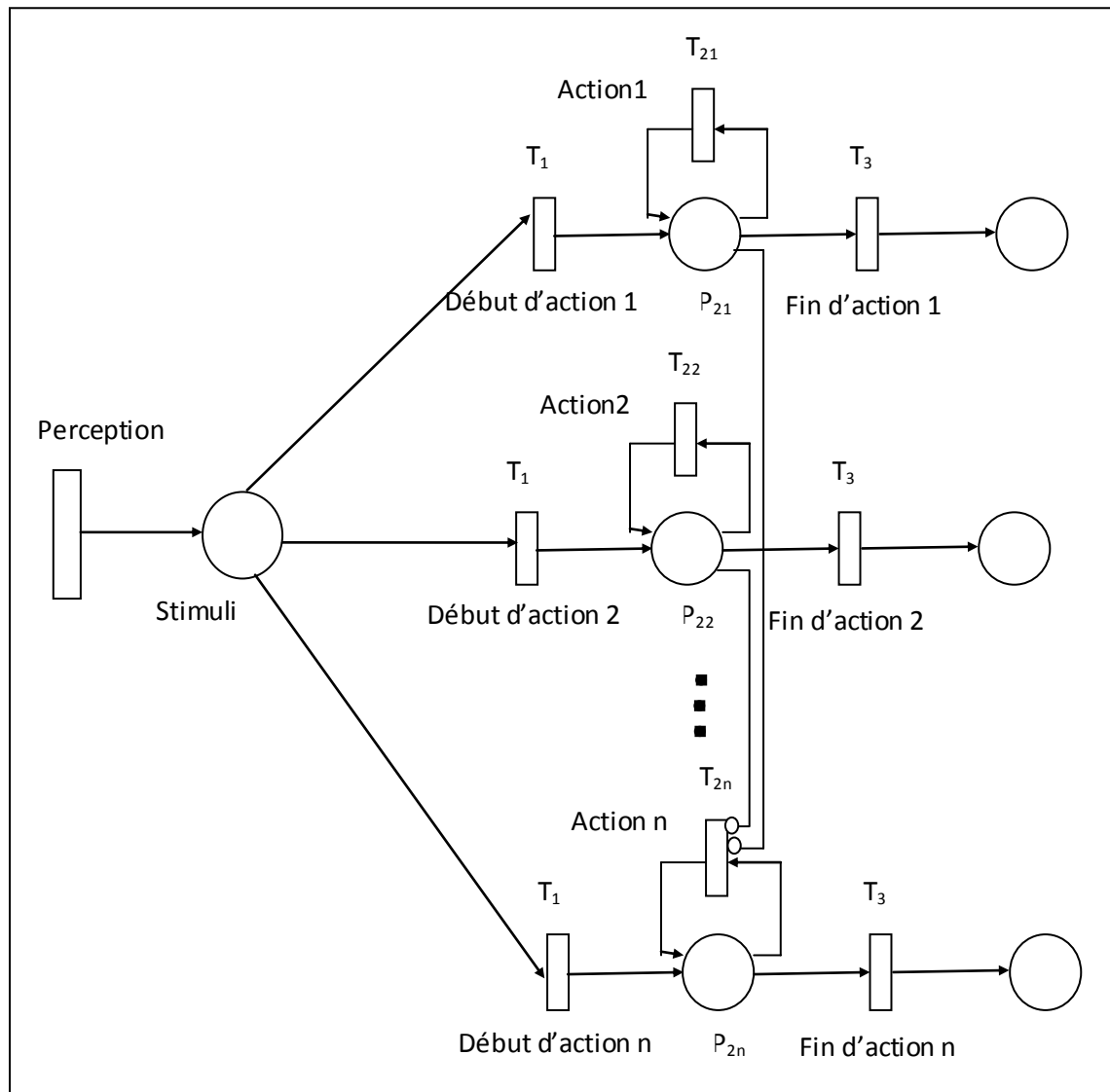


Figure III.5 : Modèle RdP d'architecture de subsomption.

Selon le marquage de la place stimuli, une transition ou plusieurs peuvent être validée(s). Pour enlever l'ambiguïté du réseau, des priorités statiques peuvent être associées aux arcs reliant la place stimuli aux différentes transitions.

III.2.2 Modélisation des interactions entre agents :

L'interaction entre les agents joue un rôle très important dans les SMA. Elle est ainsi l'un des aspects clés de ce domaine. Elle offre un moyen pour assurer la coopération et la négociation entre agents. Sans interaction (ou communication), l'agent n'est qu'un individu isolé, sourd et muet, renfermé sur sa boucle perception-délibération-action.

Dans un SMA, L'interaction se conforme souvent à un enchaînement de messages spécifié à l'avance. Les règles qui régissent l'échange de messages forment le protocole d'interaction.

III.2.2.1 Modélisation par RdP ordinaires :

Les RdP ordinaires peuvent être utilisés pour modéliser les interactions entre agents. Une telle modélisation consiste à représenter les agents jouant le même rôle (ayant le même comportement vis-à-vis du protocole) par des jetons dans les places qui représentent les états possibles de ces agents. Les messages seront représentés par des jetons dans des places. La Figure II.6 montre un envoi de message m par un agent A vers un agent B, l'action d'envoi de message est représentée par la transition T.

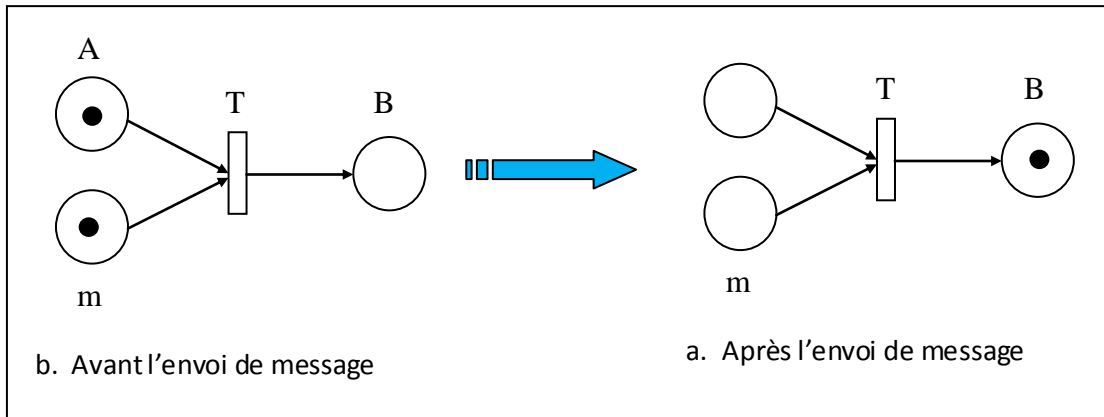


Figure II.6 : Modélisation d'envoi de message par RdP ordinaire.

La Figure III.7 montre l'envoi sélectif de messages (ou exclusif). Si le protocole est dans l'état A, deux messages peuvent être envoyés (soit m_1 soit m_2), le protocole passe à l'état B_1 après l'envoi de m_1 , ou à l'état B_2 si m_2 a été envoyé. Pour garantir qu'une seule transition (soit T_1 , soit T_2) sera franchie, la place A doit contenir un seul jeton. Les transitions T_1 , T_2 représentent respectivement l'envoi de message m_1 et m_2 .

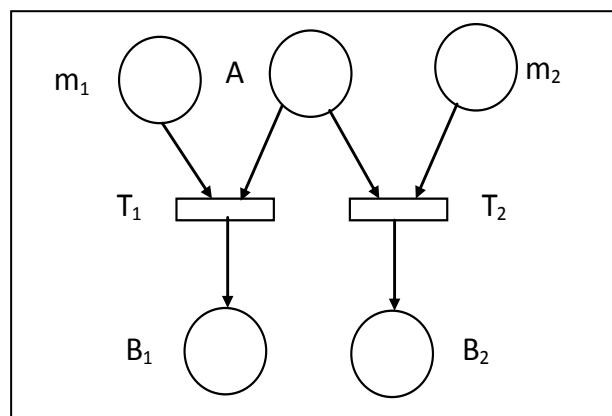


Figure III.7: Envoi Ou exclusif de message (m_1 , m_2).

Exemple :

La Figure III.8 montre le modèle RdP des interactions entre un agent voyageur (AV) et un agent interface (AI) qui joue le rôle d'interface entre l'agence de voyage et le voyageur. Trois messages possibles dans cette communication :

- 1) Le voyageur interroge le système sur les vols disponibles avec certains arguments tels que ; date de départ, destination,...etc. (message m_1).
- 2) Le système peut retourner comme réponse : vols disponibles avec leurs numéros de vol (message m_2).
- 3) ou pas de vol disponible en indiquant la raison (message m_3).

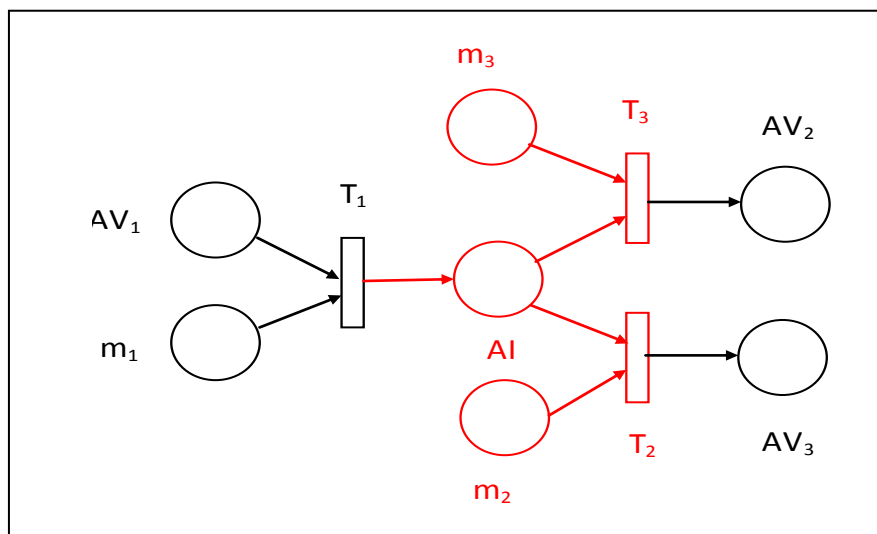


Figure III.8 : RdP des interactions entre agent voyageur et agence de voyage.

Remarque :

La présence d'un seul jeton dans la place AI, garantit qu'un seul message sera envoyé (m_1 ou m_2).

Discussion :

En fait, les RdP ordinaires sont plus appropriés pour modéliser les conversations un à un (un agent voyageur et un agent interface). Cependant dans un SMA une conversation peut se dérouler entre plusieurs agents en même temps. Graphiquement on peut modéliser de tels protocoles par des RdP ordinaires. L'inconvénient est que l'on perd l'individualité des agents du fait que les jetons sont uniformes. De plus les messages sont différents et doivent être distingués les uns des autres. Les RdP ordinaires deviennent alors de tailles très importantes. Pour remédier à cette insuffisance, on utilise les extensions des RdP, par exemple le RdP coloré.

III.2.2.2 Modélisation par RdP colorés :

L'idée sous-jacente des RdP colorés est de pouvoir regrouper des structures similaires de comportements en gardant la même structure graphique (places, transitions et arcs) et sans perte d'information. Comme nous l'avant vue dans le chapitre II, dans un RdP colorés les jetons ne sont pas uniformes mais portent de couleurs.

a) Représentation structurelle :

Dans le modèle RdP coloré d'un protocole d'interaction :

- L'envoi de message est modélisé au moyen d'une place partagée PI. Cette dernière est vue par l'émetteur A comme place de sortie et par le récepteur B comme place d'entrée.
- Chaque participant à l'interaction est représenté par une suite de type Place-Transition-Place (Figure III.9), dont :
 - Les places représentent les états des agents (avant et après l'envoi ou la réception de message).
 - Les transitions représentent les actions d'envoi et de réception de message ou un traitement (prise de décision, exécution d'une action).
- Les types définis:
 - Type Agent : ensemble de tous les agents.
 - Type acte : ensemble des actes de communications utilisés dans le protocole, {propose, refuse, ...}.
 - Type message : ensemble de message échangés lors de l'interaction. Un message est un élément complexe < Agent émetteur, Agent récepteur, acte > (ce type peut contenir d'autres informations).

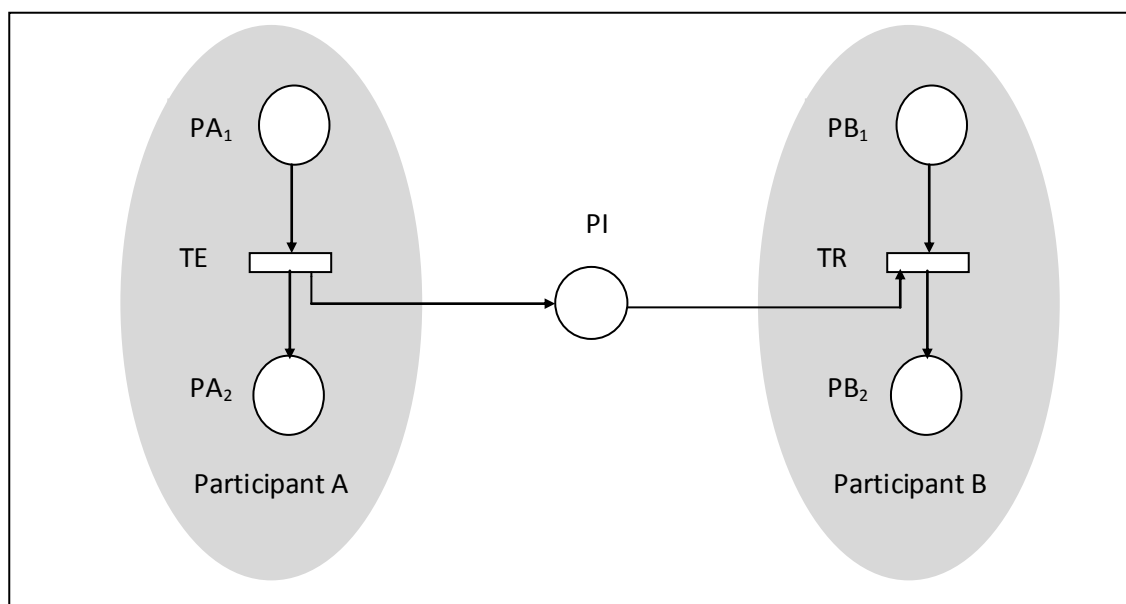


Figure III.9 : Représentation d'un échange de message par RdP colorés.

Remarque :

Lorsqu'un cercle grisé apparaît dans une figure, cela indique que seule une partie du RdP a été représentée.

b) Sémantique opérationnelle du modèle :

La définition précédente spécifie la syntaxe du modèle RdP colorés d'un échange de message. Nous en donnons maintenant une sémantique opérationnelle du modèle.

L'échange d'un message en RdP colorés d'un agent A vers un agent B se fait en trois étapes comme suit :

1) *Émetteur A*: la présence d'un jeton dans la place PA_1 signifie le début d'envoi. Le franchissement de la transition TE explique l'envoi lui-même et la présence d'un jeton dans la place PA_2 indique la fin d'envoi du message. Le message est déposé dans la place PI.

2) *Récepteur B*: la présence d'un jeton dans la place PB_1 signifie que B est en attente d'un message (jeton) sur la place partagée PI. Le franchissement de la transition TR explique la réception elle-même et la présence d'un jeton dans la place PB_2 indique la fin de la réception.

3) A partir du message lui-même, le début de la transaction prend effet quand la transition de l'émetteur TE est franchie et la fin quand la transition du récepteur TR est franchie.

c) Autres éléments de modélisation :

L'envoi *parallèle* ou *concurrent* des messages (m_1, m_2, \dots, m_n) est modélisé par une transition T avec deux ou plusieurs places en sortie (Figure III.10). La transition T décide quelles transitions seront tirées (une, quelques unes, ou toutes) en déposant les jetons nécessaires dans les places correspondantes.

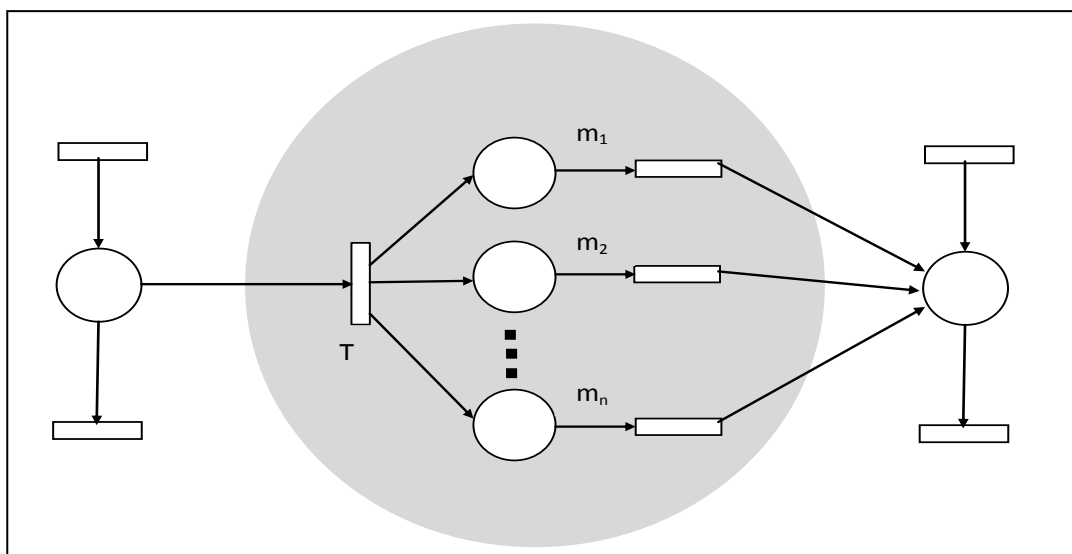


Figure III.10 : Envoi parallèle ou concurrent des messages.

