REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET TECHNOLOGIE
« HOUARI BOUMEDIENE »
FACULTE DE L'ELECTRONIQUE ET L'INFORMATIQUE

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

EN : INFORMATIQUE

Spécialité: Programmation & Système

Par : M^r Mansouri Djamel

Formalisation des Systèmes Multi-Agents par Les Réseaux de Petri

Soutenu le : 09/07/2005, devant le juré composé de :

M^r Ahmed Nacer. M :	Professeur,	USTHB,	Président
M^me Boukala. M :	Maître de Conférence,	USTHB,	Directeur de Thèse
M^r Azoune. H :	Maître de Conférence,	USTHB,	Examinateur
M^r Boukala. M.C :	Chargé de Cours,	USTHB,	Invité

SOMMAIRE

Chapitre III : Approche Système Multi-Agents Réseaux de Pétri

Chapitre IV : La Modélisation SMA

Introduction Générale

Les systèmes multi-agents constituent une nouvelle technique de modélisation qui place l'objet d'étude au centre de sa démarche. Ces modèles représentent les actions, individuelles, les interactions entre les acteurs et les conséquences de ces interactions sur le fonctionnement dynamique du système.

Les systèmes multi-agents semblent être une approche adéquate pour la modélisation des systèmes distribués, coopératifs, et des systèmes ouverts. En effet, ils ont une vocation de parallélisme, de résistance aux pannes et aux interférences, d'adaptation et d'extensibilité.

L'étude des systèmes coopératifs, répartis et parallèles, nécessite la définitions des modèles, de développer des méthodes et de réaliser des outils permettant à la fois de concevoir, de valider et d'évaluer les architectures proposées, tant d'un point de vue qualitatif (vérification) que quantitatif (évaluation).

Toutefois, la réalisation d'un système modélisé comprend les phases de spécification, de conception, d'implémentation, de validation et de tests. On part généralement d'une description informelle du système, utilisant le langage naturel. Le passage à la phase de conception est intuitif et rien ne garantit que le système ainsi conçu corresponde bien aux spécifications qui ont été posées. Lorsque la fiabilité du système est un enjeu capital, il devient nécessaire de partir d'une spécification formelle

Les intérêts d'une approche formelle sont principalement une compréhension approfondie du système par la mise en évidence des ambiguïtés laissées dans l'ombre par les spécifications informelles, l'utilisation d'outils de preuve et d'évaluation, la possibilité de raffiner jusqu'à l'obtention d'une implémentation abstraite et la possibilité d'engendrer automatiquement des jeux de tests.

Dans le cas des SMA, la spécification formelle correspond à l'élaboration d'un modèle, c'est-à-dire une abstraction du réel. Un enjeu important est de proposer des formalismes qui permettent l'expression cohérente des différents aspects relatifs aux SMA à savoir le raisonnement, le contrôle, la synchronisation des agents ou encore les contraintes temporelles, spatiales, etc. De plus, il faut que des propriétés de bon fonctionnement ou de sécurité soient vérifiables dans ces formalismes.

Les réseaux de Pétri constituent un formalisme qui s'apprête bien à la formalisation des systèmes multi-agents, car d'une part, ils permettent de représenter très bien le parallélisme, une des caractéristiques principales des SMA et d'autre part ils fournissent des méthodes d'analyse permettant de vérifier et d'analyser certaines propriétés générales (absence de blocage, équité, ...) et des propriétés spécifiques.

L'objectif du travail présenté dans ce rapport est de faire une étude sur l'utilisation des réseaux de Pétri (réseaux de haut niveau) pour la formalisation des systèmes multi-agents, étant donné que les SMA offre un riche pouvoir d'expression mais ne permettent pas de vérification formelle.

Dans le premier chapitre on présentera les différents concepts de base relatifs aux systèmes multi-agents à savoir le concept agent, les architectures, les interactions, ainsi que le besoin de la formalisation des systèmes multi-agents.

Le deuxième chapitre sera consacré à une présentation des notions de base des réseaux de Pétri et leurs propriétés fondamentales.

Quant au troisième chapitre, il sera consacré à la présentation des différents formalismes des systèmes mutli-agents à base de réseaux de Pétri.

Dans le quatrième chapitre nous présenterons une description d'agent et de ses systèmes, qui comporte notamment une description formelle des éléments de l'architecture des agents cognitifs et réactifs sous forme de réseaux de Petri, en plus une formalisation des différentes formes d'interactions entre les agents à savoir la coopération, la coordination d'actions et la négociation.

Enfin, on donnera une conclusion, dans laquelle seront présentées les perspectives.

Chapitre I
Les Systèmes Multi-agents

I-1- Introduction

Les recherches issues de l'intelligence artificielle distribuée (IAD) et en particulier les systèmes multi-agents sont apparues aux Etat-Unis à la fin des années 80. Les premières applications de laboratoire furent le développement d'applications de reconnaissance vocale. Depuis, ce domaine a eu un essor formidable et a amené un certain renouveau de l'intelligence artificielle traditionnelle. Dans les années 1990 une volonté de formalisation a permis l'introduction de nouveaux concepts et l'extension des travaux précédents vers d'autres types d'applications.

Actuellement des applications industrielles apparaissent et nous nous apercevons non seulement des problèmes théoriques sous -jacents mais aussi la nécessite d'une méthodologie dans la conception de tels systèmes. Cet essor croissant est sans doute lié principalement à l'intégration de différents facteurs tels que, l'intérêt théorique de cette nouvelle problématique et l'intégration de nouvelles technologies de l'informatique dans la vie courante.

Les systèmes multi-agents 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 [12]. 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 monde hypothétique [26].

En général, les systèmes multi-agents et l'IAD permettent d'appréhender des problèmes de taille et surtout de complexité plus importante que ceux abordés par les approches classiques de l'Intelligence Artificielle. L'IAD s'appuie sur la distribution de l'intelligence sur des agents formant une société dans laquelle chacun a une certaine autonomie.

Les systèmes d'IAD ne sont plus fermés sur leurs propres connaissances, mais sont de véritables sociétés d'individus qui doivent se mouvoir, planifier, agir et réagir dans un milieu, dans lequel ils entrent parfois en conflit avec d'autres agents. Un agent "vit" au sein d'une communauté d'agents appelée système ou univers multi agents. Les agents peuvent être nombreux ou non. Ils peuvent avoir des objectifs individuels ou un objectif global à atteindre.

Pour réaliser leurs objectifs, les agents peuvent coopérer, entrer en conflit, négocier,…, etc. Un système multi-agents est alors un espace virtuel ou non, composé d'un environnement, d'objets passifs et d'agents communiquant directement ou indirectement et pouvant opérer sur les objets de l'environnement.

De la notion de systèmes multi-agents se dégage immédiatement l'idée d'un système qui est constitué de plusieurs agents. Le concept d'agent reste donc le pivot de ce domaine et l'interprétation que l'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 cerner comment les systèmes multi-agents sont construits.

I-2- Les agents : Définitions

Le concept d'agent est le concept de base du paradigme multi-agents. Différentes définitions de ce concept ont été données dans la littérature.

Un agent peut être défini comme une entité active physique (capteurs, processeurs...) ou abstraite (tâches à réaliser, déplacement...), capable d'agir sur lui même ou sur son environnement, disposant d'une représentation partielle de cet environnement, pouvant communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec d'autres agents [40].

De nombreuses définitions du terme agent existent, aucune ne fait réellement l'unanimité jusqu'à présent en raison de la grande variété d'entités dénommées agents. Certaines proposent de ne pas dissocier l'agent du Système Multi-Agents et de l'environnement. D'autres réduisent l'agent à un simple processus communicant. Des définitions générales communément admises sont ici proposées. Nous nous appuierons dans ce document sur les définitions utilisées dans la communauté des chercheurs en SMA qui considèrent qu'un agent représente une entité qui agit, plus ou moins de façon autonome, qui est capable de modifier son environnement, et qui est un composant d'un système multi-agents.

I-2-1- Définition de Russell & Norving

L'intelligence artificielle a pour but de reproduire l'intelligence humaine au niveau des entités informatiques. Si ces entités sont capables de percevoir et d'agir au sein d'un environnement, nous pouvons les appeler agents [52].

Russell & Norving définissent donc un agent comme étant n'importe quelle entité qui perçoit son environnement au travers de ses capteurs et sur qui elle agit via ses effecteurs. Un être humain peut donc être vu comme un agent. Un robot muni de caméras, de capteurs infrarouges,…, et de différents moteurs pour générer des actions, peut être aussi considéré comme un agent.

Cette définition nous permet d'introduire la notion d'interaction entre l'agent et l'environnement. Cette interaction peut être définie comme étant l'influence mutuelle entre ces deux entités, qui est établie via les processus de perception de l'agent de son environnement et de son action sur ce dernier.