L'envoi *ou exclusif* de messages (m_1, m_2, \dots, m_n) est modélisé par une place à l'entrée de deux ou plusieurs transitions (Figure III.11). La place P doit contenir un seul jeton. En effet une seule transition de la sortie de cette place sera tirée.

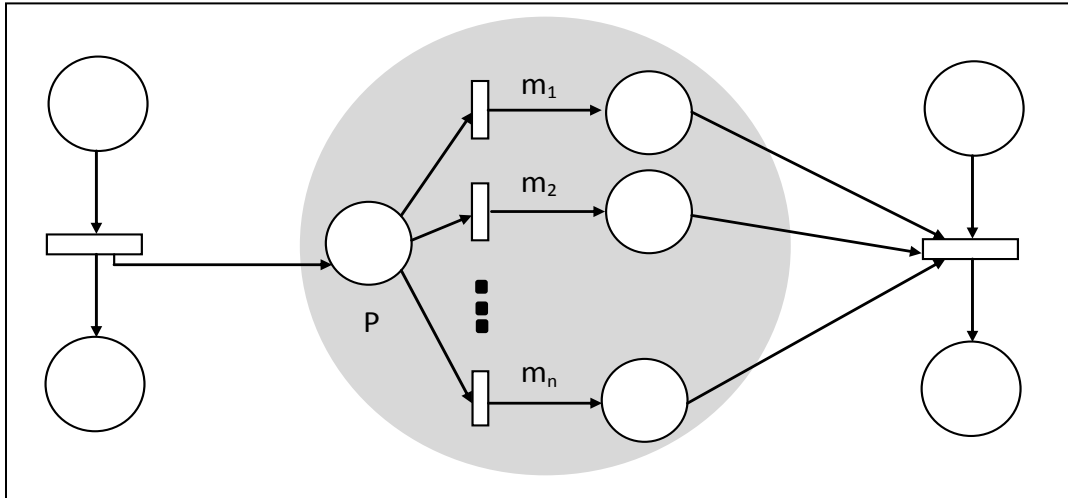


Figure III.11 : Envoi ou exclusif de messages (m_1, m_2, \dots, m_n).

III.2.2.3 Analyse des interactions :

L'utilisation des RdP pour analyser modéliser et valider les interactions entre agents semble être un bon choix de formalisme. En effet, il permet, entre autres de profiter des outils d'analyse dont ils disposent. L'analyse de protocole spécifié permet ainsi au concepteur de vérifier deux types de propriétés : des propriétés générales que l'on retrouve dans tous les protocoles comme des problèmes d'inter-blocage, de réception non spécifiée, et des propriétés fonctionnelles qui sont dépendantes de la fonction du protocole et qui sont décrites dans le cahier des charges lors de la phase d'analyse. En outre, les RdP offrent une représentation graphique des interactions qui reste formelle.

III.2.3 Modélisation des plans d'actions :

III.2.3.1 Principe :

Un plan d'actions est un ensemble coordonné des sous-actions (ou sous-tâches). Il représente la manière d'exécuter une tâche. Graphiquement, la décomposition des tâches est souvent représentée sous forme d'*arbre ET/OU* des sous-tâches qui doivent être réalisées. Les RdP fournissent une expression naturelle et graphique de l'ordonnancement des actions (relation d'ordre causale ou temporelle) des plans de résolution (Figure III.12). La modélisation de plans d'actions par les RdP offre plusieurs avantages :

- ✚ Les propriétés formelles des RdP autorisent une analyse quantitative et qualitative (validation à priori) des plans de résolution.
- ✚ De plus, la modélisation des plans (la décomposition des tâches complexes) par les RdP permet à l'agent de choisir dynamiquement entre plusieurs plans concurrents à partir des propriétés structurelles des réseaux de Petri.
- ✚ Enfin, les modèles RdP des tâches peuvent être utilisés pour l'affectation dynamique des tâches aux agents (section III.2.2).

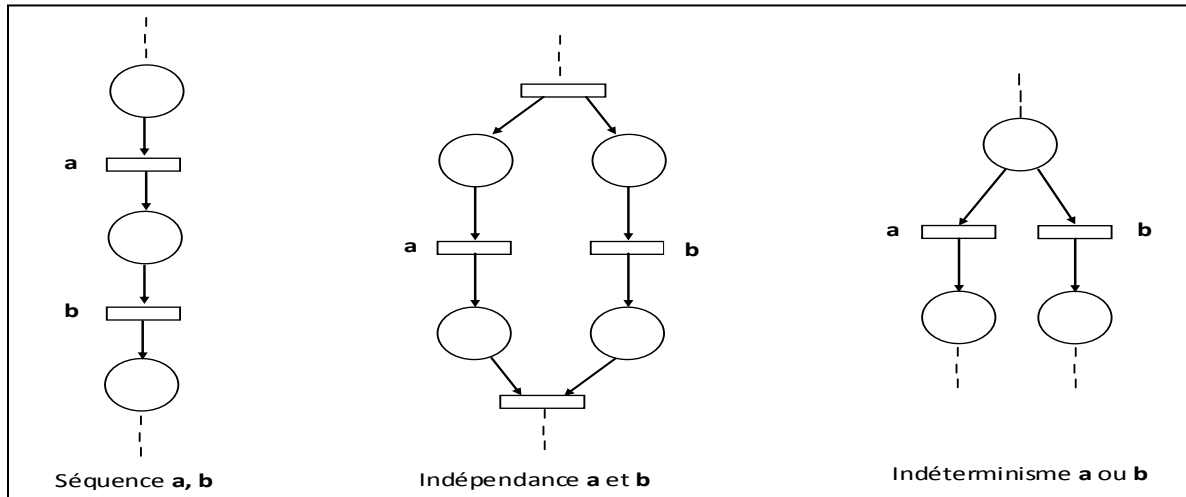


Figure III.12 : Quelques éléments de modélisation.

Pour les plans complexes, les RdP introduisant la notion de la modularité telle que les RdP récursifs, ou hiérarchiques semblent plus adéquats. La Figure III.13 donne un exemple de représentation de plan d'actions au moyen d'un RdP récursif. La partie gauche de la figure spécifie le plan abstrait qui consiste à réaliser les deux actions **a** puis **b**, la partie droite montre un raffinement possible pour l'action **a**.

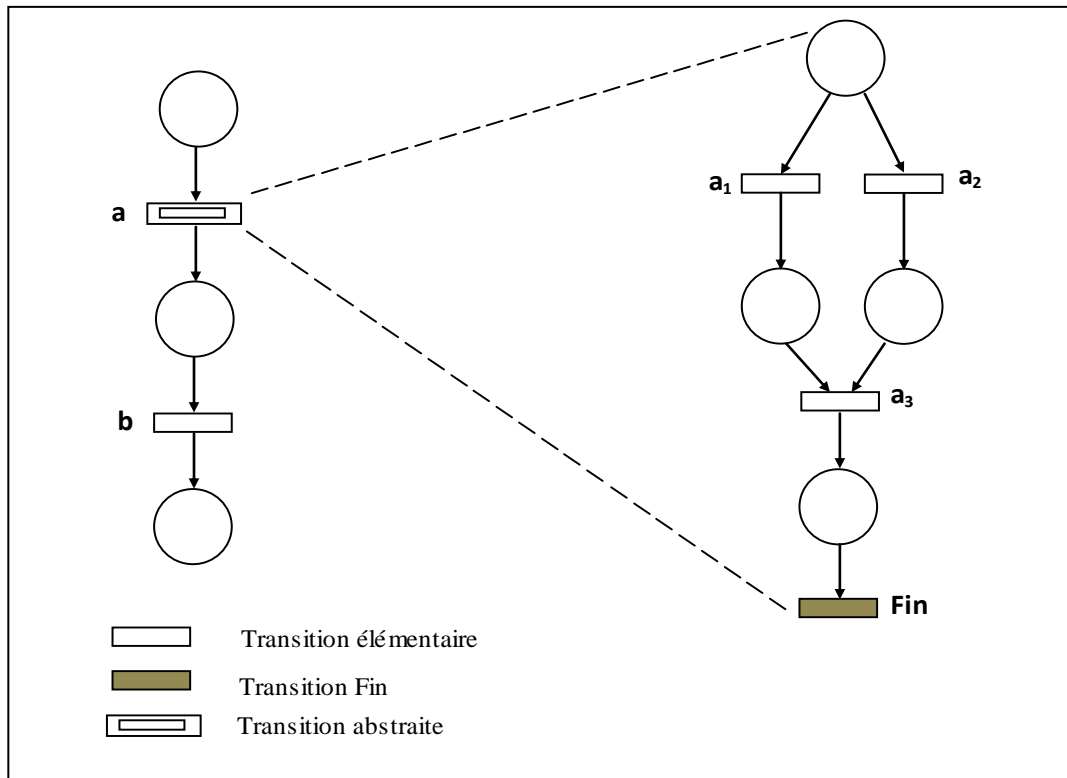


Figure III.13 : Représentation de plans par RdP récursif [12].

III.2.3.2 Modélisation des plans d'actions pour l'allocation des tâches :

La coopération est un concept clé de l'intelligence artificielle distribuée (l'IAD) en général et des SMA en particulier. C'est l'un des concepts qui différencie les SMA des autres disciplines tel que les systèmes orienté objets, les systèmes distribués et les systèmes experts.

La collaboration par partage des tâches est une forme de coopération. La décomposition et l'allocation de tâches constitue l'un des défis des SMA. D'après Stuart Russell [12], la plus part des protocoles de coopération basent sur le principe *Diviser-pour-régner* qui consiste à décomposer les tâches en sous-tâches. Selon R. Davis et R. G. Smith [12] la réalisation d'une tâche passe par trois étapes ;

- 1) Décomposer la tâche en sous-tâches.
- 2) Répartir les sous-tâches entre les différents agents.
- 3) Synthétiser (éventuellement) les résultats partiels pour obtenir un résultat global.

L'avantage principal d'une telle démarche est de réduire la complexité des tâches à réaliser.

En effet, les sous-tâches sont plus simples et requièrent peu de ressources et moins de compétences.

L'un des mécanismes dynamiques d'allocation de tâches est basé sur le principe d'appel d'offre. Un agent ayant besoin de voir une tâche accomplie émet un appel d'offres à destination de tous les agents qu'il croit pouvoir accomplir cette tâche. Les agents qui reçoivent cet appel et qui

peuvent effectivement y répondre envoient une proposition à l'émetteur, qui choisit la proposition la plus adaptée à ses besoins. Ce mécanisme présente l'avantage de la *simplicité* de mise en œuvre, de la *flexibilité* (il permet de nouvelles décompositions des tâches) et de la *robustesse* (la panne d'un agent ne limite pas l'espace de recherche d'un gestionnaire). Cependant il lui manque un processus formel pour évaluer les soumissions des agents contractants. Pour pallier ce problème, il nous semble intéressant de faire appel aux RdP pour leur puissance d'analyse.

a) Hypothèses :

- Les agents du système sont des entités intelligentes *agents cognitifs*.
- On suppose que les buts des agents sont *compatibles*, et que les agents sont *bienveillants*. La bienveillance signifie que les agents cherchent à s'aider les uns les autres ou à parvenir à un compromis si des intérêts portant sur des ressources sont en jeu [17]. (la coopération nécessite la compatibilité des buts des agents et leur bienveillance).
- La communication se fait à l'aide de la technique du *tableau noir*.
- Le système à modéliser peut être ouvert (le nombre d'agent change dans le temps).

b) Architecture générique du système :

Le système est constitué d'un ensemble d'agents, ces agents travaillent en collaboration pour atteindre un objectif commun. En effet les tâches complexes seront décomposées en un nombre fini de sous-tâches. La décomposition des tâches complexes est modélisée par les RdP. Les modèles qui en résultent seront utilisées pour affecter les différentes sous-tâches aux agents du système et pour suivre l'avancement de l'exécution. Nous décrivons dans ce qui suit, les autres composants du système (Figure III.14).

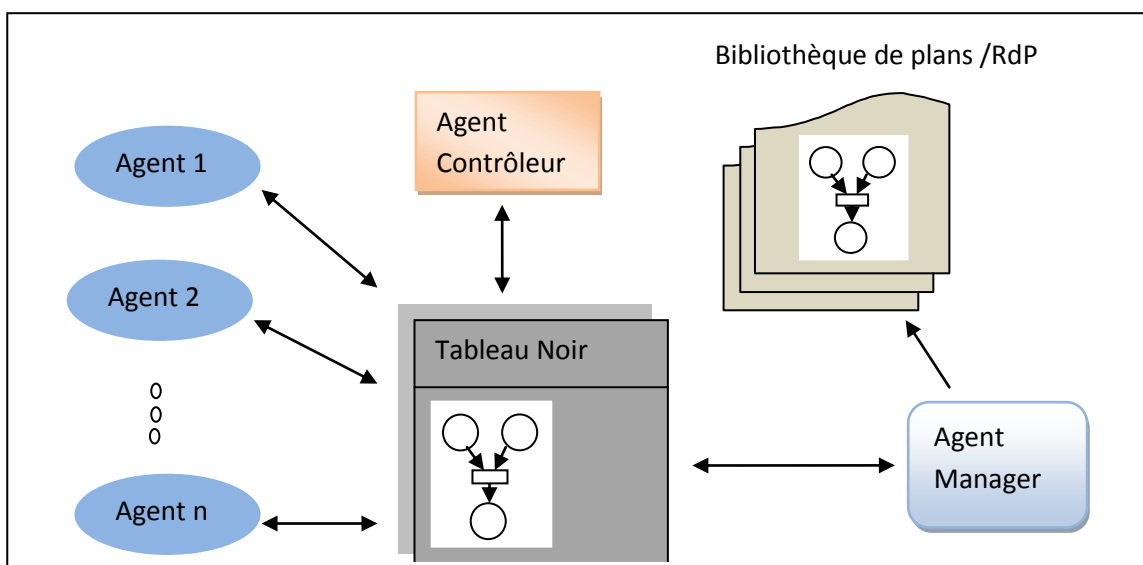


Figure III.14: Architecture globale du système.

❖ Bibliothèque de tâches :

L'ensemble des RdP représentant les plans d'actions seront regroupés dans une bibliothèque. Ces modèles sont génériques (non instanciés), ils seront instanciés au moment de l'exécution. L'agent *manager* accède à cette bibliothèque pour ressortir les plans décrivant la réalisation d'une tâche complexe.

❖ Tableau noir :

L'allocation et le suivi de l'exécution des tâches se fait à l'aide de la technique du *tableau noir* (définie dans la section I.2.4). Les systèmes basés sur un tableau noir sont caractérisés par une communication indirecte et centralisée. La technique du *tableau noir* consiste à utiliser une structure de donnée commune dans laquelle les agents enregistrent, modifient et consultent des informations. La résolution du problème commence par la description du problème dans le tableau. Cette structure comporte en plus les résultats partiels fournis par les agents.

Les principaux avantages de l'utilisation de la technique de tableau noir pour la distribution des sous-tâches par rapport à la technique d'appel d'offre par envoi directe de messages comme le réseaux contractuel Smith [12] telle qu'elle a été proposée dans S.Hsieh [12] sont les suivantes :

✚ *La réduction de taux de communication* : La diffusion des appels d'offres engendre une surcharge du réseau de communication et un ralentissement de l'activité des fournisseurs dû au nombre important de messages qu'ils doivent traiter. Ce problème se complique lorsque les agents fournisseurs envoient leurs propositions sous forme de RdP à l'agent *manager*. Le réseau contractuel devient d'une part, très coûteux en terme de communications lorsque le nombre de fournisseurs et le nombre d'appels d'offres deviennent importants. Il engendre d'autre part des traitements inutiles dans la mesure où les fournisseurs doivent gérer tous les appels d'offres qu'ils reçoivent.

✚ *Le suivi de l'avancée de la résolution du problème* : la technique du tableau noir permet à la fois, la description du problème, l'allocation des sous-problèmes et enfin le suivi de la résolution du problème.

Dans notre système. L'agent *manager* pose le RdP décrivant la tâche à réaliser dans le tableau noir, en vue d'obtention des propositions des différents agents de système. Ces derniers consultent le *tableau noir*, les agents dont leurs compétences et leurs ressources leur permettent de réaliser une ou plusieurs sous-tâches constituant la tâche principale posent leurs propositions sur le *tableau noir*. L'agent *manager* évalue les propositions et fait son choix sur les propositions les plus avantageuses.

Nous avons donc scindé le tableau noir en trois sous-espaces (Figure III.15) :

- *les demandes* : correspondent aux descriptions sous forme des Réseaux de Petri des problèmes à résoudre.
- *Les offres* : correspondent aux offres concernant les demandes ou les problèmes à résoudre.
- *Les états d'avancement* : contiennent des informations sur les états d'avancements des problèmes en cours de réalisation.

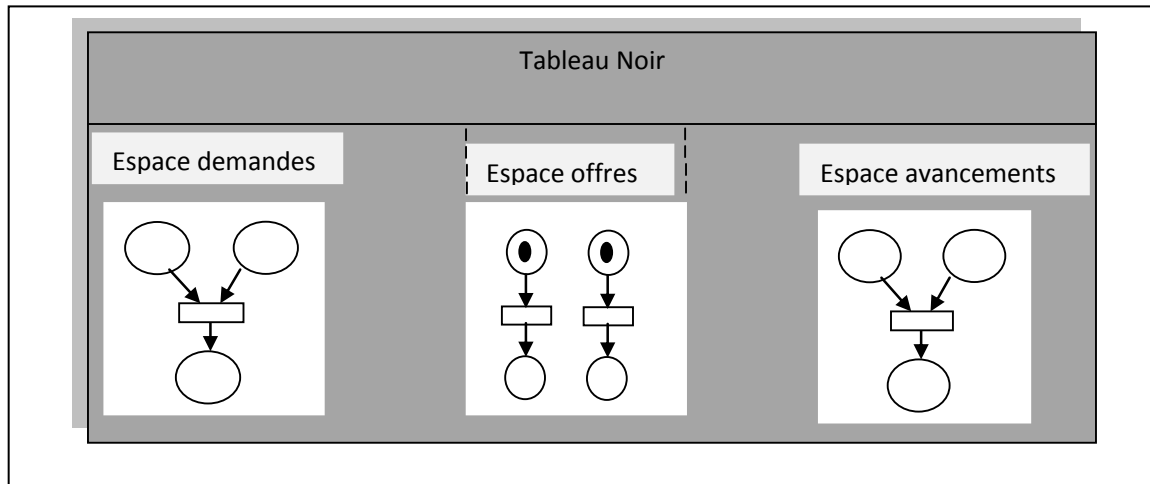


Figure III.15 : Structuration du tableau noir.

❖ Agent contrôleur :

L'utilisation de la technique du tableau noir pour la communication impose l'existence d'un dispositif de contrôle qui gère les conflits d'accès entre les agents. Le rôle principal de l'agent *contrôleur* est de résoudre ce problème de conflits qui peut se produire lorsque les agents viennent lire ou écrire dans le tableau noir. Il peut ainsi détecter les agents intéressés par le dernier changement du tableau noir.

c) Principe et modélisation par réseaux de Petri :

Dans notre travail, l'étude de problème d'allocation de tâches par RdP passe par plusieurs étapes :

- Le *gestionnaire (Manager)* qui est le responsable du contrôle de l'exécution d'une tâche, pose le modèle RdP de la tâche dans le tableau noir.
- Les agents viennent consulter le tableau. Et pose leurs propositions sous forme de RdP.
- Les agents consultant une annonce de tâches à accomplir évaluent l'annonce, ceux qui ont les ressources appropriées, l'expertise ou l'information requise pour accomplir la tâche, élaborent des propositions qu'ils posent dans le tableau noir sous forme des modèles RdP.
- L'agent *manager*, évalue individuellement chacune des propositions. Seules les propositions jugées intéressantes seront évaluées collectivement.
- L'agent *manager* fusionne les modèles RdP de propositions filtrées et le modèle RdP de tâche. Il analyse le modèle résultant pour choisir les meilleures propositions.

❖ Modélisation des tâches :

Une tâche peut être vue comme un ensemble coordonné (lié) des sous-tâches, nous la représentons par un RdP qui fournit une expression naturelle et graphique des synchronisations des sous-tâches parallèles ou de leur enchaînement. Pour modéliser une tâche par un RdP nous utilisons les transitions pour représenter les sous-tâches (les opérations).

Dans un modèle de tâche :

- Il y a une seule transition initiale, et une ou plusieurs transitions finals (pour les tâches admettant plusieurs chemins d'exécution).
- Toute transition du modèle RdP de la tâche appartient à un chemin qui a pour origine la transition de début $T_{initiale}$ de la tâche, et pour extrémité une transition de la fin de la tâche T_{finale} .

❖ Modélisation des propositions :

Nous proposons également une modélisation par RdP des propositions des potentiels contractants. Ces derniers doivent reformuler leurs propositions sous forme de modèles RdP. Nous supposons :

- Qu'un agent peut proposer d'effectuer plusieurs sous-tâches.
- Qu'une proposition ne concerne qu'une seule sous-tâche.

Toute proposition est représentée par un RdP constitué d'une place d'entrée P_{ia} . D'une transition T_i , et une place de sortie P_{id} (Figure III.16). Les informations sur l'agent, et les critères (temps, coûts,...etc.) proposés pour la réalisation de la sous-tâche sont représentés par un jeton coloré dans la place d'entrée P_{ia} .

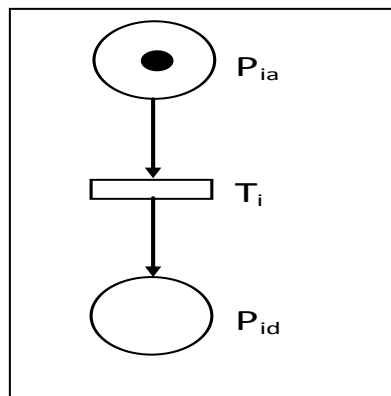


Figure III.16 : Modèle RdP d'une proposition.

❖ Fusion des modèles réseaux de Petri :

Le but de cette étape est d'obtenir un seul RdP décrivant à la fois les différentes tâches et les relations entre agents de protocole.

En premier lieu, on s'intéresse à la fusion des modèles RdP constituant l'ensemble des propositions des potentiels contractants et le RdP décrivant la tâche afin d'obtenir un seul RdP décrivant à la fois la tâche et les relations entre agents et les sous-tâches (Figure III.17). La fusion consiste à regrouper en premier lieu les modèles RdP des propositions de façons à obtenir un seul modèle RdP pour chaque sous-tâche (Figure III.17 (a)). Ensuite les modèles RdP des propositions obtenus et le modèle RdP de la tâche seront fusionnés pour avoir qu'un seul modèle (Figure III.17 (b)).

Remarque :

Dans notre exemple, on utilise le RdP coloré pour différencier les différents marquages des différentes propositions.

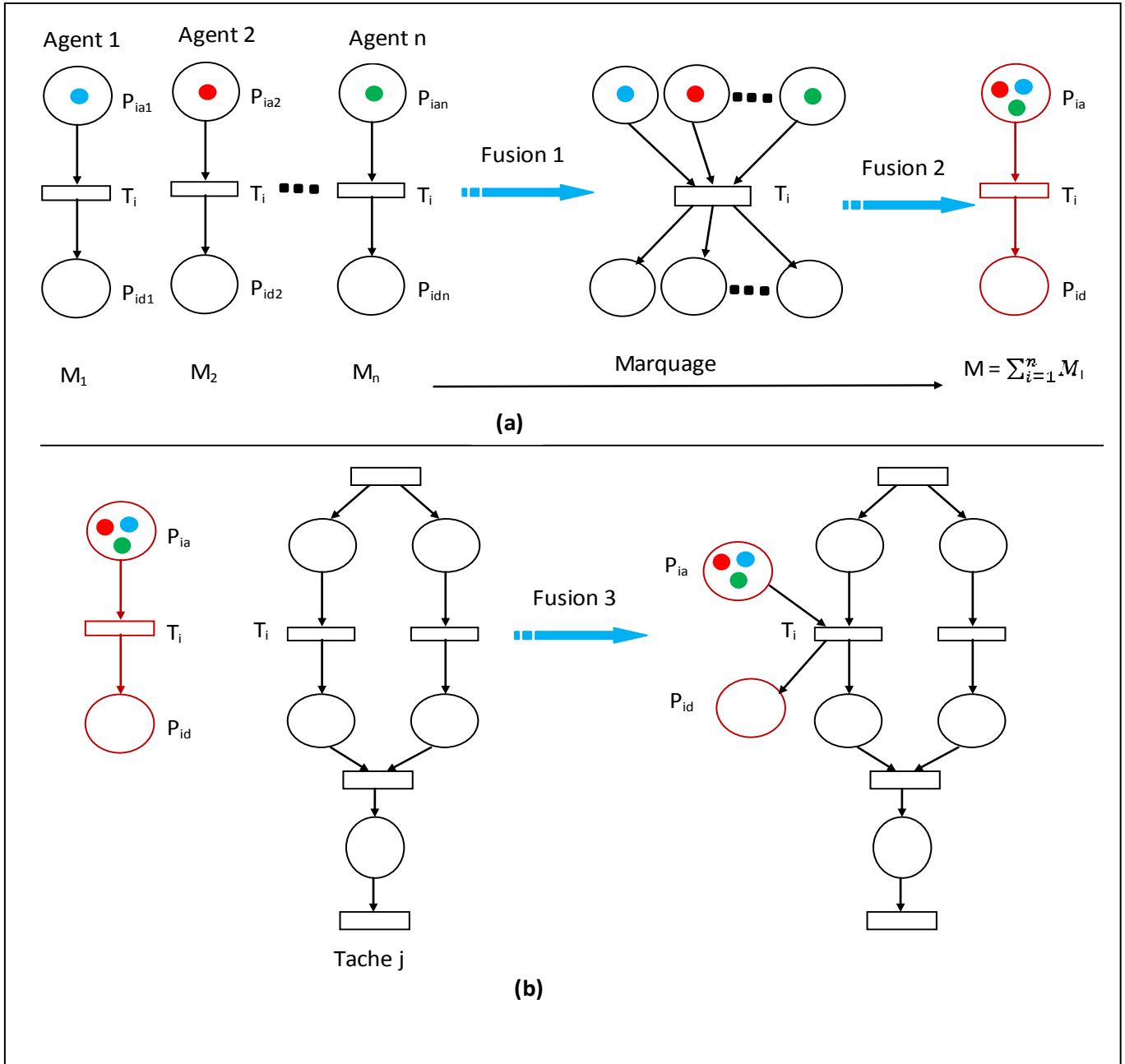


Figure III.17 : Fusion de modèle tâche et modèles propositions.

En suite, on fusionne l'ensemble de tâches pour avoir un seul modèle de réseau de Petri qui décrit notre système. Pour ce faire, il nous faut des Arcs de fusion, ces arcs servent à fusionner deux Places ou deux transitions de deux réseaux différents. Les Places doivent avoir les mêmes caractéristiques, à savoir accepter les mêmes types de Jetons.

Ces Arcs de fusion ont un type, c'est 'rr' dans le cas où cet Arc fusionne deux Nœuds qui se trouvent dans deux sous-réseaux de même niveau (c'est-à-dire au même niveau hiérarchique), c'est 'pr' quand la fusion a lieu entre un Nœud d'un réseau et un Nœud de l'un de ses sous-réseaux direct, 'rp' quand la fusion a lieu entre un Nœud d'un sous-réseau et un Nœud de son réseau englobant (Figure III.18) [27].

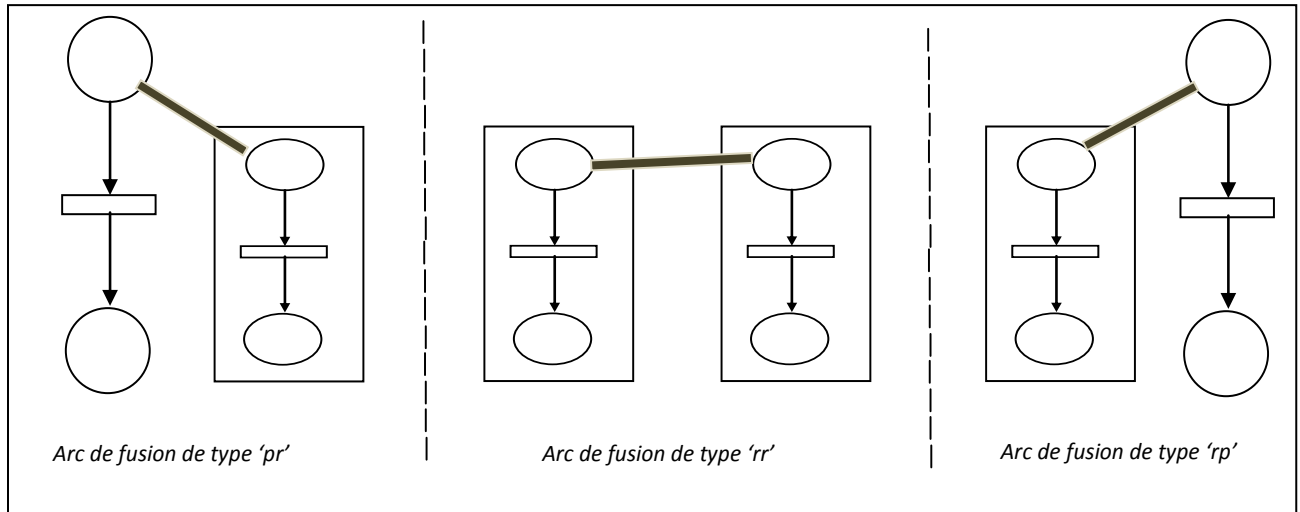


Figure III.18 : les différents types d'arcs de fusion.

d) Un réseau de Petri permet de visualiser :

Une des caractéristiques importantes des RdP est de pouvoir présenter graphiquement certaines relations, de visualiser certaines notions. Nous allons en donner des exemples.

❖ **Parallélisme :**

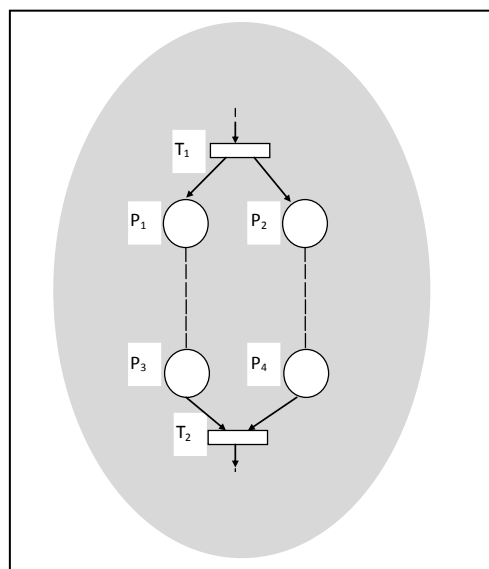


Figure III.19 : Parallélisme.

Ceci est représenté sur la figure (III.19). On voit clairement qu'après le franchissement de T_1 , et jusqu'au franchissement de la transition T_2 , on a des évolutions en parallèle, de la place P_1 à la place P_3 d'une part et de la place P_2 à la place P_4 d'autre part. Chacune de ces deux évolutions peut se faire à son propre rythme.

❖ **Synchronisation :**

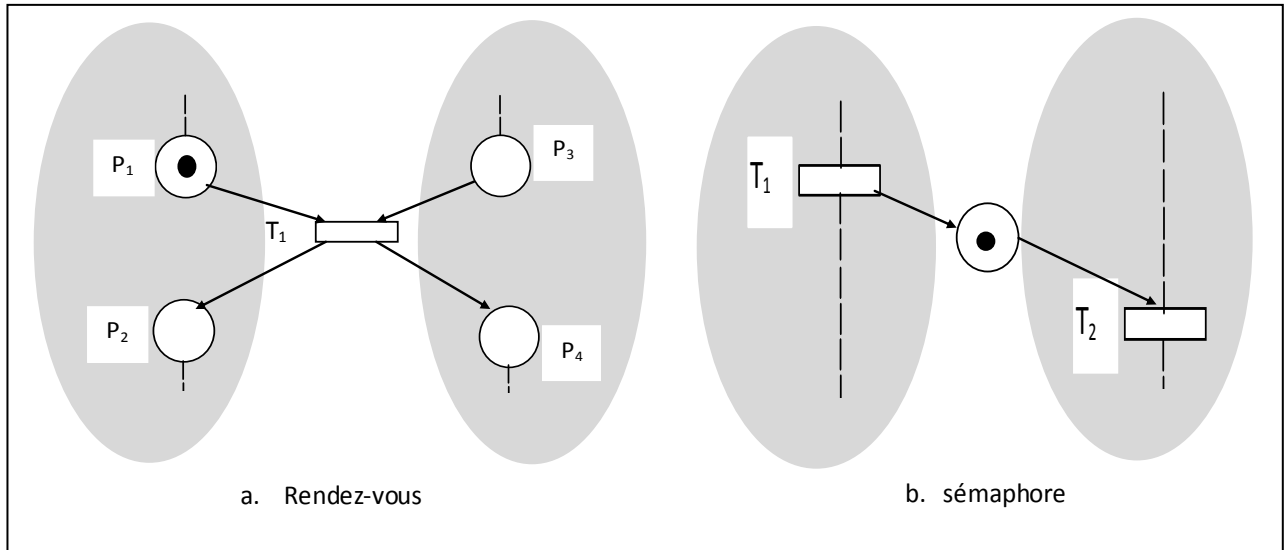


Figure III.20 : Synchronisation [19].