La définition de Russell et Norving introduit également la notion d'environnement sans la définir. Elle est aussi très générale, ce qui permet d'identifier les agents dans plusieurs domaines d'application. Mais, nous pensons que les entités qui sont en interaction avec l'environnement doivent avoir certaines propriétés afin de pouvoir être appelées "agents". Nous présentons donc dans ce qui suit d'autres définitions pour l'agent, basées sur leur propriétés, leur caractéristiques et leur fonctionnalités.

I-2-2- Définition d' Erceau

"Les agents sont des entités physiques (capteurs, processeurs,...) ou abstraites (tâches à réaliser, déplacements,...), qui sont capables d'agir sur leur environnement et sur elles-mêmes, c'est-à-dire de modifier leur propre comportement. Pour ce faire, elles disposent, d'une représentation partielle de cet environnement et de moyens de perception et de communication [21].

I-2-3- Définition de Ferber

Jacques Ferber, propose cette définition [26] : On appelle agent une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement ;
- qui peut communiquer directement avec d'autres agents ;
- 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) ;
- qui possède des ressources propres ;
- qui est capable de percevoir (mais de manière limitée) son environnement.
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune) ;
- qui possède des compétences et offre des services ;
- qui peut éventuellement se reproduire, mourir et changer d'état ;
- 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.

I-2-4- Définition de Maes

Maes [43] propose une bonne définition générique du terme agent: "An agent is a computational system that inhabits a complex, dynamic environment. The agent can sense, and act on, its environment, and has a set of goals or motivations that it tries to achieve through these actions."

Il ressort de cette définition trois notions fondamentales permettant de caractériser un agent : percevoir, décider et agir :

Percevoir **:** Le fait de percevoir implique le fait que l'agent soit plongé dans un univers et qu'il communique dans cet univers. Il peut ou non percevoir les actions des autres agents, percevoir des mouvements d'objets ou tout type de transformation de son environnement.

Décider **:** Cela induit le fait qu'un agent ait des connaissances qui lui sont propres et un comportement autonome lui permettant d'exploiter ses connaissances et de raisonner. Il décide en fonction de ses perceptions.

Agir **:** L'agent peut agir sur son environnement. Il peut manipuler des objets, les déformer, communiquer avec d'autres agents, engager une coopération ou un conflit. Toute action en tant que telle modifie d'une certaine façon une partie de l'environnement.

I-2-5- Définition de Jenning & Wooldridge

Jennings & Wooldridge [58] ont proposé la définition suivante pour un agent: Un agent est un système informatique, *situé* dans un environnement, et qui 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:

***Situé*:** L'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement. Exemples: systèmes de contrôle de processus, systèmes embarqués, etc.

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

***Flexible*:** l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans les temps requis.

***Proactif*:** l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au moment opportun.

***Social*:** l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ses agents à accomplir les leurs.

I-3- D'autres types d'agents

Les chercheurs en SMA proposent donc chacun leur définition de l'agent. On trouve néanmoins dans l'industrie logicielle, d'autres agents qui ne répondent pas aux définitions des chercheurs du domaine. Ces agents peuvent être réduits à de simples programmes (processus) qui accomplissent des tâches et adoptent une forme de mobilité. Il est possible de distinguer plusieurs types d'agents sur Internet : les agents pour le commerce et les achats, les agents assistants, les agents pour la recherche d'information.

Nous constatons, donc qu'il y'a plusieurs définitions de ce qu'est un agent, et que la communauté des SMA ne s'est pas mis d'accord sur une définition précise [38]. En effet, étant donnée la diversité des domaines d'applications des agents, ces définitions ont été généralement introduites dans le cadre d'un domaine d'applications bien spécifique tel que la robotique et les applications de réseaux. Néanmoins il est possible d'engager les principales fonctionnalités de l'agent afin de le définir. En effet, il existe un noyau commun à la plupart des définitions de la littérature.

L'agent agit en fonction d'un certain nombre d'informations reçu et perçu de son environnement et en fonction des buts qu'il poursuit. Son comportement donne l'impression que ses actions sont dirigées par raisonnement. Il est ainsi plus aisé de décrire un agent par ce qu'il fait (son comportement) que par ce qu'il est.

En résumé, un agent est capable de **percevoir** au moins partiellement, son environnement, d'**agir** sur son environnement; de se comporter de manière rationnelle. Ses actions sont le produit d'un raisonnement, ou d'un ensemble de règles de comportement dans le but d'obtenir un état souhaité de l'environnement.

Les fonctionnalités de l'agent sont donc de trois ordres qui répondent aux trois étapes générales de réalisation d'une tâche :

- Perception : l'agent perçoit son environnement et met à jour ses représentations et ses connaissances internes;

- Cognition : l'agent détermine ce qu'il doit faire et décide quant et comment le faire. IL existe plusieurs types de raisonnements; les plus utilisés sont le raisonnement réactif (réflexe) et le raisonnement cognitif ;

- Action: correspond à la réalisation effective des actions ayant été décidées par l'agent;

I-4- Typologie des agents

Les agents peuvent être classés en deux catégories principales selon leur comportement et leur cognition. Cette notion de cognition exprime la complexité de raisonnement d'un agent afin de distinguer les agents dits "intelligents" des agents moins "intelligents". On parle d'agents cognitifs et d'agents réactifs. Les agents cognitifs, peuvent être assimilés à des systèmes experts distribués. La connaissance des systèmes experts en question étant utilisée pour la représentation du monde que se fait l'agent modélisé, ainsi que pour exprimer ses intentions d'actions. Par contre, les agents réactifs, ne possèdent pas de vision globale du système mais répondent par des actions à des stimuli. C'est le cas par exemple de l'agent " fourmi " dans son environnement " fourmilière ". La communication entre agents réactifs s'effectue à l'aide de propagation de signaux ayant une signification intrinsèque mais dénuée de sémantique profonde.

I-4-1- Agents cognitifs

C'est un type d'agents complexe [21]. De tels agents sont non seulement capables de percevoir et d'agir sur leur environnement, mais en plus ils ont des capacités de cognition leur permettant de raisonner sur les autres agents ou sur l'avancement de la résolution. Ils font souvent appel à des modes de communication (coopération) plus complexes qu'une simple perception. Ils communiquent généralement grâce à des structures de données partagées ou par des communications directes. L'action d'un tel agent n'est plus seulement une réaction à un stimulus externe mais une réelle production de son comportement interne.

Un agent est cognitif s'il est "intelligent" : il dispose d'une base de connaissances comprenant l'ensemble des informations et des savoir-faire nécessaires à la réalisation de ses tâches et à la gestion des interactions avec les autres agents et avec son environnement. Il dispose d'une représentation symbolique et explicite du monde à partir de laquelle il peut raisonner. Un agent cognitif peut être intentionnel s'il possède des plans explicites lui permettant d'accomplir ses buts. Un agent est dit rationnel [49] s'il utilise au mieux ses connaissances et ses capacités de réflexion. Il choisit d'agir au mieux pour satisfaire un but.

Par extension, un agent est dit irrationnel, lorsque son développement arrive au point de manifester des états mentaux. L'agent peut alors avoir des croyances et exprimer un doute ou au contraire une certaine conviction. Il répond alors davantage à des impulsions internes plutôt qu'à une analyse rationnelle de la situation. L'agent cognitif devra apprendre de ses observations, de ses réussites et de ses échecs. Il modifiera en conséquence sa perception du monde.

I-4-2- Agents réactifs

C'est un type d'agents qui ne peut pas vraiment être qualifié d'intelligent, étant donné que son fonctionnement est principalement basé sur le principe du stimulus/action. Cependant il réagit uniquement à sa perception de l'environnement et agit en fonction de cette perception. A la perception d'un stimulus particulier, l'agent fournit une réponse automatique. Néanmoins, l'agent réactif présente des caractéristiques intéressantes [15] : simplicité de la description du comportement local et émergence de comportements globaux. Les principaux travaux utilisant des agents réactifs concernent les robots [05], [53], la planification [32] ou la résolution collective de problèmes [14].

D'après Ferber, un agent réactif ne répond qu'à des stimulus de l'environnement, son comportement étant guidé intégralement par l'état local du monde dans lequel il se trouve plongé. Il agit de manière totalement réflexe aux états du monde, il n'a aucun but ni aucun état interne [26]

La différence entre l'agent cognitif et réactif se situe d'une part dans la complexité des représentations au sein des agents, mais aussi dans la complexité des informations échangées. En utilisant leurs capacités cognitives, les agents cognitifs échangent des données plus riches que les agents réactifs. En effet, ils sont souvent pourvus de connaissances et de savoir-faire leur permettant des raisonnements plus élaborés (inférences, filtrage des informations, accointances, ...). La gestion de leurs interactions s'en trouve donc différente [09]. Bien souvent, les agents cognitifs disposent d'une représentation explicite des autres agents et de la manière d'interagir avec eux. Cette représentation est désignée par le terme de réseau d'accointance.