La figure (III.20.a) représente la synchronisation réciproque de deux évolutions. Dans la partie gauche, la place P_1 est marquée, mais avant de poursuivre l'évolution on attend que la place P_3 soit marquée aussi. Dès que ces deux places sont marquées, la transition T_1 peut être franchie, ce qui entraîne le marquage des places P_2 et P_4 et la poursuite de l'évolution dans chacune des parties du RdP. De façon réciproque, si la place P_3 avait été marquée avant la place P_1 , c'est la partie droite qui aurait attendu pour se synchroniser avec la partie gauche (*type rendez-vous*).

Le cas de la figure (III.20.b) est différent. L'évolution de la partie gauche est indépendante de celle de la partie droite. En revanche, le franchissement de la transition T_2 doit se faire après le franchissement de la transition T_1 . La place P_1 mémorise l'autorisation de franchir T_2 . Notons que si la place P_2 est marquée avant la place P_1 , dès que la transition T_1 est franchie, la transition T_2 peut être franchie aussi (*type sémaphore*).

❖ **Partage de ressource :**

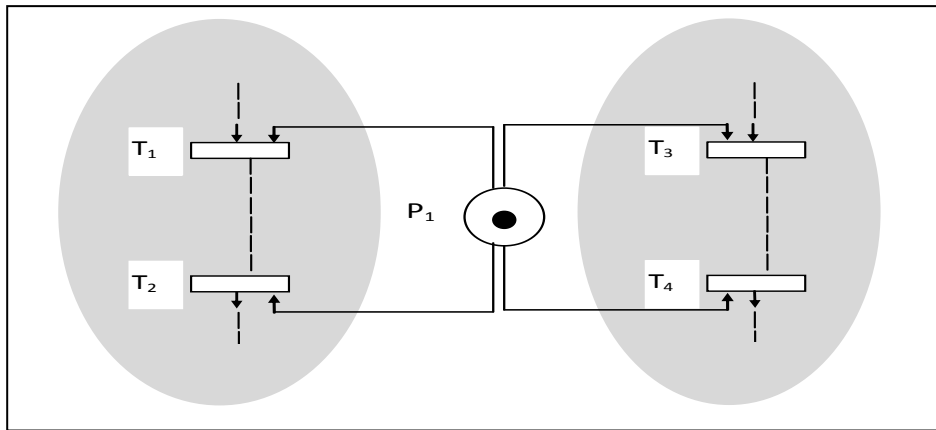


Figure III.21 : Partage de ressource.

Sur la figure (III.21), la place P_1 modélise la disponibilité d’une ressource qui peut être utilisée par la partie gauche à partir du franchissement de T_1 et jusqu’au franchissement de T_2 ou par la partie droite dans des conditions similaires, mais pas par les deux simultanément.

En informatique, l’exécution mutuelle de parties de programmes qui utilisent une même ressource, comme une mémoire commune par exemple, peut se modéliser ainsi.

Dans le cas plus général, il peut y avoir plusieurs ressources identiques (plusieurs marques dans P_1) et la partie gauche, ainsi que la partie droite, peuvent utiliser plusieurs ressources simultanément.

❖ **Mémorisation :**

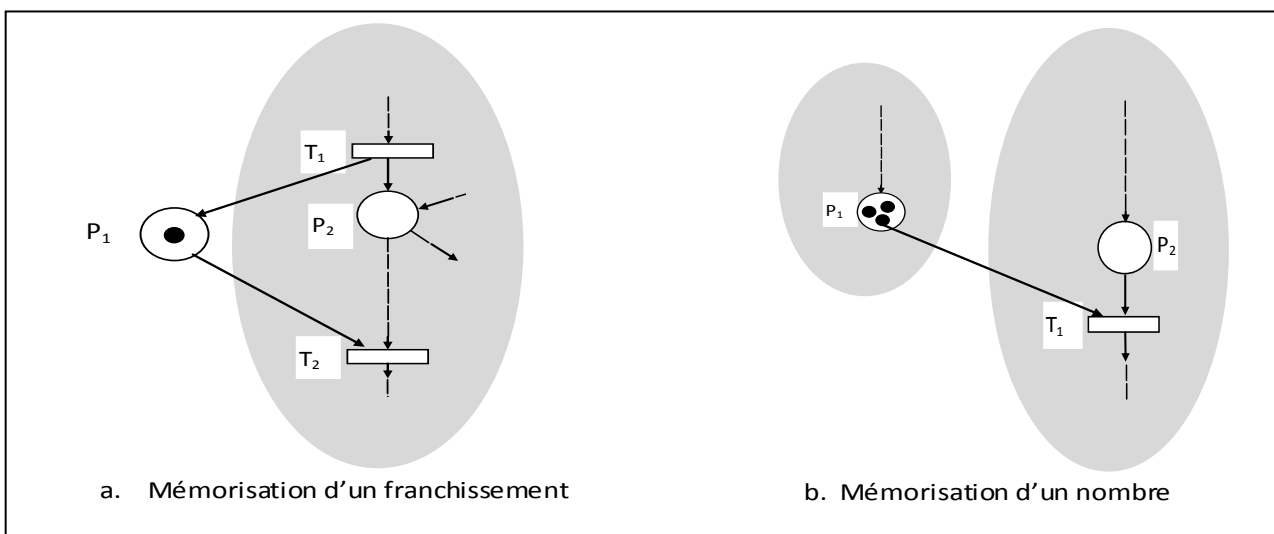


Figure III.22 : Mémorisation [19].

La place P_1 de la figure (III.22.a) mémorise le fait que la transition T_1 a été franchie, ce qui autorise le franchissement ultérieur de T_2 , sous service que les autres conditions de validation soient remplies (ce cas est assez similaire à celui de la figure (III.20.b)).

On peut également mémoriser un nombre (nombre de pièces dans un stock par exemple) par la présence de plusieurs marques dans une place. Il y a trois marques dans la place P_1 de la figure (III.22.b).

III.2.4 Suivi d'exécution :

Finalement, le RdP de la tâche sera utilisé lors de l'exécution des sous-tâches pour contrôler l'état d'avancement de la tâche. Le RdP de la tâche est un RdP réduit. Un agent finalisant la réalisation d'une sous-tâche doit poser un jeton dans la place ressource à l'entrée de la transition qui représente la sous-tâche réalisée. Il tire le RdP pour mettre à jour l'état du système. Le franchissement de la transition finale indique la réalisation réussie de la tâche entière.

CONCLUSION

Dans ce chapitre nous avons présenté quelques aspects du domaine système multi-agent (SMA) pour lesquels une modélisation par réseaux de Petri (RdP) sera appropriée. Nous avons proposé une modélisation par RdP du concept agent, des interactions entre agents et enfin de plans d'actions des agents.

Etant une approche de modélisation orientée analyse, les RdP, permettent d'entamer une phase d'analyse permettant au concepteur de valider à priori le modèle par vérification des propriétés. Comme nous l'avons vu dans ce chapitre, les RdP peuvent être utilisés par les agents eux même pour valider leurs interactions, où pour affecter les tâches aux différents agents du système.

Le but du prochain chapitre est d'appliquer notre travail à une étude de cas pour montrer l'apport de l'utilisation des RdP pour la modélisation des SMA et l'analyse de quelques une de leurs propriétés.

Chapitre IV:

Etude de cas

INTRODUCTION

La croissance de la complexité des systèmes industriels modern, en particulier les systèmes flexibles, rend de plus en plus la tache difficile aux concepteurs pour les développer, d'autre part, ils sont de plus en plus réponsus et on un intérêt économique car ils produisent des produits variés à des couts compétitifs.

Le système de production flexible est un ensemble de machine-outil à commande numérique desservies par un, ou plusieurs, robot(s) qui assure(nt) le transfert des pièces. D'une façon générale, de tels système sont appréhendés comme des systèmes à événements discrets (SED) dont l'évolution est asynchrone et se trouve conditionné par l'occurrence de phénomènes (événements), leur conduite nécessite ainsi la coordination d'événements. La résolution distribuée par l'approche Multi-Agents s'adapte bien avec la nature des systèmes de production flexibles.

Il est nécessaire ainsi de mener une étude efficace de ces systèmes avant toute implémentation, les concepteurs doivent disposer d'outils d'investigation et d'évaluation qui permettent d'orienter et de valider leur choix. Les réseaux de Petri (RdP) qui disposent d'une symbolique graphique et reposant sur des concepts mathématiques peuvent aider les concepteurs dans leur travail.

Ce chapitre a pour objectif de montrer l'apport des RdP et des SMA, dans l'implémentation, l'analyse et la validation d'un système de production. Cette étude est illustrée à travers un exemple réel qui est la chaine de fabrication des armoires frigorifiques de l'ENIEM.

IV.1 Modélisation et validation de protocole Winrdp :

La modélisation des systèmes de production conduit généralement à des modèles très complexes ; le réseau de petri correspondant est de très grande taille, ce qui rend les techniques d'analyse très difficiles. Pour cela, il existe plusieurs logiciels (Contract Net, Winrdp,...) permettant de faciliter l'analyse et le suivi de la dynamique du système étudié (modélisé) quelque soit le degré de ça complexité. Notre choix de logiciel s'est porté sur le logiciel Winrdp, ce dernier a été réalisé par S.Cherief et S.Rachedi sous la direction de monsieur C.Amer-Yahia de l'université Mououd Mammeri de Tizi-Ouzou. Un tel logiciel permet d'assister les utilisateurs des RdP dans l'analyse, l'évaluation des performances et la simulation des systèmes de production.

Le logiciel Winrdp permet donc [15] :

- ✓ La saisie graphique ou textuelle d'un RdP modélisant un système de production.
- ✓ La validation du système modélisé par la recherche des bonnes propriétés du RdP.
- ✓ L'évaluation des performances d'un système modélisé par un graphe d'événement temporisé.
- ✓ La simulation graphique ou textuelle du réseau par la visualisation de l'évolution du marquage en fonction de temps.

IV.1.1 Présentation du logiciel [15] :

L'analyse du logiciel passe tout d'abord par le choix d'un modèle RdP. Il existe quatre modèles différents : les RdP ordinaires, les RdP T-temporisés, le RdP P-temporisés et enfin le modèle le plus général, les RdP P-T-temporisés.

Le logiciel peut être utilisé de deux façons différentes soit, en utilisant la partie saisie graphique soit, la partie saisie textuelle. Ces interfaces permettent toutes les deux l'obtention de résultats identiques, mais sous deux formes différentes.

a) Présentation de l'interface de saisie graphique :

L'interface de saisie graphique est très interactive puisqu'elle offre à l'utilisateur la possibilité de saisir son réseau tout en contrôlant les paramètres tels que la taille des places et des transitions, les marquages des places, les temps des places et des transitions, le déplacement du réseau à l'intérieur de la fenêtre et la copier dans d'autres. La fenêtre principale peut contenir plusieurs fenêtres permettant ainsi le travail sur plusieurs fenêtres.

Après la saisie du réseau, l'utilisateur a la possibilité de sauvegarder le RdP et ses paramètres au moyen d'un fichier de sauvegarde sous l'extension adéquate (*.RdP, *.RdPTT, *.RdPPT, *.RdPPTT) et cela suivant le modèle choisie au départ.

Les propriétés du réseau peuvent être vérifiées en utilisant le menu validation de l'interface.

A partir du menu performance, on calcule les circuits élémentaires du réseau, le temps de cycle et la performance maximale du système modélisé.

b) Présentation de l'interface de saisie textuelle :

De même que l'interface de saisie graphique, l'interface de saisie textuelle est très interactive puisqu'elle offre à l'utilisateur la possibilité de saisir les RdP de grande taille via les matrices d'incidence avant et arrière et de saisir le marquage de places, le temps associé aux places et/ou aux transitions d'un réseau et cela suivant le modèle choisi (RdP ordinaire, RdP T-temporisé, RdP P-temporisé, RdP P-T-temporisé).

Après la saisie des différentes matrices, l'utilisateur a la possibilité de sauvegarder ces matrices au moyen de fichiers de sauvegarde sous l'extension adéquate (*.MAT, *.MATT, *.MATP, *.MATPT) et cela suivant le modèle choisie au départ.

Les propriétés du réseau peuvent être vérifiées en utilisant le menu validation de l'interface, ainsi que le calcul de sa performance maximale et les circuits élémentaires du réseau.

IV.2 Description du problème :

L'objectif principal de la gestion de production est de déterminer l'allocation des ressources afin de réaliser l'ensemble des commandes arrivant dans le système. Une telle gestion doit respecter différents types de contraintes notamment :

- *Des contraintes sur les ressources* : le traitement d'une commande ne peut commencer que si les ressources nécessaires à son usinage sont disponibles ;
- *Des contraintes temporelles* : le traitement d'une commande ne peut commencer qu'à une certaine date correspondant à l'arrivée de la commande dans l'usine et devra finir avant une certaine date limite.

IV.2.1 Application de la modélisation :

Cette application reprend intégralement une grande partie du rapport interne rédigé par R.Kara et C.Amer-Yahia du département automatique. L'étude porte sur un cas industriel, chaîne de fabrication des armoires frigorifiques de l'Entreprise Nationale de l'Industrie de l'ElectroMénager (ENIEM), entreprise implantée à Tizi-Ouzou.

IV.2.1.1 Description de la chaîne de fabrication des armoires frigorifiques:

L'atelier considéré est un système flexible, il produit différents types de réfrigérateurs.

La figure (IV.1) montre une vue de dessus de la chaîne de fabrication des cuves de l'entreprise ENIEM.

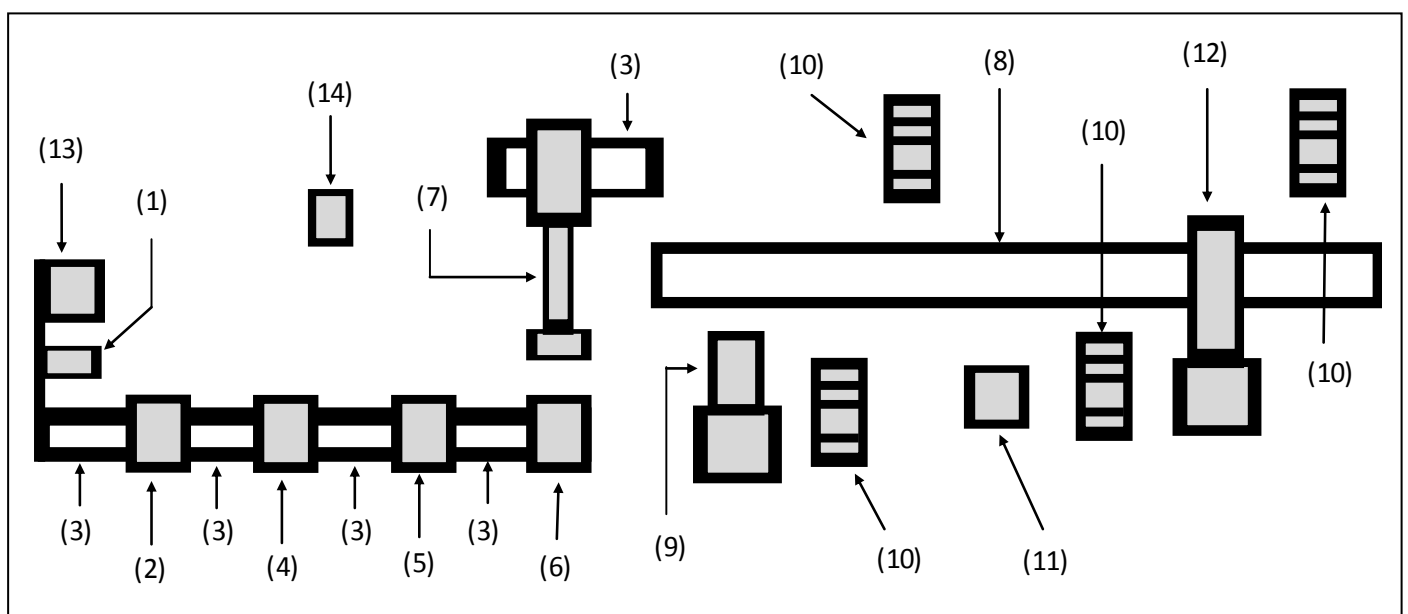


Figure IV.1 : Vue de dessus de la chaîne de fabrication des armoires frigorifiques de l'ENIEM.

Elle se compose des éléments suivants :

1- *Le chargeur* : Il a pour fonction de hisser une après l'autre les plaques métalliques mises en place préalablement et de les envoyer au transporteur pour le poste suivant.

2- *Presse d'encochage* : Elle a pour fonction d'arrêter en place les plaques métalliques envoyées par le transporteur et d'y découper des trous et encoches avant de les envoyer au poste suivant de formage et de montage de pièces.

3- *Convoyeur* : Sa fonction est de mettre en position et envoyer les plaques d'une machine vers une autre.

4- *Laminoir à profiler à froid* : Il comporte des rouleaux qui assurent de manière continue le bordage des faces avant et arrière des plaques destinées aux armoires frigorifiques.

5- *Presse hydraulique spéciale* : Cette presse a pour fonction de mettre en position une plaque dont les deux bords arrière et avant sont formés, de pilier en L deux cotés courts droit et gauche et de terrasser le bord avant.

6- *La Cintreuse* : Plus exactement, outillage de cambrage qui a pour fonction de mettre en position dans les deux sens une plaque mise en forme aux postes précédents et de cintrer ses deux cotés afin de la former ainsi en boîte.

7- *Dispositif de renversement* : Il a pour fonction de recevoir au moyen de ventouses les armoires mises en forme au poste de cambrage et de les transporter au poste suivant.

8- *Transporteur* : Ce convoyeur à écailles est destiné au transport des armoires frigorifiques entre postes de soudages et au transport des armoires soudées.

9- *Machine à souder par points multiples* : Cette machine a pour fonction de souder la plaque arrière à celle latérales (les travaux de mise en place de la plaque arrière et de reprise des armoires sont effectués manuellement par les travailleurs). Lorsque l'opérateur met en place les plaques arrière et latérale et appuie sur le bouton de démarrage, cette machine effectue l'introduction, le soudage et le déchargement des armoires d'une manière continue et automatique. Les armoires soudées sont reprises par le travailleur qui les dépose sur le transporteur.

10- *Machine à souder par points* : Cette machine a pour fonction de souder les plaques de fonds, cloison et renforts sur l'armoire frigorifique. Les différentes opérations effectuées par cette machine sont les suivantes :

- a. Souder la plaque avant à l'armoire frigorifique.
- b. Souder les deux coins du bord ourlé se trouvant à la partie supérieure de l'armoire.
- c. Souder la cloison à l'armoire.
- d. Souder les renforts à la plaque avant.
- e. Souder la plaque renfort de la charnière au plafond de l'armoire. Tous les soudages sont effectués par des opérateurs humains.

11- *Machine à souder par points et par bossage* : A pour fonction de souder aux armoires les renforts droit et gauche qui rendent les armoires plus rigides.

12- *Machine à souder par points multiples* : Cette machine est utilisées pour souder la plaque de fond mise en place par le travailleur attendant au poste précédent ; l'armoire avance sur le transporteur vers cette machine, qui la prend, la soude et l'éjecte automatiquement à la chaîne.

13- *Matière brute* : Elle est représentée par des plaques métalliques. Les dimensions changent selon le type de réfrigérateurs.

14- *Séquenceur* : Il représente l'unité de commande de la chaîne.

Le produit subit les opérations de différentes machines selon un plan de fabrication ; la séquence suivante est le chemin possible de la production : (OP₁, OP₃₁, OP₂, OP₃₂, OP₄, OP₃₃, OP₅, OP₃₄, OP₆, OP₇, OP₃₅, OP₈)

Opération	Machine
OP ₁	M ₁
OP ₂	M ₂
OP ₃ (OP ₃₁ , OP ₃₂ , OP ₃₃ , OP ₅ , OP ₃₄ , OP ₃₅)	M ₃ (M ₃₁ , M ₃₂ , M ₃₃ , M ₅ , M ₃₄ , M ₃₅)
OP ₄	M ₄
OP ₅	M ₅
OP ₆	M ₆
OP ₇	M ₇
OP ₈	M ₉

Table IV.1 : Relation opération/machine.

IV.2.1.2 Modélisation par un SMA :

En analysant le problème, nous pouvons tracer les caractéristiques suivantes de l'environnement :

Non accessible pour le système : chaque entité ne possède qu'une vue partielle du système, elles n'ont qu'une connaissance locale de leur environnement ; chaque élément ne connaissant que les éléments de niveaux inférieur et supérieur.

Dynamique : les différents événements pouvant affecter le système comme l'arrivée des commandes clients, les perturbations affectant les machines (ou élément du système) (ajout / retrait des machines, dysfonctionnement des machines), arrivent à des moments imprévisibles, augmentant ainsi la dynamique de l'environnement.

Non déterministe : l'effet d'une action d'une des entités du système dépend fortement de l'état dans lequel se trouve le système. Par exemple, un élément qui reçoit la demande de réalisation de certaines tâches peut le refuser, l'accepter selon son état actuel.

Cette analyse permet ainsi de constater le caractère distribué du système. Chaque entité composée (machines) coopère pour un objectif commun : la fabrication de produits demandés par le client. Il est donc naturel de modéliser chacune des composantes du système par un agent.

a) Premier identification des agents et des rôles :

L'identification des agents et des rôles constitue une étape clé dans la modélisation des SMA. Onze types d'agent existent dans notre système:

1- *Agent Planificateur* : Le rôle de l'agent planificateur (séquenceur) est le traitement des commandes client.

2- *Agent Contrôleur* : Son rôle est la gestion de l'accès au tableau noir. Il détecte ainsi les agents intéressés par le dernier changement du tableau noir.

3- *Agent Chargeur* : L'agent chargeur est destiné à piloter une machine de chargement de tôles, pour ce faire, cet agent reflète l'état de la machine (disponible/occupée). Cet agent reçoit l'ordre de l'agent planificateur via un tableau noir.

4- *Agent d'Encochage* : L'agent d'encochage est destiné à piloter une machine, cette dernière effectue une seule opération machine, le pressage. Cet agent reflète l'état de la machine (disponible/occupée), reçoit l'ordre de l'agent convoyeur via un tableau noir.

5- *Agent Convoyeur* : L'agent convoyeur est destiné à piloter une machine, pour ce faire, cet agent reflète cinq états correspondant aux cinq niveaux de la machine, sur chaque niveau on dispose d'un capteur. Cet agent reflète l'état de la machine (disponible/occupée) sur les cinq niveaux. Cet agent reçoit l'ordre de l'agent inférieur via un tableau noir.

6- *Agent Laminoir à profiler à froid* : L'agent laminoir à profiler à froid est destiné à piloter une machine qui sert à effectuer le bordage des faces avant et arrière des plaques métalliques. Cet agent reflète l'état de la machine (disponible/occupée). Cet agent reçoit l'ordre de l'agent convoyeur via un tableau noir.

7- *Agent Presse Hydraulique Spéciale* : L'agent Presse Hydraulique Spéciale est destiné à piloter une machine de pressage. Cet agent reflète l'état de la machine (disponible/occupée), cet agent reçoit l'ordre de l'agent convoyeur via un tableau noir.

8- *Agent Cintreuse* : L'agent cintreuse est destiné à piloter une machine de cambrage. Cet agent reflète l'état de la machine (disponible/occupée), cet agent reçoit l'ordre de l'agent convoyeur via un tableau noir.

9- *Agent de renversement* : L'agent de renversement est destiné à piloter une machine qui sert à déplacer la pièce du poste de cambrage au poste suivant. Cet agent reflète l'état de la machine (disponible/occupée), cet agent reçoit l'ordre de l'agent cintreuse via un tableau noir.

10- *Agent Transporteur* : L'agent transporteur est destiné à piloter une machine, pour ce faire, cet agent reflète l'état de la machine (disponible/occupée). Cet agent reçoit l'ordre de l'agent inférieur via un tableau noir.

11- *Agent soudeur* : L'agent soudeur est destiné à piloter une machine. Cet agent effectue quatre opérations de soudage sur sept niveaux différents, la première opération est le soudage par points multiples, la deuxième est le soudage par points, la troisième est le soudage par points et par bossage, la dernière est le soudage par points multiples. L'agent reflète l'état de la machine (disponible/occupée) sur les deux niveaux. Cet agent reçoit l'ordre de l'agent inférieur via un tableau noir.

Remarques :

- Tout les agents sont des agents réactifs, ils ont donc un comportement de type stimulus-réponse face a ce qu'ils perçoivent.
- Le comportement de l'agent machine subit le deux règles suivantes :
 - s'il y a une requête (ordre) qui vient de l'agent supérieur, l'agent machine répond par son état (disponible / occupée) et par le temps opératoire correspondant.
 - Si l'agent machine est choisi pour réalisé une opération, il change l'état de la machine correspondante (occupée). Quand le traitement est terminé, l'agent machine correspondante (disponible).

Modélisation de la chaine de fabrication des armoires frigorifiques par un RdP :

Une manière très répandue de faire la décomposition, c'est de décomposer le système de production en plusieurs parties fonctionnelles ; comme déjà fait dans la section précédente pour la modélisation par SMA, de ce fait, le modèle réseaux de Petri de la chaine est obtenu par la décomposition du système en sous-systèmes, lesquels sont modélisés par des sous-réseaux de Petri. L'union de tous les sous-réseaux nous donne le réseau de Petri global de la chaine.

Un sous-système (ou un élément de la chaine, ou un agent) représente une ou plusieurs opérations machine. La chaine décompose en huit sous-systèmes, qui sont :

i) Le chargeur de tôles. ii) La presse d'encochage. iii) Le convoyeur. iv) Le laminoir. v) La presse hydraulique. vi) La cintruse. vii) Le dispositif de renversement. viii) Les différents postes de soudage et le transporteur.

✓ **Modélisation du chargeur de tôles :**

Le modèle réseau de Petri du chargeur (ou de l'agent chargeur) es facile à construire (figure IV.2), les places et les transitions sont explicitées dans la table IV.2.

Le marquage initial du réseau est : $[R_1 \ M_1 \ M_{31} \ OP_1 \ P_1]^T \equiv [1 \ 1 \ 1 \ 0 \ 0]^T$.

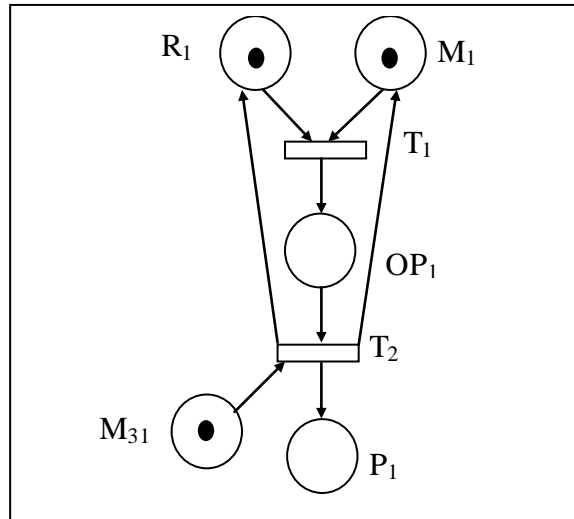


Figure IV.2 : Modèle RdP du chargeur.

M ₁	Disponibilité du chargeur
R ₁	Disponibilité de la matière brute
OP ₁	Opération de chargement
M ₃₁	Convoyeur vide et disponible au 1 ^{er} niveau
P ₁	L'état du convoyeur au 1 ^{er} niveau
T ₁	Début de l'opération de chargement
T ₂	Déchargement

Table IV.2 : Interprétation des places et transitions.

✓ **Modélisation de la presse d'encoche :**

De la même manière que précédemment on construit le modèle RdP de la presse d'encoche (ou de l'agent d'encoche) (figure IV.3). Les places et les transitions sont explicitées dans la table IV.3.

Le marquage initial du réseau est : $[P_1 \ M_2 \ M_{32} \ OP_2 \ M_{31} \ P_2]^T \equiv [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$.

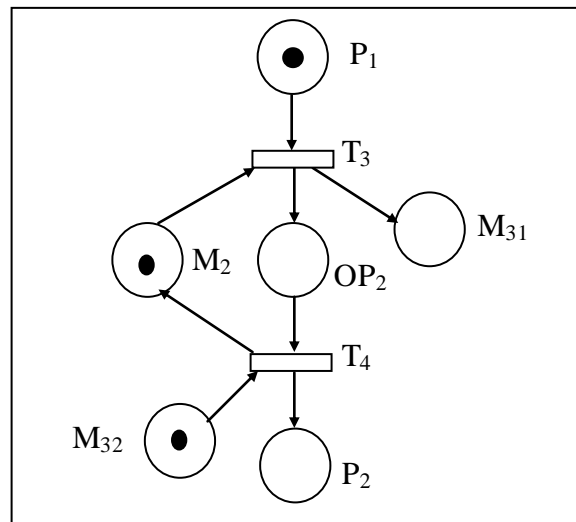


Figure IV.3 : Modèle RdP de la presse d'encoche.

M ₂	Disponibilité de la presse
OP ₂	Opération de pressage
M ₃₁	Convoyeur vide et disponible au 1 ^{er} niveau
M ₃₂	Convoyeur vide et disponible au 2 ^{em} niveau
P ₁	Convoyeur plein au 1 ^{er} niveau
P ₂	L'état du convoyeur au 2 ^{em} niveau
T ₃	Début de l'opération de pressage
T ₄	Fin de l'opération de pressage

Table IV.3 : Interprétation des places et transitions.

✓ **Modélisation du convoyeur :**

L'agent convoyeur joue cinq rôles (cinq états), il détecte cinq niveaux grâce à cinq capteur de niveaux chaque capteur est placé dans un niveaux.

Le modèle RdP de convoyeur composé de cinq sous-réseaux, qui sont :

• **RdP du 1^{er} niveau :**

Les places et les transitions sont explicitées dans la table IV.4.

Le marquage initial du réseau est : $[1 \ 1 \ 0]^T$.

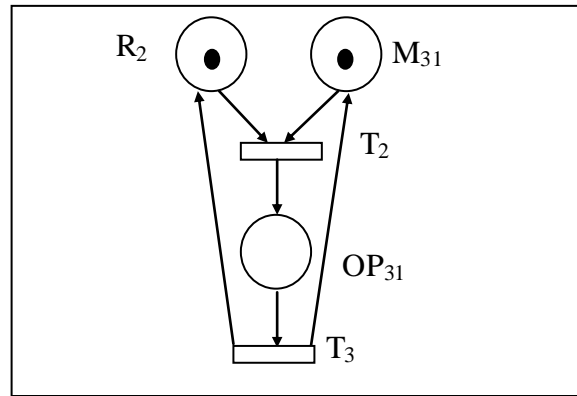


Figure IV.4 : Modèle RdP de convoyeur au 1^{er} niveau.

OP ₃₁	Opération de déplacement de la pièce sur le 1 ^{er} niveau
M ₃₁	Convoyeur vide et disponible au 1 ^{er} niveau
T ₂	Déchargement
T ₃	Début de l'opération de pressage
R ₂	La pièce est prête à être traitée

Table IV.4 : Interprétation des places et transitions.

- **RdP du 2^{em} niveau :**

Les places et les transitions sont explicitées dans la table IV.5.

Le marquage initial du réseau est : $[1 \ 1 \ 0]^T$.

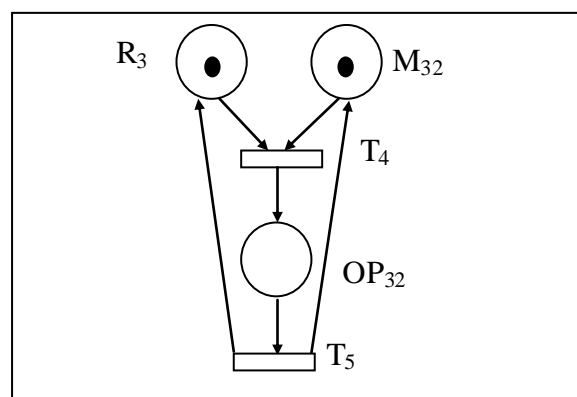


Figure IV.5 : Modèle RdP de convoyeur au 2^{em} niveau.

OP ₃₂	Opération de déplacement de la pièce sur le 2 ^{em} niveau
M ₃₂	Convoyeur vide et disponible au 2 ^{em} niveau
T ₄	Fin de l'opération de pressage
T ₅	Début de l'opération de bordage
R ₃	La pièce est prête à être traiter

Table IV.5 : Interprétation des places et transitions.

- **RdP du 3^{em} niveau :**

Les places et les transitions sont explicitées dans la table IV.6.

Le marquage initial du réseau est : $[1 \ 1 \ 0]^T$.

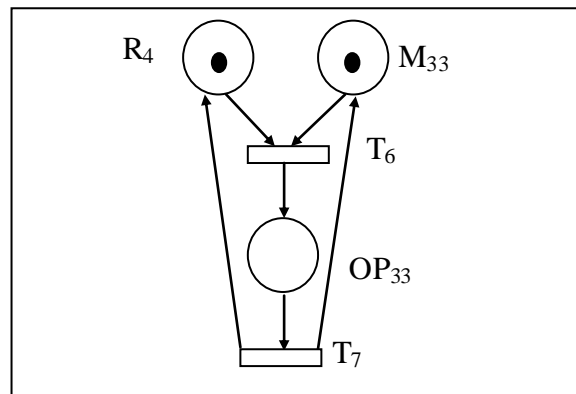


Figure IV.6 : Modèle RdP de convoyeur au 3^{em} niveau.

OP ₃₃	Opération de déplacement de la pièce sur le 3 ^{em} niveau
M ₃₃	Convoyeur vide et disponible au 3 ^{em} niveau
T ₆	Fin de l'opération de bordage
T ₇	Début de l'opération de pressage hydraulique
R ₄	La pièce est prête à être traiter

Table IV.6 : Interprétation des places et transitions.

- **RdP du 4^{em} niveau :**

Les places et les transitions sont explicitées dans la table IV.7.