I-4-3- Agents hybrides

Certains chercheurs de la communauté SMA définissent un autre type d'agents communément appelés agent hybride. Il reprend les caractéristiques des agents réactifs et des agents cognitifs pour former une entité capable de se comporter encore différemment [07], [29]. Cette forme d'agent, très peu courante, est représentée en modélisation par objets par un héritage multiple qui n'est plus en vigueur dans les langages récents. Elle fut utilisée il y a quelques années, dans la conception des premiers SMA. Nous évoquons uniquement son existence dans un souci d'exhaustivité, mais nous n'en parlerons donc plus dans les prochains chapitres.

I-4-4- Agents Situés et Agents Communicants

Une autre classification courante des types d'agents est celle fondée sur leur relation à l'espace (agents situés) et leur relation aux autres agents (agents communicants). Comme nous l'avons vu précédemment, un des éléments clé des SMA est l'environnement. Cet environnement est souvent un espace métrique et la donnée des références spéciales d'un agent est pertinente pour la modélisation d'un grand nombre de phénomènes : planification de la trajectoire en robotique, détection d'indices visuels complexes en analyse d'images, détection de contours en vision artificielle. Ces problèmes seront alors dit spécialisés [19].

Les agents sont également caractérisés par leurs liens avec les autres agents. Lorsqu'un agent possède des compétences ou des services qu'il peut offrir aux autres agents, ou lorsqu'un agent doit solliciter pour satisfaire un but, des compétences ou des services d'autres agents, il s'avère nécessaire d'établir entre eux des communications. La communication est à la base des interactions directes entre les agents dans un système multi-agents. Un agent communicant sera donc un agent disposant des aptitudes à échanger des messages, c'est-à-dire envoyer et recevoir des messages des autres agents. Un agent n'a généralement qu'une perception partielle du système. Il n'a pas donc la possibilité de communiquer avec tous les autres agents du système. Dans un système multi-agents communicant, chaque agent dispose d'un ensemble d'accroissances, (terme désignant l'ensemble des autres agents du système avec lesquels il est susceptible de communiquer). De plus, communiquer implique la mise en oeuvre de langages et de protocoles appropriés.

I-5- Le Rôle

Un agent tient un rôle parce qu'il est présent dans une société ou dans une organisation. Un rôle peut être attribué dynamiquement à un agent. Il peut par exemple jouer un rôle de client ou de fournisseur ou même les deux simultanément. D'autres rôles peuvent être définis comme le rôle de médiateur qui se charge de la mise en relation des clients et des fournisseurs.

Dans les premières architectures décentralisées, un agent présentait directement ses fonctionnalités comme un objet possédant des méthodes. Pour des raisons de découplage fonctionnel propre au génie logiciel et des raisons conceptuelles liées à la notion de sociétés d'agents, on préfère aujourd'hui présenter les fonctions d'un agent via une interface appelée le rôle.

Chaque méthodologie de conception des SMA peut appréhender le rôle de différentes façons. Certaines proposent d'associer potentiellement plusieurs rôles à un agent. D'autres spécifient qu'un rôle est au contraire tenu par plusieurs agents. Dans une méthodologie comme dans Aalaadin [27], chaque agent peut avoir plusieurs rôles et un rôle identique peut être tenu par plusieurs agents, mais le rôle sera toujours local à l'agent. C'est à dire que si un rôle nécessite lui même une organisation d'agent, on associera tout de même le rôle à un agent et l'agent pourra acquérir une compétence qui pourrait elle même être composée d'agents. Dans [15], un agent est vu comme un ensemble de rôles, parmi lesquels on peut distinguer trois niveaux :

- Les rôles individuels qui sont les différents comportements que les agents sont capables de tenir sans se soucier de la stratégie choisie pour les tenir ;

- Les rôles relationnels qui concernent comment ils choisissent d'interagir avec un autre agent, tout en respectant les dépendances mutuelles de leurs rôles individuels ;

- Les rôles organisationnels ou comment les agents peuvent gérer leurs interactions pour devenir ou rester organisés.

I-6- Le Comportement

Le comportement caractérise l'ensemble des propriétés apparentes que l'agent manifeste dans son environnement. Un comportement est une réponse à un évènement ou une situation. Dans le contexte des agents, un évènement est une chose qui se produit et qui change l'état de l'environnement ou de l'agent. Un agent définit son comportement en fonction des évènements qui lui arrivent. D'une manière générale, il s'agit de l'introduction d'un nouvel élément dans un système. Lorsqu'un évènement se produit, l'agent doit l'analyser et l'évaluer pour produire une réponse adaptée.

La décision d'un agent va donner lieu à une action. Ce processus constitue le comportement de l'agent. Une action va modifier l'état du monde. Par extension, l'action n'est pas seulement envisagée comme le résultat de ce que font les agents, mais comme le résultat des réactions du monde aux influences des agents.

Le comportement d'un agent peut être considéré d'un point de vue externe ou interne à l'agent. Le comportement externe d'un agent correspond à l'observation d'une suite d'actions entreprises par celui-ci, alors que son comportement interne correspond à l'expression de ses capacités de perception, de décision, et d'action.

I-7- L'environnement

Un agent est plongé dans un environnement. Cela peut être l'environnement géographique, social, informatique. L'environnement est une structure dans laquelle l'agent évolue. Un agent va agir sur son environnement et l'environnement va agir sur l'agent. Cette structure peut être centralisée (une seule entité) et peut aussi être formée de cellules réunies en réseau et on la représente sous forme d'un automate cellulaire. On parle dans ce dernier cas d'environnement distribué.

Dans un environnement centralisé, les agents ont accès à la même structure. Ils produisent des influences sur l'environnement et celui-ci répond en renvoyant les éléments consommés ou perçus. Les problèmes de poursuite comme les proies/prédateurs sont un exemple d'environnement centralisé. Les systèmes comme Mace [33], Mice [13] Cormas [01] utilisent de tels environnements.

Ce qui distingue un environnement distribué d'un environnement centralisé tient en quatre points :

- L'état de la cellule dépend des autres cellules, des influences produites par les agents dans les cellules voisines ;

- La perception des agents s'étend généralement au-delà d'une cellule,
- Les agents se déplacent de cellule en cellule, ce qui fait qu'il faut gérer les liens que les agents entretiennent avec une cellule particulière,
- Des signaux peuvent se propager de cellule en cellule. Cette propagation prend un certain temps et il faut alors synchroniser le déplacement des signaux avec les mouvements des agents.

Selon Z. Guessoum, la relation entre un agent et son environnement peut être plus ou moins forte selon la façon dont l'agent acquiert l'information en provenance de l'environnement [36]. À partir des informations obtenues, l'agent peut créer et modifier sa propre représentation de l'environnement. La connexion entre l'agent et l'environnement est forte dans le cas d'un agent situé, c'est-à-dire lorsque l'agent possède des capteurs et des effecteurs. Les capteurs sont capables de percevoir directement les éléments de l'environnement ou les modifications survenues dans cet environnement. Les effecteurs permettent à l'agent d'agir sur son environnement. L'environnement d'un agent situé est habituellement un espace à deux ou trois dimensions constitué de cellules géométriques pouvant contenir des éléments perceptibles et modifiables par les agents.

L'environnement est en général très spécifique à l'application dans laquelle les différents composants agents sont immergés. Le problème majeur des SMA est sûrement la représentation du monde. Il faut avant toute chose, décrire de façon précise le monde dans lequel les agents vont évoluer.

I-8- Les Architectures des agents

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-agents étant de plus en plus nombreuses, on voit émerger dans la communauté des SMA, des architectures d'agents plus ou moins dédiées à des systèmes particuliers. Les architectures les plus courantes sont :

I-8-1- Les Architectures Réactives

La modélisation par agents réactifs s'oppose à l'approche des agents qui raisonnent en manipulant des informations symboliques telles que rencontrées dans les systèmes de planification ou résolution de problèmes. Dans les systèmes d'agents réactifs, l'accent est mis sur la réalisation de tâches élémentaires par chaque agent, les interactions entre eux devant conduire à l'émergence d'un comportement global intelligent. Plusieurs architectures ont été proposées pour mettre en oeuvre les agents réactifs [25]. Un agent réactif peut être décrit par un ensemble fini de règles simples qui lient les perceptions de l'agent à ses actions. Ces règles sont de la forme :

If <percieved situation > Then <specificaction>

I-8-2- Les Architectures logiques :

Elles correspondent à l'approche traditionnelle en Intelligence Artificielle. A partir d'une représentation symbolique des faits et des règles de raisonnement, un système est capable par des manipulations syntaxiques de ces représentations de produire un comportement ''dit intelligent''.

Cette architecture peut être utilisée pour définir les agents dans les problèmes de planification. Un exemple connu est celui du monde des cubes [26]. Le problème consiste à empiler des cubes les uns sur les autres de manière à obtenir un emplacement précis à partir d'un emplacement initial en ne bougeant qu'un seul cube à la fois.

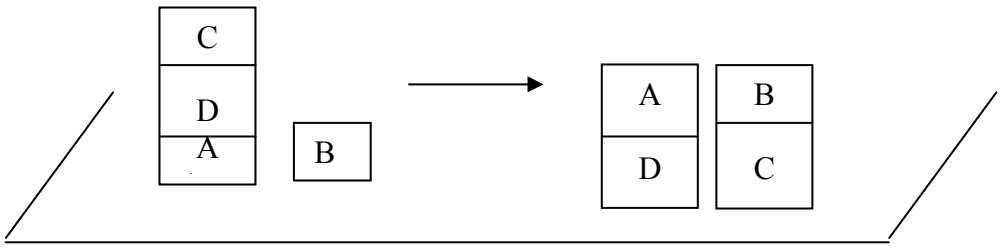


Fig 1: Exemple de manipulation de cubes

Ces situations sont définies simplement sous la forme d'un ensemble de formules atomiques :

Sinit ={sur (C,D),sur(D,A),sur(A ,Table),sur(B, Table) }
Sfin={ sur (A,D),sur(B,C),sur(D ,Table),sur(C, Table) }

Les opérateurs correspondants aux deux actions classiques d'empilement d'un cube sur l'autre et de dépôt du cube sur la table peuvent s'écrire ainsi :

Opérateur poser(x, y)
Pré : libre(x), libre(y), sur (x, z)
Suppr : libre(y),sur(x, z)
Ajouts : libre(z),sur (x, y)
Fin

Opérateur poserTable(x)
Pré : libre(x), sur(x, y)
Suppr : sur(x, y)
Ajouts : libre(y), sur(x, Table)
Fin

I-8-3- L'architecture de subsomption :

L'architecture de subsomption est très proche des architectures réactives, et certains auteurs la citent comme telle. Introduite par Brooks [04], cette architecture suggère de décomposer un agent en modules verticaux, chacun d'eux n'étant responsable que d'un type de comportement très limité. Ces modules effectuent leurs tâches en parallèle et il existe une priorité dans les modules, telle que lorsque deux modules fournissent des résultats contradictoires, seuls ceux du module dominant sont pris en compte. La figure 2 présente un exemple caractéristique d'architecture de

subsumption pour un rebot explorateur.

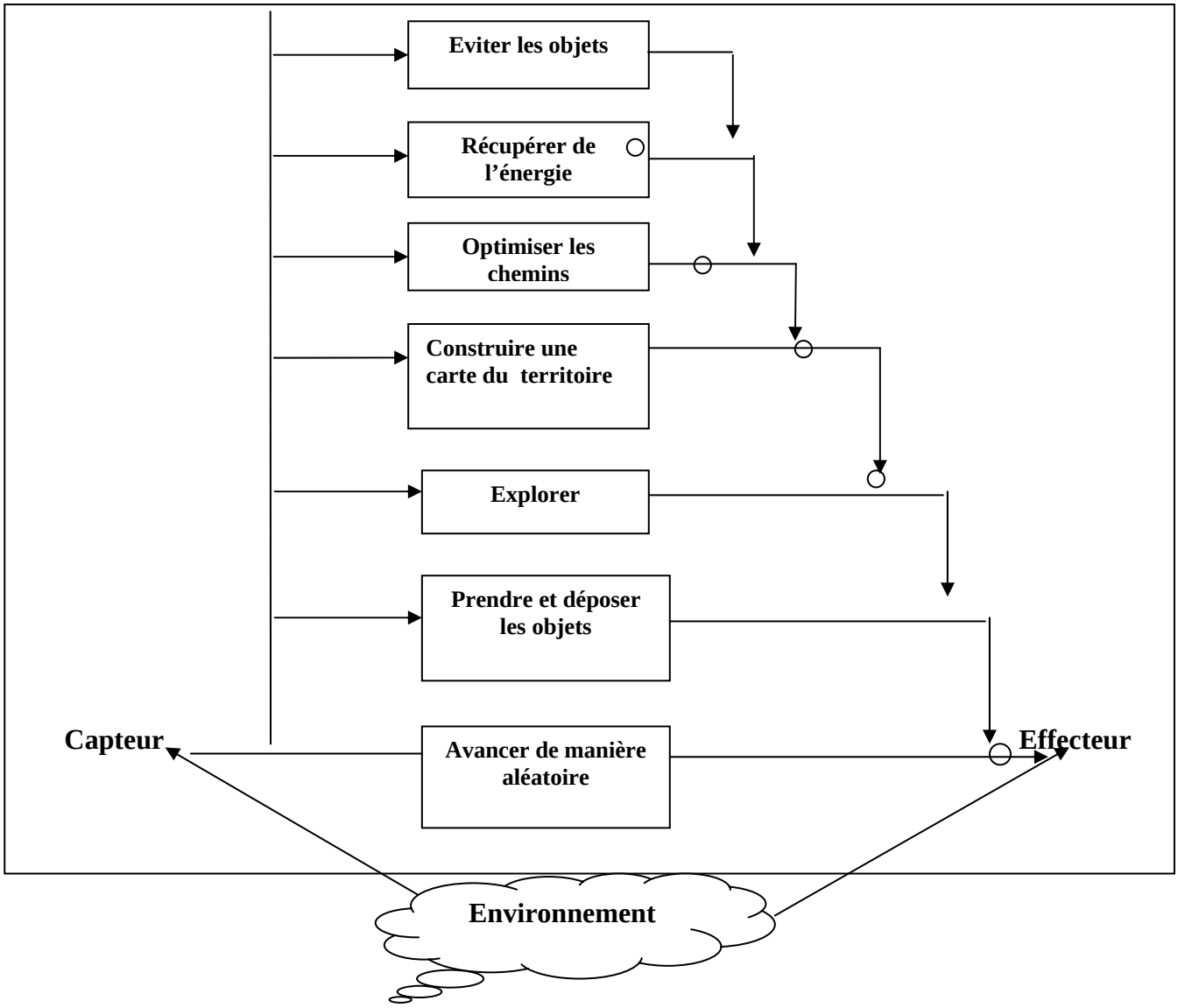


Fig 2: architecture de subsomption

I-8-4- Les Architectures BDI (Belief, Desire, Intention)

Dans le cadre du raisonnement pratique, un raisonnement vers la prise en compte des états mentaux, les chercheurs ont développé l'architecture BDI (*Belief, Desire, Intention* pour croyance, désir, et intention), [34], [52] une architecture bâtie autour du raisonnement pratique. Ces agents sont représentés par un état mental ayant les attitudes mentales suivantes :

- *Les croyances* : Ce que l'agent connaît de son environnement;
- *Les désirs* : Les états possibles envers lesquels l'agent peut vouloir s'engager;
- *Les intentions* : Les états envers lesquels l'agent s'est engagé, et envers lesquels il a des ressources.

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 proposer ses actions au vu de ses intentions.

I-8-5- les Architectures hybrides

Les systèmes réactifs pouvaient bien convenir pour certains types de problèmes et moins bien pour d'autres. De même, pour la plupart des problèmes, les solutions de l'IA classique, basée uniquement sur la planification, ne conviennent pas aussi. Dès lors, les chercheurs ont commencé à investiguer la possibilité de combiner les deux approches afin d'obtenir une architecture hybride [08]. Dans ce cas, un agent est composé de plusieurs couches, arrangées selon une hiérarchie. La plupart des architectures considèrent que trois couches suffisent amplement. 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 senseurs. 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, c'est à dire du raisonnement tenant compte des autres agents.

I-7- L'agent comme entité d'un système

Les systèmes multi-agents permettent de faire face aux limites des capacités des agents isolés. Suite à la naissance des systèmes multi-agents, différents modèles de comportement ont été proposés pour les agents, indiquant comment l'agent doit générer sa décision en fonction des buts spécifiés et de ses percepts.

Les systèmes multi-agents permettent une meilleure distribution et structuration des connaissances et du contrôle, par rapport à un système mono-agent, et de simplifier ainsi le comportement d'un agent. En effet, dans un SMA, un ensemble d'agents interagissent selon des modes plus ou moins prédéfinis, afin de répondre au problème pour lequel le SMA a été conçu. Selon Ferber, le développement des SMA est du à quatre raisons [28], que nous détaillons dans ce qui suit.

La première raison est que l'intelligence artificielle classique, qui consiste à centraliser les connaissances et le contrôle en une seule entité, ne nous permet malheureusement pas de modéliser des problèmes complexes. Ceux la sont à l'origine de l'apparition de l'intelligence artificielle distribuée (IAD), qui distribue les connaissances et le contrôle sur un ensemble d'entités. En effet, il y a eu tout d'abord les systèmes multi-experts faisant intervenir plusieurs bases de connaissances plus ou moins coordonnées. Mais, un besoin en coopération entre les systèmes s'est fait sentir afin qu'ils réussissent bien leur(s) mission(s), et qui ne pouvait pas être facilement exprimé à l'aide du formalisme des systèmes experts.