Le marquage initial du réseau est : $[1 \ 1 \ 0]^T$.

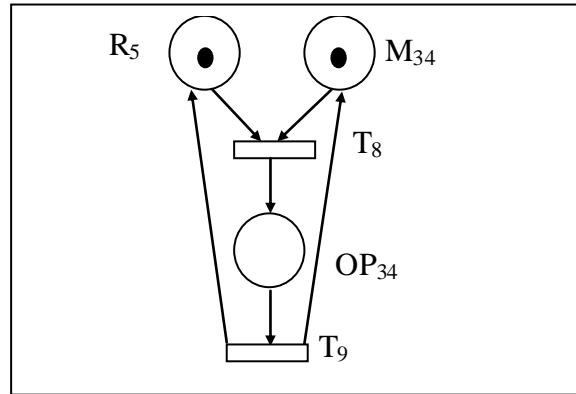


Figure IV.7 : Modèle RdP de convoyeur au 4^{em} niveau.

OP ₃₄	Opération de déplacement de la pièce sur le 4 ^{em} niveau
M ₃₄	Convoyeur vide et disponible au 4 ^{em} niveau
T ₈	Fin de l'opération de pressage hydraulique
T ₉	Début de l'opération de cambrage
R ₅	La pièce est prête à être traiter

Table IV.7 : Interprétation des places et transitions.

- **RdP du 5^{em} niveau :**

Les places et les transitions sont explicitées dans la table IV.8.

Le marquage initial du réseau est : $[1 \ 1 \ 0]^T$.

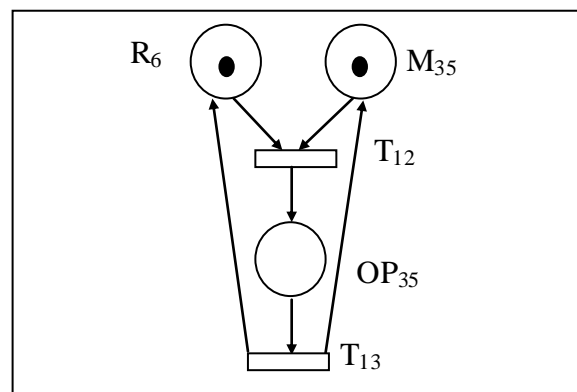


Figure IV.8 : Modèle RdP de convoyeur au 5^{em} niveau.

OP ₃₅	Opération de déplacement de la pièce sur le 5 ^{em} niveau
M ₃₅	Convoyeur vide et disponible au 5 ^{em} niveau
T ₁₂	Fin de l'opération de renversement
T ₁₃	Début de l'opération de soudage
R ₆	La pièce est prête à être traiter

Table IV.8 : Interprétation des places et transitions.

✓ **Modélisation du laminoir :**

Le modèle réseau de Petri du laminoir (ou de l'agent laminoir à profiler à froid) est facile à construire (figure IV.9), les places et les transitions sont explicitées dans la table IV.9

Le marquage initial du réseau est : $[P_2 \ M_4 \ M_{33} \ OP_4 \ M_{32} \ P_3]^T \equiv [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$.

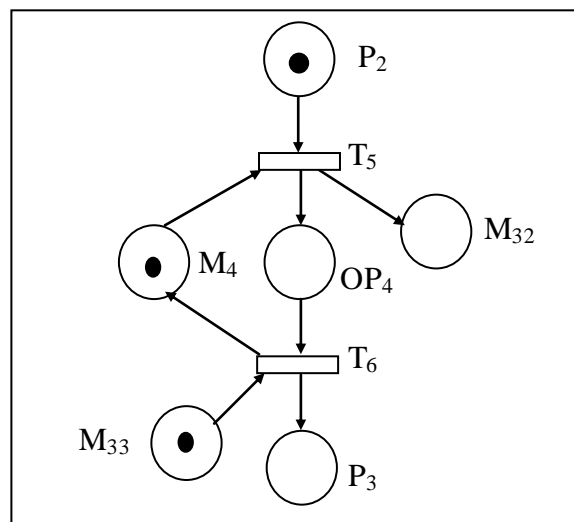


Figure IV.9 : Modèle RdP de laminoir.

M ₄	Disponibilité de laminoir
OP ₄	Opération de bordage
M ₃₂	Convoyeur vide et disponible au 2 ^{em} niveau
M ₃₃	Convoyeur vide et disponible au 3 ^{em} niveau
P ₂	Convoyeur plein au 2 ^{em} niveau
P ₃	L'état du convoyeur au 3 ^{em} niveau
T ₅	Début de l'opération de bordage
T ₆	Fin de l'opération de bordage

Table IV.9: Interprétation des places et transitions.

✓ **Modélisation de la presse hydraulique :**

De la même manière que précédemment on construit le modèle RdP de la presse hydraulique (figure IV.10). Les places et les transitions sont explicitées dans la table IV.10.

Le marquage initial du réseau est : $[P_3 \ M_5 \ M_{34} \ OP_5 \ M_{33} \ P_4]^T \equiv [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$.

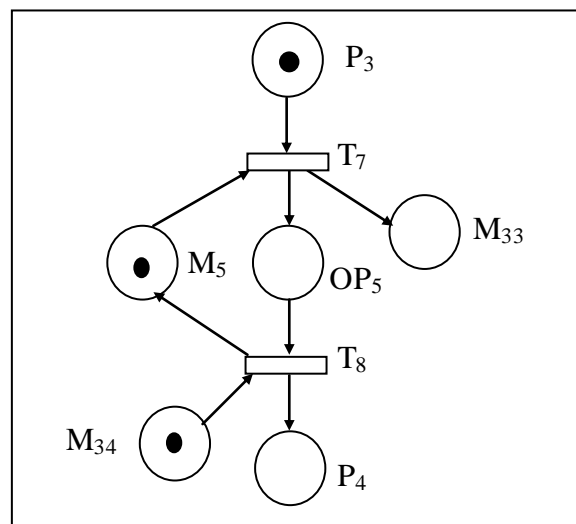


Figure IV.10 : Modèle RdP de la presse hydraulique.

M ₅	Disponibilité de la presse hydraulique
OP ₅	Opération de pressage
M ₃₃	Convoyeur vide et disponible au 3 ^{em} niveau
M ₃₄	Convoyeur vide et disponible au 4 ^{em} niveau
P ₃	Convoyeur plein au 3 ^{em} niveau
P ₄	L'état du convoyeur au 4 ^{em} niveau
T ₇	Début de l'opération de pressage hydraulique
T ₈	Fin de l'opération de pressage hydraulique

Table IV.10 : Interprétation des places et transitions.

✓ **Modélisation de la cintreuse :**

Le modèle RdP de la cintreuse est représenté sur la figure IV.11. Les places et les transitions sont explicitées dans la table IV.11.

Le marquage initial du réseau est : $[P_4 \ M_6 \ OP_6 \ M_{34} \ P_5]^T \equiv [1 \ 1 \ 0 \ 0 \ 0]^T$.

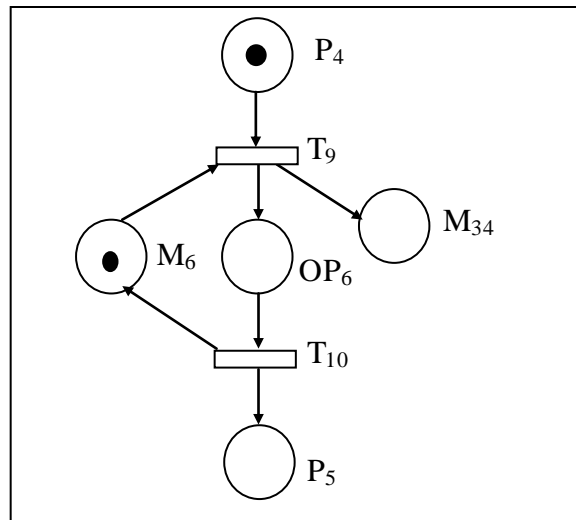


Figure IV.11 : Modèle RdP de la cintruse.

M ₆	Disponibilité de la cintruse
OP ₆	Opération de cambrage
M ₃₄	Convoyeur vide et disponible au 4 ^{em} niveau
P ₄	Convoyeur plein au 4 ^{em} niveau
P ₅	La pièce est prête à être traiter
T ₉	Début de l'opération de cambrage
T ₁₀	Fin de l'opération de cambrage

Table IV.11 : Interprétation des places et transitions.

✓ **Modélisation du dispositif de renversement:**

Le modèle RdP du dispositif de renversement est représenté sur la figure IV.12. Les places et les transitions sont explicitées dans la table IV.12.

Le marquage initial du réseau est : $[P_5 \ M_7 \ M_{35} \ OP_7 \ P_6]^T \equiv [1 \ 1 \ 1 \ 0 \ 0]^T$.

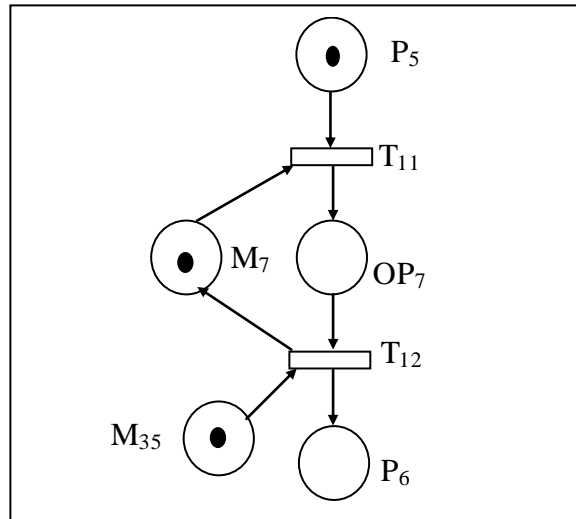


Figure IV.12 : Modèle RdP de dispositif de renversement.

M ₇	Disponibilité de dispositif de renversement
OP ₇	Opération de renversement
M ₃₅	Convoyeur vide et disponible au 5 ^{em} niveau
P ₅	La pièce est prête à être traiter
P ₆	L'état du convoyeur au 5 ^{em} niveau
T ₁₁	Début de l'opération de renversement
T ₁₂	Fin de l'opération de renversement

Table IV.12 : Interprétation des places et transitions.

✓ **Modélisation des différents postes de soudage et le transporteur:**

Dans ce cas, on aura à modéliser les différentes opérations effectuées sur les différents postes de soudage (agent soudeur). Son RdP est donné par la figure IV.13. Les places et les transitions sont explicitées dans la table IV.13.

Le marquage initial du réseau est : $[P_6 \ M_9 \ OP_8 \ M_{35} \ M_8]^T \equiv [1 \ 1 \ 0 \ 0 \ 1]^T$.

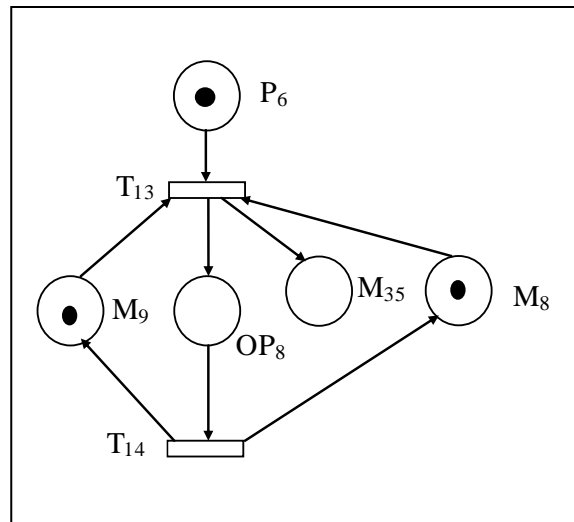


Figure IV.13 : Modèle RdP des différents postes de soudage.

M ₈	Disponibilité du transporteur
M ₉	Opération des machines et des opérateurs
M ₃₅	Convoyeur vide et disponible au 5 ^{em} niveau
OP ₈	Opération de soudage
P ₆	Convoyeur plein au 5 ^{em} niveau
T ₁₃	Début de l'opération de soudage
T ₁₄	Fin de l'opération de soudage

Table IV.13 : Interprétation des places et transitions.

✓ **Modèle réseaux de Petri final de chaine de fabrication :**

Le modèle RdP final est obtenu par l'union de tous les RdP construits pour les différents agents de la chaine à travers des places et / ou des transitions partagées. Le réseau de Petri final, donné par la figure IV.14 est de type graphe d'événement.

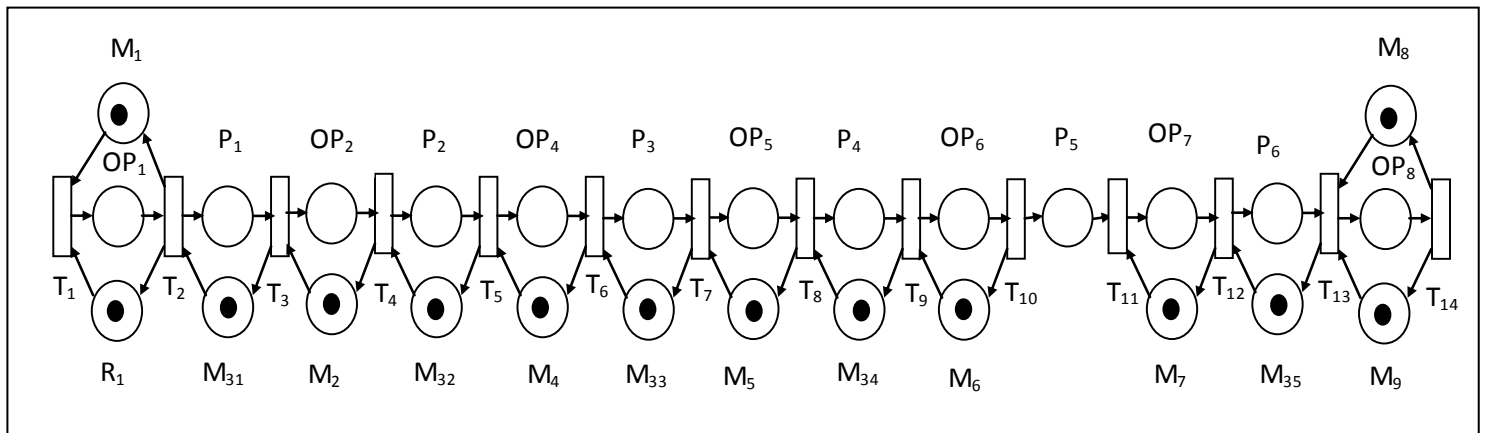


Figure IV.14 : modèle RdP ordinaire de la chaîne de fabrications des armoires frigorifiques.

La modélisation globale du plan de fabrication d'un tel système de production avec un RdP ordinaire nous mène à un RdP de taille très importante et difficile coûteux en temps d'analyse, pour cela, on adaptera le RdP récursif, qui est un RdP de haut niveau dont les transitions et / ou les places sont des actions élémentaire (voir les sections II.1.9 et III.2.3). Par conséquent, on considère, dans ce qui suit, chaque action élémentaire produite par un agent (qu'on a modélisé avec un RdP ordinaire) comme une transition ou une place dans le modèle de notre chaîne (figure IV.15). Les places et les transitions sont explicitées dans la table IV.15.

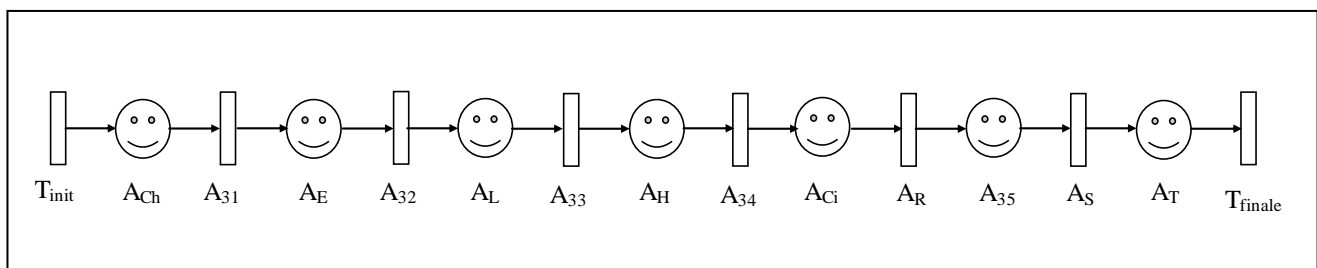


Figure IV.15 : modèle RdP récursif de a chaîne de fabrications des armoires frigorifiques.

T _{ini}	Début de la production (actionné par l'agent planificateur)
A _{Ch}	Agent chargeur
A ₃₁	Agent convoyeur au 1 ^{er} niveau
A _E	Agent d'encochage
A ₃₂	Agent convoyeur au 2 ^{em} niveau
A _L	Agent laminoir
A ₃₃	Agent convoyeur au 3 ^{em} niveau
A _H	Agent presse hydraulique
A ₃₄	Agent convoyeur au 4 ^{em} niveau
A _{Ci}	Agent cintreuse
A _R	Agent de renversement
A ₃₅	Agent convoyeur au 5 ^{em} niveau
A _S	Agent soudeur
A _T	Agent transporteur
A _{finale}	Fin de la production

Table IV.15 : Interprétation des places et transitions.