La seconde raison provient du besoin en nouvelles techniques performantes pour la modélisation et la simulation dans le domaine des sciences. En effet, il s'est avéré qu'un "simple" ensemble d'équations différentielles ne permet pas toujours de reproduire l'évolution des écosystèmes habités, et une modélisation représentant les individus par des entités informatiques paraissait intéressante.

La robotique est la troisième raison d'apparition des SMA. Des expériences ont montré qu'un ensemble de petits robots autonomes, tant au niveau énergie que décisionnel, dotés de comportements plus ou moins primitifs sont capables de produire un comportement collectif performant, qui peut être qualifié d'"intelligent" [26]. Ce comportement, connu sous le nom de comportement émergent, est aussi performant ou meilleur que celui d'un seul robot complexe doté de toutes les capacités pour le rendre "intelligent". Mais, le problème est de faire coopérer ces robots pour arriver au comportement collectif "intelligent".

Enfin, l'évolution de l'informatique et plus particulièrement celle des systèmes distribués forment la quatrième raison. La collaboration entre des composants logiciels dans des environnements hétérogènes et distribués pour la réalisation d'un objectif commun, dépasse les fonctionnalités standard d'un programme et nécessite des capacités de coopération avec les autres programmes pour atteindre l'objectif commun.

I-7-1- Définition d'un SMA

Un agent est une entité relativement autonome présente dans un système qui lui permet d'évoluer et de communiquer. Cependant il est amené à interagir avec d'autres agents et éventuellement avec son environnement. C'est l'ensemble de ces éléments (agents, environnement et interactions) que l'on appellera système multi-agents. Plus précisément, on appelle système multi-agents [26] ou SMA, un système composé des éléments suivants:

- Un environnement E, c'est à dire un espace disposant généralement d'une métrique,

- Un ensemble d'objets O situés dans l'espace. Ils sont passifs, ils peuvent être perçus, détruits, créés et modifiés par les agents,

- Un ensemble d'agents A qui sont les entités actives du système. Un ensemble de relations R qui unissent les objets entre eux,
- Un ensemble d'opérations OP permettant aux agents de percevoir, de détruire, de créer, de transformer et de manipuler les objets,
- Un ensemble d'opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification (les lois de l'univers).

Les relations (R) qui unissent les objets entre eux ainsi que les opérateurs (OP) regroupent deux notions clés de la théorie des SMA : les interactions et les communications.

Donc un système multi-agents est un ensemble d'entités (les agents) autonomes actives. Les phénomènes ou les comportements sont distribués au niveau individuel des agents. Chaque agent est alors spécialisé et agit de façon autonome. De ces actions individuelles, émerge la solution ou le comportement général du système, soit par une interaction due à la modification par les agents de l'environnement où ils évoluent, soit par une communication directe entre agents par le biais d'un langage symbolique.

Dans un système de gestion du trafic aérien par exemple, les agents sont les avions. L'environnement étant l'espace aérien dans lequel la position de chaque avion est définie par ses coordonnées. Ces agents suivent des trajectoires et doivent mettre en oeuvre des stratégies pour éviter les collisions. Ils peuvent ainsi communiquer directement entre eux ou à travers une tour de contrôle.

Un SMA peut être homogène, c'est-à-dire que tous les agents sont construits sur le même modèle (ex: colonie de fourmis) ou hétérogène, c'est-à-dire que les agents sont de modèles différents (ex: un éco-système).Un agent a une représentation partielle de son environnement, il peut utiliser des objets de son environnement et communiquer avec d'autres agents. Lorsqu'il se fixe le but de réaliser une action, il a besoin de compétences et de ressources. L'intérêt du SMA réside dans la faculté innée des agents le constituant, d'élaborer des interactions communes offrant la possibilité d'allouer des tâches de façon dynamique et de coordonner des actions.

D'une manière générale, un agent, dans la théorie des SMA, est doté de compétences, de possibilités de communication ou d'interaction avec les autres agents et/ou avec son environnement, de croyances sur certains autres agents du système, d'aptitudes qui lui permettent de raisonner et d'une attitude sociale basée sur la coopération. Le comportement de chaque agent est spécifié de manière à ce qu'il essaie d'atteindre son objectif local tout en gardant des interactions coopératives avec les autres.

Avant chaque action, l'agent analyse localement s'il est ou non en interactions coopératives. Plus précisément, il recherche s'il existe des situations non coopératives et, si tel est le cas, tente de les éliminer pour revenir à un état coopératif. La coopération est l'attitude sociale qui guide le comportement de chaque agent par la prise en compte de critères locaux. Un agent a donc deux rôles essentiels : le premier est de réaliser la fonction pour laquelle il a été spécifié et le second est d'agir sur l'organisation interne du système s'il rencontre des situations non coopératives dans le but de retourner à des situations coopératives.

I-7-2- Les Interactions

Les interactions sont un élément indispensable à la constitution d'organisations sociales [49]. C'est grâce à la coopération que les agents peuvent accomplir des actions et atteindre leurs objectifs.

L'interaction est donc le composant essentiel de toute organisation. Un agent sans interaction avec d'autres agents n'est plus qu'un corps isolé dépourvu de toute caractéristique adaptative. En plus, c'est aussi en raison de la multitude des agents dans une organisation, ces derniers doivent coordonner leurs actions et résoudre des conflits. Résoudre des conflits implique la mise au point d'un protocole de négociation et donc implique un certain nombre de communications. C'est bien là que se situe le principal point faible des SMA : faire face à une quantité d'informations considérable. Traiter des interactions, cela consiste à décrire des mécanismes élémentaires permettant aux agents d'interagir, mais également d'analyser et de concevoir les différentes formes d'interactions que les agents peuvent avoir pour accomplir leurs buts[23].

La notion d'interaction suppose :

- Un ensemble d'agents capables d'agir et de communiquer.
- Des situations susceptibles de servir de point de rencontre entre agents : collaboration, utilisation de ressources limitées, régulation de la cohésion des agents.
- Des éléments dynamiques permettant des relations locales et temporaires entre agents.

Les différentes formes d'interactions sont la collaboration et la coordination d'actions. La première consiste à répartir le travail entre plusieurs agents. La seconde s'intéresse à la manière dont les actions des agents sont organisées dans le temps et dans l'espace pour accomplir leurs buts. D'une manière générale, les interactions entre individus jouent un rôle essentiel dans le développement d'un être vivant, que ce soit au

niveau réactif ou cognitif.

Ferber définit une situation d'interaction comme " un ensemble de comportements résultant du regroupement d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles " [26].

Dans un SMA, les principales situations d'interaction peuvent être classées par rapport à trois critères : les objectifs (buts) ou intentions des agents, les relations que les agents entretiennent envers les ressources qu'ils possèdent ainsi que les moyens (compétences) dont ils disposent pour parvenir à leurs fins.

Des agents seront dans une situation de collaboration ou d'indifférence si leurs buts sont compatibles. Si leurs buts ne sont pas compatibles, ils seront dans une situation d'antagonisme.

Les ressources dont disposent les agents sont une composante très importante dans une situation d'interaction. Une ressource est un élément environnemental ou matériel utile à la réalisation d'une action. Par conséquent, tout agent requiert des ressources pour accomplir ses tâches. La quantité de ressources limitées est souvent une source de conflit entre agents.

Si un agent a besoin de ressources pour accomplir une tâche, il a également besoin d'une ou de plusieurs compétences. S'il dispose de toutes les compétences requises pour accomplir sa tâche, il peut l'exécuter seul. Néanmoins, il se peut que l'agent ne possède pas toutes les compétences nécessaires, par conséquent, il est contraint de travailler en collaboration avec un autre agent ayant les compétences manquantes. Il peut ainsi sous-traiter une tâche à un autre agent. Les interactions sont donc dans ce cas très bénéfiques, puisque les actions de chacun des agents contribuent à satisfaire un objectif commun. Le système résultant de ces interactions dispose alors de propriétés nouvelles qui s'expriment parfois comme une fonctionnalité émergente.

La coopération est une forme d'interaction. Elle consiste à établir qui fait quoi, avec quels moyens, de quelle manière et avec qui. Cela sous-entend qu'il faut trouver des solutions aux différents sous-problèmes que constitue la collaboration par répartition de tâches, la coordination d'actions et la résolution de conflits. La coopération se résume donc par la formule :

Coopération = collaboration + coordination d'actions + résolution de conflits

Dans le tableau ci dessous, plusieurs agents coopèrent ou sont dans une situation de coopération si les conditions relatives aux buts, ressources et compétences sont vérifiées. Cependant, plusieurs mécanismes (allocation des tâches, coordination d'actions, négociation et arbitrage) de résolution des différentes situations d'interactions ont été mis en œuvre par les chercheurs en SMA. Ces mécanismes sont expliqués plus en détail dans le paragraphe suivant.