Validation du modèle de la chaîne frigorifique :

Après construction de différents modèles RdP des différents sous-systèmes (ou agents) de la chaîne, on peut vérifier facilement les propriétés dynamiques et structurelles de chaque sous-réseau et celui du réseau final à l'aide de logiciel de validation *Winrdp*.

Les résultats de validation des différents réseaux des agents sont les suivant (table IV.16) :

Propriété le RdP de	Matrice d'incidence	Bornitude	Quasi- vivacité	Vivacité	Invariant (propriétés structurelle)
L'agent chargeur	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 0 \\ 1 & 1 & -1 & -1 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent d'encoche	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent convoyeur	$W = \begin{bmatrix} -1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	Vivant	Conservatif, non répétitive
L'agent laminoir	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent P- hydraulique	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent cintreuse	$W = \begin{bmatrix} -1 & -1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent de renversement	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & -1 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent soudeur	$W = \begin{bmatrix} -1 & -1 & 1 & 1 & -1 \\ 0 & 1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive

Table IV.16 : Les résultats de la validation des différents agents.

Remarque :

- Pour l'analyse des propriétés de RdP finale on utilise le RdP ordinaire ; le logiciel Winrpd de validation ne dispose pas de RdP récursif.

Interprétation :

Après la validation du modèle RdP globale de la chaîne de fabrication, on conclure que ce modèle est un graphe d'événement, non borné, non réinitialisable et non vivant (car le graphe des marquages n'est pas fortement connexe).

Donc notre modèle du système étudié, avec notre choix des agents, ne répond pas à notre besoin, on aura un problème avec un tel modèle d'analyse par exemple on ne pourra pas déterminer les performances (par exemple le calcul du temps de cycle) de la chaîne de fabrication.

Pour régler ce problème de modélisation, on propose un autre choix des agents pour la modélisation Systèmes Multi- Agents.

b) deuxième identification des agents et des rôles :

Notre solution du problème sera la suivant : l'identification des agents et des rôles dans cette deuxième manière de modéliser le système est la même que la première, la modification sera aux niveaux de la cintruse et le dispositif de renversement, qui va nous conduire à avoir dix types d'agents dans notre système de production, cette solution sera comme suit :

L'agent de renversement sera destiné à piloter *deux* machines, la première est la cintruse qui est un outillage de cambrage et la deuxième qui sert à déplacer la pièce du poste de cambrage au poste suivant. Cet agent reflète l'état de la machine (disponible/occupée), cet agent reçoit l'ordre de l'agent Presse Hydraulique Spéciale via un tableau noir.

Modélisation de la chaîne de fabrication des armoires frigorifiques par un RdP :

La modélisation par RdP de la chaîne de fabrication sera de même que la première modélisation, le changement sera effectué aux niveaux de la modélisation de l'agent de renversement par RdP et de la numérotation des places et des transitions du modèle RdP de l'agent soudeur. Ces deux modèles RdP deviennent :

✓ Modélisation du dispositif de renversement :

Le modèle RdP de la cintruse est représenté sur la figure IV.16. Les places et les transitions sont explicitées dans la table IV.17.

Le marquage initial du réseau est : $[P_4 \ M_6 \ M_{35} \ OP_6 \ M_{34} \ P_5]^T \equiv [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$.

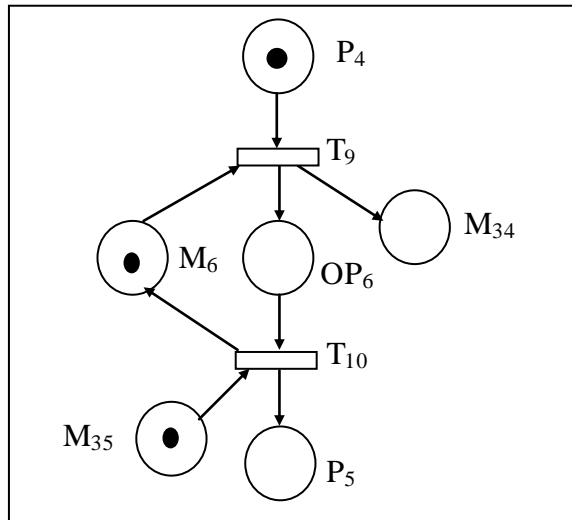


Figure IV.16 : Modèle RdP de dispositif de renversement.

M ₆	Disponibilité de dispositif de renversement
OP ₆	Opération de renversement
M ₃₄	Convoyeur vide et disponible au 4 ^{em} niveau
M ₃₅	Convoyeur vide et disponible au 5 ^{em} niveau
P ₄	Convoyeur plein au 4 ^{em} niveau
P ₅	l'état du convoyeur plein au 5 ^{em} niveau
T ₉	Début de l'opération de reversement
T ₁₀	Fin de l'opération de reversement

Table IV.17 : Interprétation des places et transitions.

✓ **Modélisation des différents postes de soudage :**

Dans ce cas, on aura à modéliser les différentes opérations effectuées sur les différents postes de soudage (agent soudeur). Son RdP est donné par la figure IV.17. Les places et les transitions sont explicitées dans la table IV.18.

Le marquage initial du réseau est : $[P_5 \ M_8 \ OP_7 \ M_{35} \ M_7]^T \equiv [1 \ 1 \ 0 \ 0 \ 1]^T$.

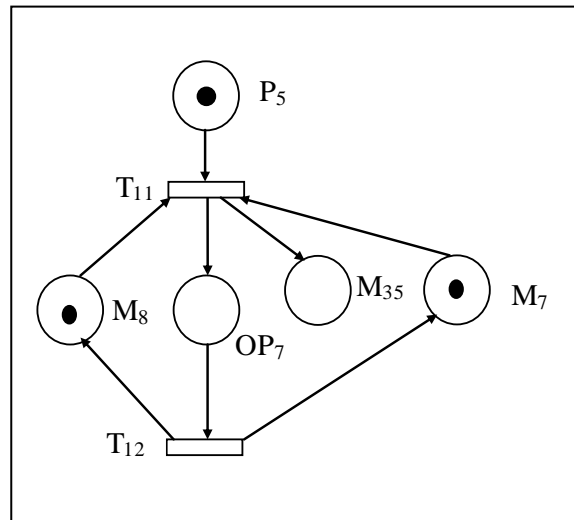


Figure IV.17 : Modèle RdP des différents postes de soudage.

M ₇	Disponibilité du transporteur
M ₈	Opération des machines et des opérateurs
M ₃₅	Convoyeur vide et disponible au 5 ^{em} niveau
OP ₇	Opération de soudage
P ₅	Convoyeur plein au 5 ^{em} niveau
T ₁₁	Début de l'opération de soudage
T ₁₂	Fin de l'opération de soudage

Table IV.18 : Interprétation des places et transitions.

✓ **Modèle réseaux de Petri final de chaine de fabrication :**

Le modèle RdP final est obtenu par l'union de tous les RdP construits pour les différents agents de la chaine à travers des places et / ou des transitions partagées. Le réseau de Petri final, donné par la figure IV.18 est de type graphe d'événement.

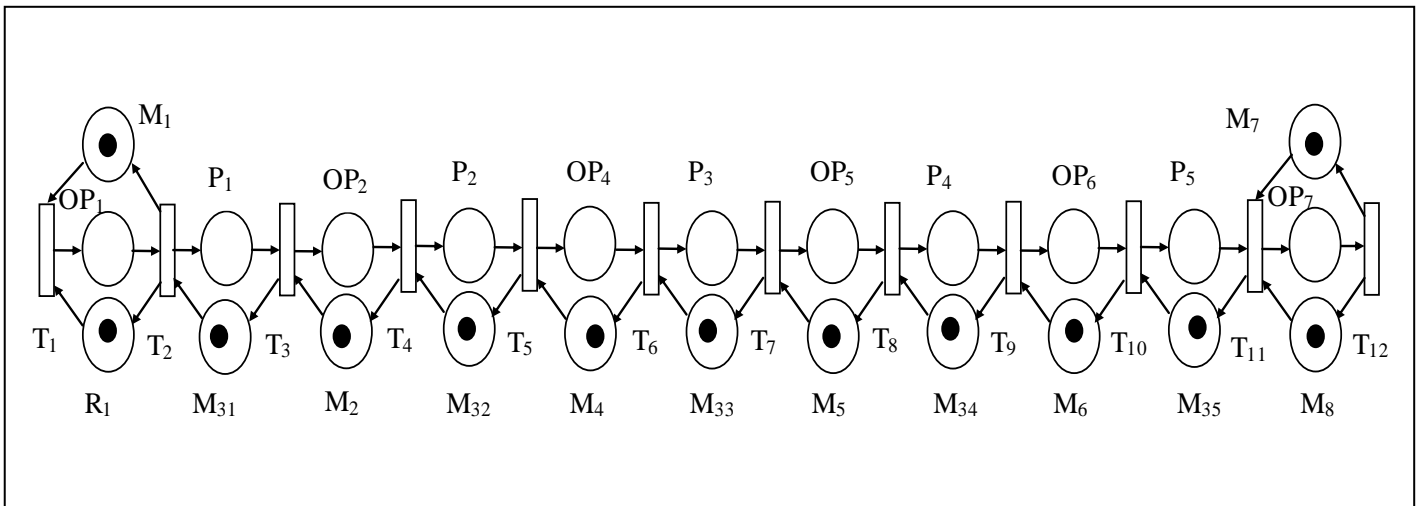


Figure IV.18 : modèle RdP récursif de la chaîne de fabrications des armoires frigorifiques.

La modélisation global du plan de fabrication de tel système de production avec un RdP ordinaire nous mène à un RdP de taille très importante et difficile ainsi couteux en temps d’analyse, pour cela, on adaptera le RdP récursif, qui est un RdP de haut niveau dont les transitions et / ou des places sont des actions élémentaire (voir les sections II.1.9 et III.2.3). Par conséquence, on considère, dans ce qui suit, chaque action élémentaire produite par un agent (qu’on a modélisé avec un RdP ordinaire) comme une transition ou une place dans le modèle de notre chaîne (figure IV.19). Les places et les transitions sont explicitées dans la table IV.19.

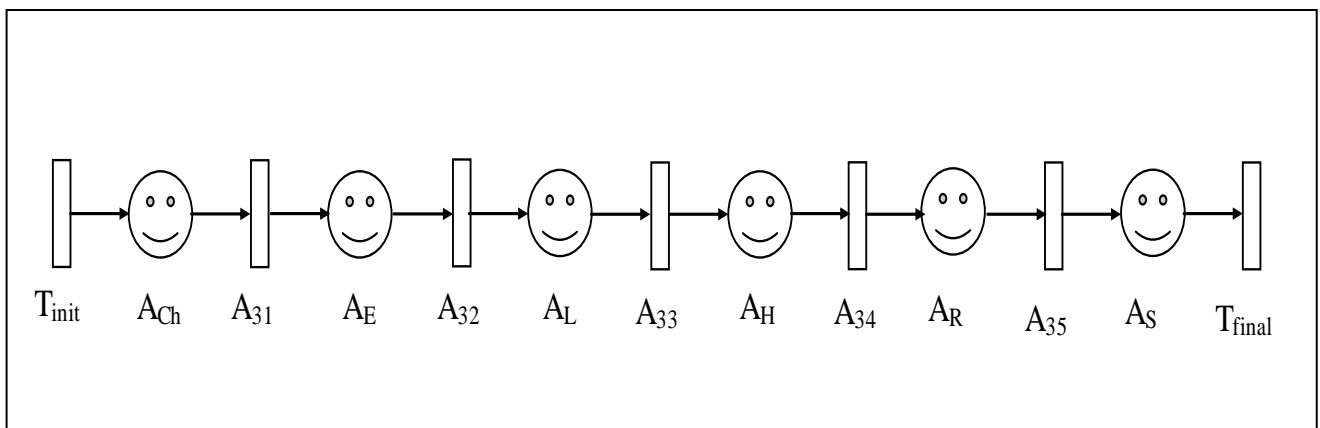


Figure IV.19 : modèle RdP récursif de la chaîne de fabrications des armoires frigorifiques.

T _{ini}	Début de la production (actionné par l'agent planificateur)
A _{Ch}	Agent chargeur
A ₃₁	Agent convoyeur au 1 ^{er} niveau
A _E	Agent d'encochage
A ₃₂	Agent convoyeur au 2 ^{em} niveau
A _L	Agent laminoir
A ₃₃	Agent convoyeur au 3 ^{em} niveau
A _H	Agent presse hydraulique
A ₃₄	Agent convoyeur au 4 ^{em} niveau
A _R	Agent de renversement
A ₃₅	Agent convoyeur au 5 ^{em} niveau
A _S	Agent soudeur
A _{finale}	Fin de la production (actionné par l'agent transporteur)

Table IV.19 : Interprétation des places et transitions.

Validation du modèle de la chaîne frigorifique :

Après construction de différents modèles RdP des différents sous-systèmes (ou agents) de la chaîne, on peut vérifier facilement les propriétés dynamiques et structurelles de chaque sous-réseau et celui du réseau final à l'aide de logiciel de validation *Winrdp*.

Les résultats de validation des différents réseaux des agents sont les suivant (table IV.20) :

Propriété le RdP de	Matrice d'incidence	Bornitude	Quasi- vivacité	Vivacité	Invariant (propriétés structurelle)
L'agent chargeur	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 0 \\ 1 & 1 & -1 & -1 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent d'encoche	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent convoyeur	$W = \begin{bmatrix} -1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	Vivant	Conservatif, non répétitive
L'agent laminoir	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent P- hydraulique	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent de renversement	$W = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive
L'agent soudeur	$W = \begin{bmatrix} -1 & -1 & 1 & 1 & -1 \\ 0 & 1 & -1 & 0 & 1 \end{bmatrix}^T$	Borné, sauf	Quasi- vivant	non vivant	Conservatif, non répétitive

Table IV.20 : Les résultats de la validation des différents agents.

Remarque :

- Pour l'analyse des propriétés de RdP finale on utilise le RdP ordinaire ; le logiciel Winrdp de validation ne dispose pas de RdP récursif.

Interprétation :

Le réseau de Petri global obtenu est un graphe d'événement, borné, vivant et reinitialisable. Nous n'avons donc aucun problème au niveau de la conception et au niveau de d'analyse des performances du système.

Donc notre modèle du système étudié, avec notre choix des agents, répond parfaitement à nos besoins, on n'aura aucun problème au niveau d'analyse des performances (par exemple le calcul du temps de cycle) de la chaîne de fabrication.

L'interaction entre deux agents différents de deux éléments différents dans la chaîne de production est représentée par le diagramme d'interaction de la Figure IV.20.

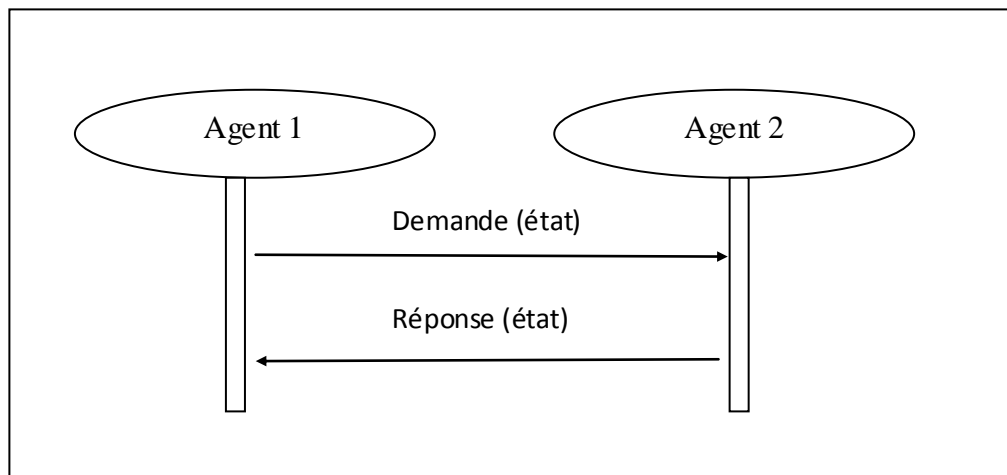


Figure IV.20 : Diagramme d'interaction.

Les tâches effectuées dans le système sont :

- Traitement des commandes client.
- Fabrication du produit selon un plan de fabrication (plan d'action) décrivant l'enchaînement des opérations OP_i de sa gamme opératoire.

c) *Scénario d'allocation*

Soit le contenu du tableau noir à un instant donné, illustré dans la Figure (IV.21). L'agent planificateur pose le plan de fabrication du produit dans le tableau noir (Figure IV.21(a)). Les différents agents consultent le tableau noir, détectent la disponibilité des machines, et posent leurs propositions (les meilleures selon leurs ressources) dans la partie propositions dans le tableau noir (Figure IV.21 (b)).