But	Ressources	Compétences	Type de Situation	Catégorie	Solution
Commun	Suffisantes	Suffisantes	Indépendance	indifférence	/
Commun	Suffisantes	Insuffisantes	Collaboration Simple	Coopération	Allocations des tâches
Commun	Insuffisantes	Suffisantes	Encombrement		Coordination d'actions
Commun	Insuffisantes	Insuffisantes	Collaboration Cordonnée		Allocations des tâches et Coordination
Incompatible	Suffisantes _	Suffisantes	Compétition individuelle pure	Antagoniste	/
Incompatible	Suffisantes	Insuffisantes	Compétition collectives pures		/
Incompatible	Insuffisantes	Suffisantes	Conflits individuels pour des ressources		Arbitrage et Négociation
Incompatible	Insuffisantes	Insuffisantes	Conflits collectifs pour des ressources		Arbitrage et négociation

Tableau 1 : Les types de situation d'interaction

II-7-2- L'allocation des tâches

Le processus d'allocation des tâches permet de répartir les tâches, les informations ou les ressources entre les agents en fonction des buts et des compétences des agents et des contraintes contextuelles (types et quantités de ressources, nature de l'environnement etc).

-.Rôles d'allocation des tâches

Un système d'allocation automatique des tâches doit être capable de mettre en rapport des agents qui ont besoin d'une information ou de la réalisation d'une tâche. Le système est composé d'agents clients ou demandeurs, et des agents capables de fournir un service, les fournisseurs ou serveurs. Souvent les mêmes agents peuvent être à la fois clients et serveurs, la qualité de client ou de fournisseur étant déterminée généralement de manière dynamique lors du fonctionnement d'un système multi-agent. On dit alors que les agents prennent le rôle de client ou le rôle de fournisseur. D'autres rôles peuvent être définis, tels que celui de médiateur (trader), qui s'occupe de la mise en contact des serveurs et des clients [26].

- Les Modes d'allocation de tâches

La gestion de la répartition des tâches peut s'effectuer soit en centralisant le processus d'allocation soit en le distribuant à l'ensemble des agents concernés.

Dans un mode d'allocation de tâches centralisé, la répartition passe par la définition d'agents spéciaux, les médiateurs, qui gèrent l'ensemble du processus d'allocation en centralisant les demandes des clients et les offres de services des serveurs, afin de mettre en correspondance ces deux catégories d'agents. Par contre dans un mode d'allocation distribué, chaque agent s'occupe individuellement d'obtenir les services des fournisseurs qui peuvent lui être utiles pour la réalisation de ses projets. On distingue deux mécanismes d'allocation distribuée :

- Le mode ***d'allocation par réseau d'accointances :*** Dans ce cas les agents possèdent une représentation des autres agents et des capacités dont ils disposent. Il n'est pas nécessaire que les agents connaissent les capacités de tous les autres agents pour pouvoir résoudre leur problème, mais seulement que le réseau formé par l'ensemble des accointances soit fortement connexe (c'est-à-dire il existe un chemin allant de n'importe quel agent vers un autre agent) et que la cohérence est assurée (c'est-à-dire que les représentations sur les compétences des accointances d'un agent correspondent bien aux compétences effectives des agents).

- Le mode ***d'allocation par appel d'offre :*** C'est le mode le plus connu en intelligence artificielle distribuée et sous le nom de réseau *contractuel.* Il présente l'avantage d'une très grande dynamicité et s'avère particulièrement facile à mettre en oeuvre.

Ces modes s'appliquent à des systèmes d'agents cognitifs. Il existe une autre forme d'allocation moins connue généralement et plus caractéristique des systèmes réactifs dans lesquels les communications s'effectuent par des propagations de stimuli.

I-7-3- La coordination d'actions

Des agents qui travaillent dans un SMA accomplissent un certain nombre de tâches. Il existe également des tâches supplémentaires liées à la mise en relation des autres tâches pour améliorer globalement le déroulement. Il s'agit d'une articulation des actions individuelles de chacun des agents de manière à ce que l'ensemble du système se comporte de façon cohérente et performante. Ces tâches supplémentaires sont appelées des tâches de coordination. Dans [44] la coordination d'actions est décrite comme " l'ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement multi-agents et qu'un seul agent poursuivant les mêmes buts n'accomplirait pas ".

La coordination des actions est donc l'une des principales méthodes pour assurer la coopération entre agents autonomes. Il y a plusieurs raisons pour utiliser une coordination : dans un SMA, les ressources sont limitées et il faut bien souvent éliminer les actions inutiles qui consomment des ressources. En outre, les agents ont besoin d'informations et de résultats que seuls d'autres agents peuvent fournir. La

coordination d'actions est nécessaire pour quatre raisons :

- les agents ont besoin d'informations et de résultats que seuls d'autres agents peuvent fournir.

- les ressources sont limitées. On ne fait parfois attention aux actions des autres que parce que les ressources dont on dispose sont réduites et que l'on se retrouve à plusieurs à les utiliser. Qu'il s'agisse de temps, d'espace, d'énergie, d'argent ou d'outils. Les actions pour être accomplies ont besoin de ressources qui n'existent pas en quantités infinies, il faut donc partager les ressources de manière à optimiser les actions à effectuer (éliminer les actions inutiles, améliorer le temps de réponse, diminuer les coûts) tout en essayant d'éviter les conflits éventuels (conflit d'accès, collision, actions contradictoire).

- Optimiser les coûts de coordination des actions permet aussi de diminuer les coûts en éliminant les actions inutiles et en évitant les redondances d'actions.

- On veut permettre à des agents ayant des objectifs distincts mais dépendant les uns des autres de satisfaire leurs objectifs et d'accomplir leur travail en tirant éventuellement parti de cette dépendance.

On note plusieurs formes de coordination d'actions :

1. La coordination par ***synchronisation :*** Il s'agit de la forme la plus élémentaire de la coordination. Toute coordination doit décrire l'enchaînement des actions, ce qui introduit une nécessaire synchronisation de leur exécution. On parle généralement de synchronisation lorsqu'il s'agit de gérer la simultanéité de plusieurs actions et de vérifier que le résultat est cohérent. Très utilisée en automatisme, cette forme de coordination se formalise souvent par des réseaux de Petri.

2. La coordination par ***planification :*** Cette technique repose sur un découpage .de l'action en deux phases: dans la première, on réfléchit sur l'ensemble des actions à effectuer pour atteindre un but en produisant un ensemble de plans. Du fait que l'environnement peut évoluer, les plans peuvent être amenés à être révisés au cours de leur exécution, de ce fait les agents sont contraints de revoir leur planification en temps réel, ce qui est très difficile à implémenter. De plus dans les systèmes multi-agents, les différents plans élaborés par les agents peuvent entraîner des conflits d'objectifs ou d'accès à des ressources. Il faut alors coordonner des plans de manière à résoudre les conflits et satisfaire ainsi les buts des différents agents.

3. La coordination ***réactive*** **:** C'est en quelque sorte le contraire de la planification. Des agents réactifs gèrent leur boucle de perception-action en temps réel. En fonction d'un stimulus de l'environnement, l'agent déclenche spontanément une action réflexe. Il n'y a donc pas du tout de planification. Bien souvent cette approche n'assure pas une optimisation du système sur le plan de l'efficacité mais assure une robustesse du système et une grande adaptabilité.

Ce type de coordination est celui qu'utilisent les bancs de poissons, les oiseaux migrateurs ou les meutes de loups. Des simulateurs comme ICHTYUS [48] ont montré la validité de cette approche.

4. La coordination par ***réglementation :*** Cela consiste à édicter des règles de comportement qui visent à éliminer les conflits potentiels entre agents. On pourrait appeler ces règles, des heuristiques. C'est une extension d'une coordination réactive avec des règles d'un niveau plus haut. Le meilleur exemple est le code de la route.

I-7-3- L'Arbitrage et la Négociation

Lorsqu'un conflit apparaît entre deux ou plusieurs agents, il y a deux méthodes pour le résoudre : l'arbitrage et la négociation. L'arbitrage consiste à définir des règles de comportement (les textes de lois) qui agissent comme des contraintes sur l'ensemble des agents. Le résultat global a pour effet de limiter les conflits et donc empêcher la déstabilisation du système. Dans ce cas les agents sont très cognitifs et sont capables d'un libre-arbitre. L'implémentation de ce genre d'agents n'est guère facile, Il est néanmoins possible de donner à ces agents cognitifs des règles de comportements et lorsqu'une règle n'a pas été édictée pour résoudre une situation de conflit, un agent arbitre décide d'une action à accomplir pour régler la situation.

Dans des cas très restreints, l'agent arbitre prend une décision, même si ce n'est pas la meilleure, et débloque la situation. Il peut donc s'agir d'un arbitre réactif qui choisit aléatoirement entre deux alternatives dans le seul but de débloquer le système. Une façon de régler un conflit est l'utilisation d'un protocole de négociation. Deux agents en conflit vont entrer en communication pour trouver un accord bilatéral. Cette forme de résolution a été particulièrement étudiée par les chercheurs, Durfee [16] et Sycara [55]. Cette négociation ne fonctionne évidemment qu'avec des agents cognitifs. Dans le cadre des agents purement réactifs, les conflits sont souvent rendus impossibles par la description précise que l'on peut faire de l'environnement et par des règles de fonctionnement intégrées aux agents.

I-8-Avantages et Inconvénients des SMA

Les systèmes multi-agents sont de plus en plus utilisés dans les sciences de la vie et de l'environnement pour leur possibilité de représenter directement les différents acteurs biologiques, sociaux et économiques, leurs comportements et leurs interactions.

D'un point de vue technique, les systèmes multi-agents présentent de nombreux avantages, tels que :

- Tolérance aux failles : Comme un SMA, est un système distribué, constitué d'agents autonomes, l'endommagement d'une où plusieurs entités ne conduit guère à l'arrêt du système entier.
- Logiciel modulaire et extensible : Dans un SMA, le problème est décomposé en plusieurs sous problèmes partagés entre les agents. Un agent peut se joindre au système à tout moment et proposer ses services, ou bien le quitter dès qu'il a fini le plan qu'il s'est engagé à prendre en charge.
- Résolution rapide des problèmes : Résolution parallèle du problème par plusieurs agents.
- Systèmes flexibles : Plusieurs agents dotés de différents comportements qui travaillent en équipe pour résoudre le problème.

Ces avantages permettent de résoudre les problèmes du contrôle séquentiel et des architectures centralisées [16].

Bien que les systèmes multi-agents présentent des avantages considérables, ils soulèvent plusieurs inconvénients, tels que :

- La définition d'un modèle, c'est-à-dire un modèle qui assure au modélisateur que l'outil informatique implémente bien le modèle décrit et que les conséquences observées lors de la simulation sont bien directement des conséquences du modèle et non le produit d'un comportement non désiré du système informatique.

- La simultanéité des actions : les simulateurs existants ne savent pas bien prendre en compte des actions simultanées. Ainsi, deux agents qui, dans le modèle, devraient agir en même temps, vont être animés l'un après l'autre dans la simulation. Le problème n'est pas seulement technique, il est aussi théorique : il a nécessité le développement d'une théorie de l'action, le modèle "influence/réaction", une extension du calcul situationnel, qui permet de décrire des comportements simultanés d'actions.

- L'explosion du nombre de communications entre agents pour l'organisation du SMA. Les agents sont des entités autonomes plongées dans un environnement et très communicantes. Si les SMA peuvent apporter des solutions à des problèmes de nature intrinsèquement distribuée [50], ils apportent avec leur solution une autre problématique qui est l'augmentation non négligeable des communications. Il arrive fréquemment qu'une application SMA soit moins rapide à l'exécution qu'une application non répartie. La simple

répartition du calcul est loin de garantir automatiquement des temps de simulation plus courts. Une conception inadaptée de la simulation pourra conduire à un nombre important de messages de partage d'informations entre agents simulés ou environnement transitant par le réseau, causant dans certains cas une exécution plus lente qu'une simulation non distribuée.

En plus des inconvénients cités ci-dessus, le recours à une modélisation des systèmes multi-agents s'avère nécessaire. En effet, le code des programmes permettant aux agents de s'exécuter n'est pas suffisant à la compréhension de leur fonctionnement. Pour un système multi-agent, comme d'ailleurs pour n'importe quel système informatique, analyser le comportement d'un programme à partir de son implémentation présente les inconvénients suivants *[26]* :

- Il est particulièrement difficile d'entrer dans le cœur d'un agent et d'analyser son comportement simplement en regardant son implémentation ;
- Le code d'un programme introduit un grand nombre de détails inutiles à la compréhension du fonctionnement de l'agent ;
- Des implémentations différentes, en termes de langages et de choix de structure de données, présentent des manifestations semblables. Alors que des modifications légères dans un programme peuvent avoir des conséquences importantes sur le comportement de ce dernier ;
- L'implémentation des agents s'effectuant en parallèle, il est encore plus difficile de comprendre leur fonctionnement à partir de leur code, du fait du non déterminisme inhérent au parallélisme ;
- Enfin, et surtout, les systèmes multi-agents sont des logiciels complexes, difficiles à appréhender et à concevoir. Il est donc nécessaire de réduire leur complexité en dégageant leur structure et en analysant séparément leurs différents composants sans perdre de vue leur organisation générale.

I-9- La modélisation SMA

I-9- 1- Les élément modélisables dans un SMA

On peut distinguer quatre parties principales pour lesquelles la modélisation s'avère particulièrement pertinente [26]:

- La modélisation des actions des agents et des conséquences de celles-ci dans l'environnement. Ce dernier peut être complexe et peut présenter aussi une évolution autonome, distincte des conséquences des actions des agents.
- Le fonctionnement d'un agent, aussi bien dans le comportement observable que dans la mécanique interne de l'agent.
- L'interaction des agents entre eux et en particulier les différents modes à savoir, la coopération, la coordination d'action et la résolution de conflits.
- L'évolution du système multi-agents, ce dernier étant considéré dans son ensemble, à partir d'expérimentations et de formalisations.

I-9-2- Les outils de modélisation des SMA

Les formalismes et les modèles les plus adaptés pour la modélisation et la compréhension des fonctionnements et des comportements des systèmes multi-agents sont les automates à états finis, les automates à registre et les réseaux de Petri [26].

I-9-2-1 Automates à états finis

Un automate à états finis est décrit par l'ensemble de ses états et par la fonction de transition qui spécifie l'état suivant en fonction de l'état actuel et des informations qu'il reçoit en entrée. On représente un automate à états finis sous la forme d'un graphe orienté dont les nœuds sont les états de l'automate et les arcs, ses transitions. Un SMA est décrit par un ensemble d'automates à états finis décrivant le comportement local de chacun de ses composants. Ces derniers sont définis par des équations algébriques et évoluent en exécutant des actions.

L'intérêt des automates à états finis est, d'une part, de pouvoir décrire très facilement le comportement d'un agent capable de mémoriser l'état dans lequel il se trouve et d'autre part, d'être soutenus à la fois par un support théorique important et d'avoir été utilisés dans de nombreux domaines de l'informatique.

I-9-2-2- Automates à registres

Les automates à registres sont décrits, comme les automates à états finis, par un ensemble d états et par la fonction de transition sur ces états. La différence majeure vient du fait que l'on ajoute aux états un ensemble de registres qui peuvent être manipulés lors du déclenchement des transitions. Ce qui permet d'appréhender des comportements plus élaborés qu'avec les automates à états finis.

L'intérêt des automates à registres est bien évidemment d'être beaucoup plus simples que leurs homologues à états finis, dès que le nombre d'états devient trop important. Grâce à leurs registres et aux conditions supplémentaires associées aux transitions, ils factorisent des informations qui doivent être réparties sur plusieurs états dans les automates à états finis.

Les inconvénients des automates

Les automates à états finis comme les automates à registres s'avèrent très limités pour représenter des comportements plus complexes, et encore moins pour décrire l'évolution d'un système multi-agent essentiellement pour deux raisons :

1- Leur capacité de calcul est limitée, du fait que leur nombre d'états est fini ;

2- Ils ne peuvent représenter que des processus séquentiels et sont donc bien limités pour prendre en compte les aspects du parallélisme, une des principales caractéristiques des systèmes multi-agents.

Les automates à états finis et les automates à registre peuvent être utilisés pour formaliser les structures d'agents qui apparaissent de manière isolés, par contre leur capacité de calcul est réduite du fait de la représentation explicite de tous les états. Ces modèles sont inadaptés pour prendre en compte le parallélisme, une des caractéristiques essentielles des systèmes multi agents. Il faut passer alors à des formalismes plus puissants et adaptés à la concurrence tels que les réseaux de Petri.

I-9-2-3- Les réseaux de Petri

Les réseaux de Petri sont un formalisme qui permettent de représenter de manière naturelle le parallélisme, la synchronisation, le partage de ressources et la lecture. Ils constituent un outil de modélisation puissant pour décrire et vérifier le comportement des systèmes distribués. Les RdP sont des outils de spécification et de description des processus fonctionnant en parallèle et avec des contraintes de synchronisation, pour lesquels, il existe un grand nombre de résultats théoriques, ainsi que des outils informatiques. Ils permettent de mettre en évidence l'absence de blocage lors de l'évolution d'un système et l'existence d'un régime permanent.