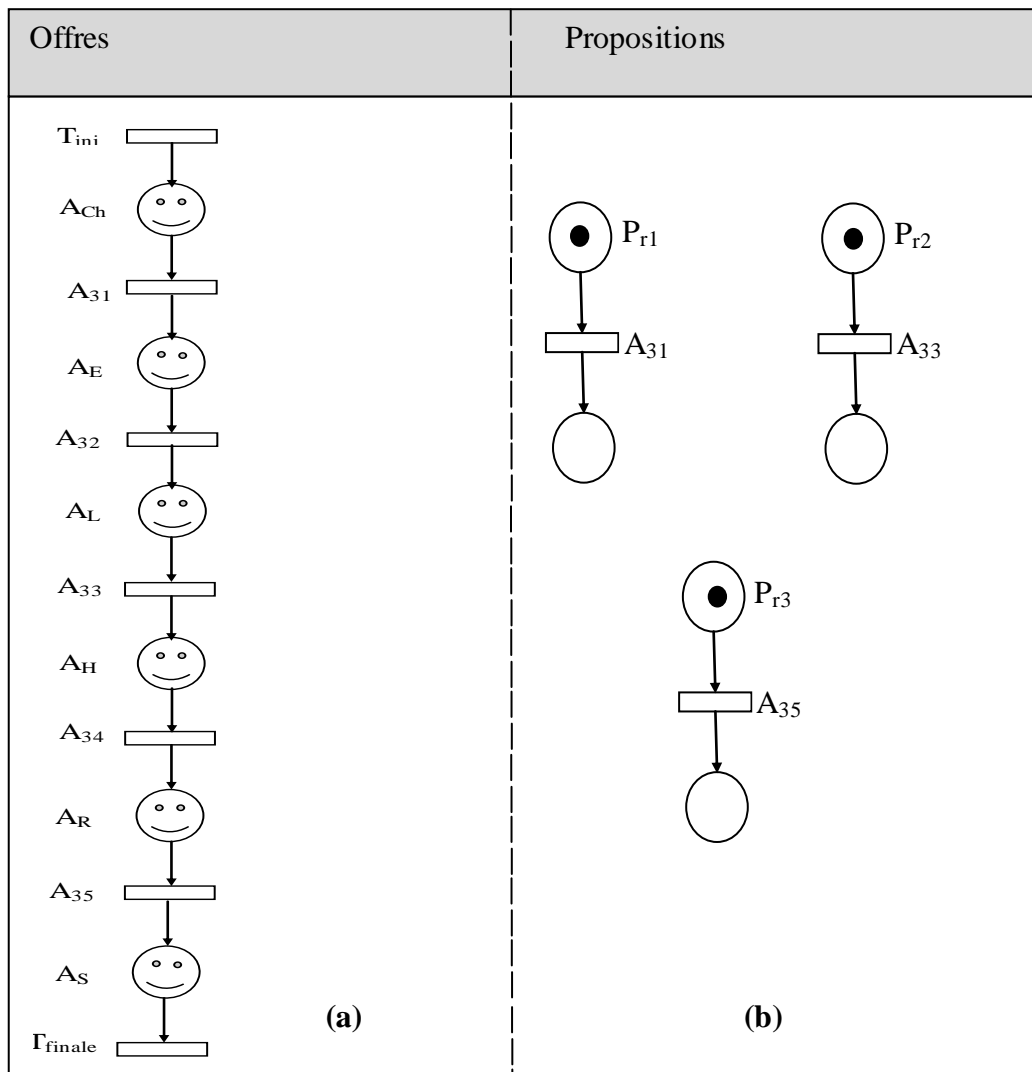


Figure IV.21 : Contenu du tableau noir à un instant t .

Fusion des modèles Rdp de tâches et Rdp des propositions :

Nous considérons que toutes les propositions illustrées dans la Figure IV.22 sont jugées intéressantes. Elles vont donc être destinées à une évaluation collective. Pour ce faire l'agent manager (le planificateur) doit fusionner tous le modèles Rdp pour en avoir qu'un seul.

Si on a plusieurs propositions concernant une seule transition, la fusion se fait en deux étapes : i) fusion de propositions concernant la même opération, ii) fusion des modèles résultant de la fusion précédente et le modèle Rdp de la tâche.

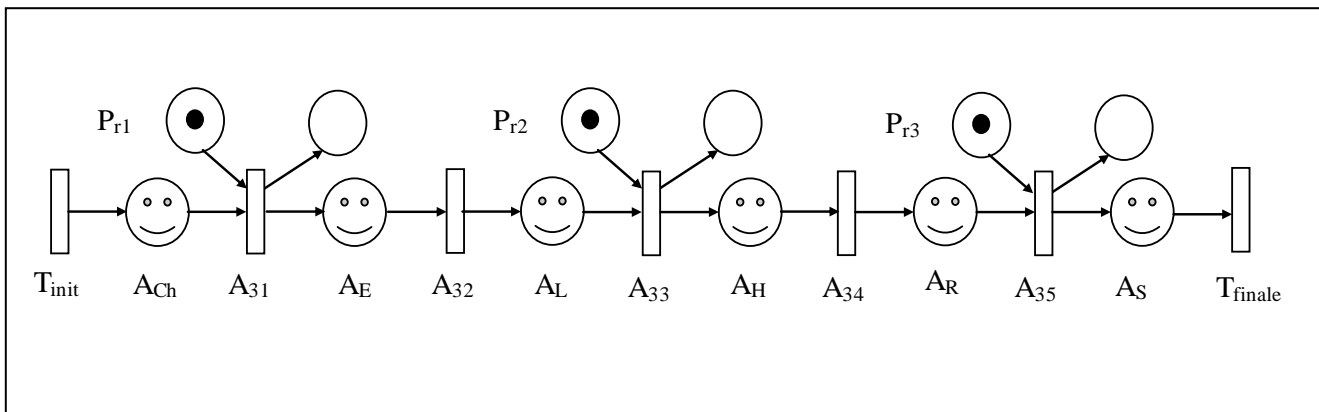


Figure IV.22 : Fusion des modèles RDP

CONCLUSION

Nous avons présenté dans ce chapitre une étude de cas. La première partie de l'étude consiste à appliquer les RDP pour modéliser et valider les interactions entre agents à l'aide d'un logiciel de validation qui est *Winrdp*. Le but était de montrer à travers des scénarios comment utiliser les modèles RDP de protocoles d'interaction établis dans la phase de pré-implémentation pour valider les interactions entre agents.

Dans deuxième partie de cette étude, nous nous sommes intéressés à l'utilisation de RDP pour modéliser les plans d'actions. Le but s'était de montrer la faisabilité du mécanisme d'allocation de tâches que nous avons proposé.

Notre objectif principal est de mettre en évidence l'intérêt et le principe d'utilisation des RDP dans le contexte des SMA.

Conclusion

Générale

Conclusion Générale

L'objectif de ce travail était l'intégration des réseaux de Petri (RdP) comme approche orientée analyse dans la modélisation des Systèmes Multi-Agents (SMA). Pour pouvoir faire une telle intégration, nous avons proposé la modélisation de certains aspects SMA par les RdP.

Dans les deux premiers chapitres nous avons présenté respectivement le domaine des SMA et le formalisme des RdP. L'étude du domaine de SMA nous a permis de constater que, les SMA s'intéressent aux comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelées agents, que ces interactions tournent autour de la coopération, de la concurrence ou de la coexistence entre ces agents. Le domaine des SMA est un domaine rempli de défis à surmonter, autrement dit c'est un domaine très ouvert pour la recherche.

Comme nous l'avons vu dans le deuxième chapitre, les RdP sont un ensemble d'outils en continuelle évolution permettant à la fois la modélisation et l'analyse de la dynamique des systèmes complexes. Ils ont le double avantage de fournir à la fois une représentation graphique, et de posséder des fondations mathématiques permettant d'étudier des propriétés analytiques pour évaluer le comportement du système modélisé.

Il est intéressant de noter encore une fois, que notre but n'est plus de modéliser tous le SMA par les RdP, pour les deux raisons suivantes : i) la richesse d'un SMA fait que aucun formalisme ne peut couvrir tous ses aspects ; ii) voir tous le système comme un ensemble de RdP diminue la lisibilité et impose une solide maîtrise de RdP ce qui peut handicaper certains concepteurs. La nécessité donc d'utiliser plus qu'une approches pour la spécification et la modélisation des SMA est par conséquent un besoin inéluctable.

Différents usages des RdP sont possibles dans le contexte des SMA. On peut représenter un agent par un RdP, certains modules de l'agent, modéliser un ensemble d'agent, modéliser les interactions entre agents,...etc.

Dans ce travail, on s'était intéressé à la modélisation :

De l'entité agent par les RdP, nous avons présenté une modélisation d'un agent réactif. Chaque module de cette architecture a été représenté par un RdP. L'assemblage de tous les modules forme un RdP qui modélise le comportement global de l'agent.

Des interactions entre agents, nous avons présenté les différents éléments de modélisation qui peuvent aider le concepteur à élaborer les modèles RdP des protocoles d'interaction. Nous avons ainsi proposé une approche de validation des interactions entre agents en utilisant les modèles RdP des protocoles d'interaction.

L'utilisation des RdP pour la modélisation des SMA est une idée intéressante dans le sens où la symbolique assez simple des RdP permet une compréhension rapide de l'architecture globale. Elle permet entre autre d'entamer une phase d'analyse et de bénéficier des différents outils disponibles pour l'édition, l'analyse et la validation de modèles conçus. Ainsi, on peut bénéficier de la documentation produite dans la phase pré-implémentation (modèles RdP) pour valider le comportement des agents par eux même.

Dans l'état actuel de cette étude, nous pouvons envisager plusieurs perspectives, en voici quelques unes :

La prise en compte d'autres architectures de l'agent, tel que l'architecture BDI. Les différentes vues de l'agent ; la vue modulaire de Conclusion et perspectives de l'agent, dans ce cas l'agent est vu comme un ensemble de composants réactifs ou actifs en interaction entre eux. On peut représenter chaque composant par un RdP, cela permet d'entamer une phase de validation. La modularité offre plusieurs avantages, entre autre la réutilisation des composants validés (qui peuvent être regroupés dans une bibliothèque). Elle offre la possibilité de définir des agents à granularité variable, la possibilité de définir des agents à structure adaptative. Chaque agent peut dynamiquement changer ces composants ainsi que les relations entre ces différents composants. Comment tenir compte de cette notion d'adaptation et d'ouverture associée au concept agent lui même et la forte dynamique des SMA en générale par les RdP reste une question ouverte.

Si les RdP sont assez simples à comprendre, il s'avère difficile de modéliser un comportement à cause de la grande souplesse et du grand pouvoir d'expression du modèle. Cela requiert parfois un peu d'expertise. Il nous semble être nécessaire de fournir un guide de modélisation pour tout aspect à modéliser par les RdP qui peut aider le concepteur à élaborer son modèle RdP.

Enfin notre travail, n'est qu'un point de départ d'une intégration effective de la modélisation par RdP dans une méthodologie de développement des SMA.

Bibliographie

Bibliographie

- [1] **I. BAKAM TCHIAKAN**, “ *Des Systèmes Multi-Agents aux Réseaux de Petri pour la Gestion de Ressources Naturelles : le cas de la forme à l’Est-Cameroun.* “, Doctorat de l’Université YAOUDE I, 2002-2003.
- [2] **Jean-Michel CONTET**, “ *Modèles Multi-Agents Réactifs pour la Navigation Multi-Véhicules : spécification formelle et vérification.* “, Doctorat de l’Université de Technologie de BELFORT-MONTBELIARD, 2009.
- [3] **Christine BALAGUE**, “ *Les Systèmes Multi-Agents en Marketing : modélisation par Réseaux de Petri.* “, Ecole des Hautes Etudes Commerciales, 2005.
- [4] **François KLEIN**, “ *Contrôle d’un Système Multi-Agent Réactif par Modélisation et Apprentissage de sa Dynamique Globale.* “, Doctorat de l’Université NANCY II, 2009.
- [5] **Raida EIMANSOORI**, “*Modélisation et Vérification des processus Métiers dans les Entreprises Virtuelles : une approche basée sur la transformation de graphes.* “, Doctorat de l’Université MENTORI CONSTANTINE.
- [6] **NGUYEN Thi Truc Vien**, “ *Modélisation et Implémentation d’Un Système Distribué de Pages Blanches au Sein d’Une Plateforme Multi-Agent.* “, Centre de Recherche Informatique de MONTREAL, 2003.
- [7] **Jean-Pierre BRIOT**, “ *Généralisation Cartographique par Agents Communicants : le modèle Cartacon* “, Doctorat de l’Université PARIS 6, 2004.
- [8] **Cyril DROCOURT**, “ *Localisation et Modélisation de l’Environnement d’un Robot Mobile par Coopération de deux Capteurs Omnidirectionnels* “, Doctorat de l’Université de Technologie de COMPIEGNE, 2002.
- [9] **Rémi BASTIDE**, “ *Spécification Comportementale par Réseaux de Petri : Application aux systèmes distribués à objets et aux systèmes interactifs* “, Doctorat de l’Université TOULOUSE I, 2000.

- [10] **Jean Paul SANONNET**, “ *Réutilisation des Protocoles d’Interaction et Démarche Orienté Modèles pour le Développement Multi-Agents* “, Doctorat de l’Université de REINS, 2006.
- [11] **Lilia REJEB**, “*Simulation Multi-Agents de Modèles Economiques Vers des Systèmes Multi-Agents Adaptatifs* “, Doctorat de l’Université de REINS, 2005.
- [12] **Houda HAIOUNI**, ” *Approche Mixte de Modélisation par RdP et SMA* “, Ecole de Doctorat en Informatique de l’Est Pole de Constantine, 2010.
- [13] **Philippe RONGIER**, “ *Développement et Evaluation de Méthodes Multi-Agents pour les Applications aux Véhicules Robotisés* “, Doctorat de l’Université de MONTPELLIER II, 2000.
- [14] **Abdelaziz HAMMOUCHE**, “ *Systèmes Multi-Agents Appliquées en Segmentation d’Images* “, Magister de Université Mouloud Mammeri de TIZI-OUZOU, 2008.
- [15] **S.CHERIEF et S.RACHEDI**, “ *Logiciel d’Aide à l’Evaluation des Performances des Systèmes de Production Modélisé par de Graphes d’Evénements Temporisés* “, thèse d’Ingénieur, Université Mouloud Mammeri de TIZI-OUZOU, 2000.
- [16] **Y. HADDAD**, ” *Logiciel de Validation et de Simulation de Systèmes Flexibles de Production Modélisés par Réseaux de Petri* “, thèse d’Ingénieur, Université Mouloud Mammeri de TIZI-OUZOU, 1997.
- [17] **Jacques FERBER**, “ *Les Systèmes Multi-Agents : vers une intelligence collective* “, Inter Editions, Paris, 1995.
- [18] **Choquet GENIE ANNIE**, “ *Les Réseaux de Petri : un outil de modélisation, cours et exercice* “, Edition Dunod, Paris, 2006.
- [19] **R.DAVID et H.ALLA**, “ *Du Grafset aux Réseaux de Petri* “, Edition Hermes, Paris, 1997.

- [20] **Pierre LADET**, “ *Réseaux de Petri* “, Technique d’Ingénieur, Vol S2, Num : R7252.
- [21] **Pascale RICHARD et Christophe HARO**, “ *Application des Réseaux de Petri* “, Technique d’Ingénieur, Vol S2, Num : S7254.
- [22] **Renné MANDIAU et Emmanuelle GRISLINLE STRUGEON**, “*Systèmes Multi-Agents* “, Technique d’Ingénieur, Vol S1, Num : S7216.
- [23] **François BARTHELEMY**, “ *Les Systèmes Multi-Agents Basés sur des Réseaux de Petri Colorés : technique de pointe ou technologie dépassée ?* “, Institut d’Informatique des Facultés Notre-Dame de la Paris (FUNDP), Namur (Belgique).
- [24] **Olivier GUTKNECHT, Fabien MICHEL et Jacques FERBER**, “*Madkit : une architecture de plate-forme multi-agent générique* “, Université Montpellier II, R.R.LIRMME 00061, 2000.
- [25] **Guillaume HUTZER, Hanna KLAUDEL et Dong Yue WANG**, “ *Automates Temporisés et Systèmes Multi-Agents Temps-Réel* “, Université Evry-Val d’Essonne/CNRS.
- [26] **J.F. PRADAT-PEYRE**, “ *Spécification des Problèmes Concurrents : modélisation avec les réseaux de petri et les réseaux de haut-niveau.* “, 2002.
- [27] **F.Raclot, D.Andreu, T.Libourel, R.Passama**, “*E-NetObject : un éditeur de réseaux de petri à objets.* “, Université Montpellier II, Rapport LIRMM n^o 02180.
- [27] **B.Chaib-Draa**, “*Systèmes Multi-Agents* “, Université Laval, Canada.
- [28] **G.SCORLETTI, G.BINET et E.MAGAROTTO**, “*Réseaux de Petri* “, Cours Master I Académique, Université de Caen, 2006.
- [29] **R.KARA**, “ *Productique.* “, Cours Master II Académique, Université de Mouloud Mammeri, Tizi-Ouzou, 2011.

[30] **N.SABOURET**, “ *Systèmes Multi-Agents* “, Cours de l’Université Sorbonne, 2010/2011.

[31] **Achour H**, “ *Automatismes Industriels (Partie RdP)* “,

<http://dl.getdrophax.com/u/1627779/RDP>