Les réseaux de Petri peuvent être considérés comme des outils de modélisation des systèmes multi-agents sous tous leurs aspects à savoir architecture, interaction et évolution. Ils sont fondés sur une représentation à la fois graphique et mathématique. L'aspect graphique permet de visualiser simplement les agents et de faciliter la conception et la communication de modèles de comportement, tandis que la formulation mathématique assure une analyse quantitative et qualitative. La formalisation des systèmes multi-agents par les réseaux de Petri , objet de ce mémoire sera traitée de manière très détaillée dans le chapitre 4

I-10- Conclusion

Ce chapitre nous a permis d'une part, de mettre le point sur les systèmes multi-agents en présentant les grands concepts qui définissent le domaine, à savoir les agents les typologies d'agents, les plates formes existantes, les interactions ainsi que architectures SMA. Et d'autre part, la puissance d'expression des modèles multi-agents qui permettent la représentation des entités autonomes en interactions, dotées de comportement et pouvant évoluer dans un environnement

Nous avons vu qu'un agent est une entité autonome ayant des facultés de perception, de communication, de décision et d'action. Un des aspects importants définissant le concept d'agent repose notamment sur son autonomie décisionnelle. Cette autonomie dépend du type d'agent et du modèle interne sur lequel repose le fonctionnement des agents, c'est à- dire du raisonnement et du comportement. Un agent est plongé dans un environnement et interagit avec celui-ci. Les interactions entre les agents et la résolution distribuée qu'elles mettent en place induisent alors une contradiction avec ce principe d'autonomie de décision propre au concept d'agent. Les agents peuvent avoir conscience de cette résolution collective (agents cognitifs) ou non (agents réactifs), mais de toute façon, ils utiliseront ou subiront à un moment ou à un autre des contraintes issues de leurs interactions avec les autres agents du système.

Toute la difficulté de concevoir des systèmes multi-agents consiste alors à gérer cette contradiction entre le principe d'autonomie des agents et la résolution collective par le système qu'ils composent. Tous les agents doivent s'intégrer au même système pour former un tout cohérent résolvant le problème à traiter. Cette nécessité d'intégration et d'interaction avec les autres fait intervenir des mécanismes et des notions permettant la coordination de la résolution distribuée du problème pour obtenir un comportement global cohérent et efficace du système. Les notions d'interactions et d'organisation permettent d'appréhender cette nécessité.

En plus, la vérification des systèmes modélisés par SMA repose souvent sur les simulations qui permettent de les évaluer par rapport à leurs objectifs. Cependant les simulations à elles seules ne permettent pas d'identifier les propriétés globales et générales des systèmes modélisés. Aussi, elles ne renseignent pas sur la non observabilité de certaines configurations non désirées du système, voir la non existence de situation de blocage, et les conclusions sur les évolutions des systèmes modélisés ne peuvent être que partielles.

Dès lors, l'utilisation des méthodes formelles de spécification pour les modèles multi-agents s'avère nécessaire, car ces dernières offrent des outils qui permettent d'analyser, de vérifier et de valider les systèmes modélisés par les systèmes multi-agents.

Ainsi le modèle réseaux de Petri s'apprête bien à la formalisation des systèmes multi-agents, car ces derniers sont un outil formel particulièrement privilégié pour modéliser des systèmes distribués, par leur capacité de représenter facilement le parallélisme et la synchronisation. En plus ils fournissent de nombreuses techniques d'analyse et de validation (l'examen du graphe des marquages accessibles,...).

Chapitre II

Les Réseaux de Petri

II-1- Introduction

Les réseaux de Petri (RdP) ont été introduits en 1962 par Carl Adam Petri. Ils constituent un outil très bien adapté à la modélisation des systèmes complexes. Le but de cette modélisation est de spécifier, d'analyser et de vérifier certaines propriétés qualitatives et quantitatives des systèmes modélisés.

Les réseaux de Petri permettent de spécifier, de décrire et modéliser des processus fonctionnant en parallèle et avec des contraintes de synchronisation. Ils permettent par exemple de mettre en évidence l'absence de blocage lors de l'évolution d'un système. Ils sont utilisés dans divers domaines tel que l'analyse des protocoles de communications, les pilotages des ateliers de fabrication, la conception de logiciels temps réels ou encore l'évaluation des performances des systèmes discrets.

L'analyse qualitative des modèles RdP permet de détecter les erreurs de conception importantes pour le fonctionnement du système, de réduire les risques ainsi que le temps et le coût de conception. L'analyse des propriétés des RdP procure aussi l'occasion de vérifier que tous les besoins pour le fonctionnement du système ont été pris en compte. Cet aspect est particulièrement important dans le cas des systèmes dont la sûreté de fonctionnement est un problème crucial. Par exemple l'apparition d'états non prévus lors de l'exploitation d'une centrale nucléaire peut conduire à l'occurrence d'évènements catastrophiques.

Les RdP sont représentés par des places et des transitions qui sont reliées par des arcs et des jetons (ou marques) décrivant le fonctionnement des différentes activités modélisées.

Dans ce chapitre nous présenterons les notions fondamentales des modèles réseaux de Petri de base, les réseaux de Petri de haut niveau (réseaux de Petri colorés) ainsi que leurs propriétés comportementales et structurelles. Les principales propriétés comportementales permettent par exemple la vérification de l'accessibilité (ensemble des marquages accessibles), la bornitude, la vivacité et la réinitialisation [10].

II-2- Les réseaux de Petri

II-2-1- Définitions

Un RdP est un graphe biparti orienté comportant deux types de sommets, les places et les transitions. Une place est généralement représentée par un cercle, et une transition est souvent représentée par une barre. Les places et les transitions sont connectées par des arcs. Un arc connecte soit une place à une transition, soit une transition à une place.

II-2-1-1-Définition Formelle [10]

Un RdP (généralisé marqué) est un 5-uplet

- **R= (P, T, Pré, Post,** Mo) avec:
- **P= {p_1, p_2,..., p_m}** est un ensemble fini de places.
- **T= {t_1, t_2,..., t_n}** est un ensemble fini de transitions, PUT ≠ Ø et P ∩T = Ø.
- **Pré : (P X T)** ⟶ **N** (ensemble des entiers naturels) fonction d'incidence avant, qui donne les poids des arcs orientés allant des places vers les transitions.
- **Post `(P X T)** ⟶ **N** (ensemble entiers naturels) fonction d'incidence arrière, qui donne les poids des arcs orientés allant des transitions vers les places.
- **Mo : P** ⟶ **N** le marquage initial (état initial).

Si Pré (p, t) = k (respectivement Post (p, t) = k), alors il existe k arcs orientés reliant simultanément la place p à la transition t (respectivement la transition t à la place p). On dit que k est le poids de l'arc (p, t) respectivement (t, p). Graphiquement, ce sont k arcs qui sont représentés par un seul arc évalué par son poids k. Généralement, la valeur par défaut du poids est égale à 1.

II-2-1-2- Matrice d'incidence

La matrice d'incidence ou de transition est définie par :

$$\mathbf{C = Post\ (p_i\ -t_i).\ -\ Pré\ (p_i,\ t_j)}$$

II-2-1-3- Notion de Marquage

Un marquage affecte à chaque place un nombre positif ou nul de jetons. Les jetons (les marquages) des places correspondent souvent à des ressources.

On appelle marquage d'un réseau de Petri R, l'application M définie de P⟶ N qui associe à chaque place p_i de P, le nombre de marques M (p) contenus dans p. On note alors :

$$\mathbf{M \in R\ (M_o)}$$

L'évolution de l'état du système réel modélisé par un RdP est caractérisée par l'évolution du marquage.

Remarques :

-L'état initial d'un RdP marqué est défini par un marquage initial noté M_o.

-Un RdP modélise la structure statique du système étudié. La notion de marquage exprime sa sémantique. Autrement dit, le franchissement des transitions qui entraîne le déplacement de marques entre les places traduit le fonctionnement dynamique du réseau.

II-2-1-4- Règle de Sensibilisation

Une transition t est dite sensibilisée si chacune des places d'entrée de p contient un nombre de jetons supérieur ou égal au poids de l'arc connectant p à t ($m_i \geq$ pré (p,t)).

II-2-1-5-Règle de Franchissement

1. Une transition sensibilisée peut être franchie ou non. Le franchissement est une opération instantanée.

2. Le franchissement d'une transition t sensibilisée fait déplacer de chacune de ces places d'entrée p un nombre de jetons égal au poids de l'arc reliant p à t et fait déposer sur chacune des places de sortie p un nombre de jetons égal au poids de l'arc reliant t à p.

Soit <R, M_o > un RDP marqué :

Une transition t de T est dite franchissable pour tout marquage M si et seulement si $\forall p \in P$ **M (p) ≥ Pré (p, t)** et l'on note **M [t >**.

Le marquage **M'**obtenu après le franchissement de t est tel que :

$$\forall p \in P \quad M'(P) = M(p) + C(p, t)$$

La figure 3 illustre le nouveau marquage des places p_1, p_2, p_3 après le franchissement de la transition t.

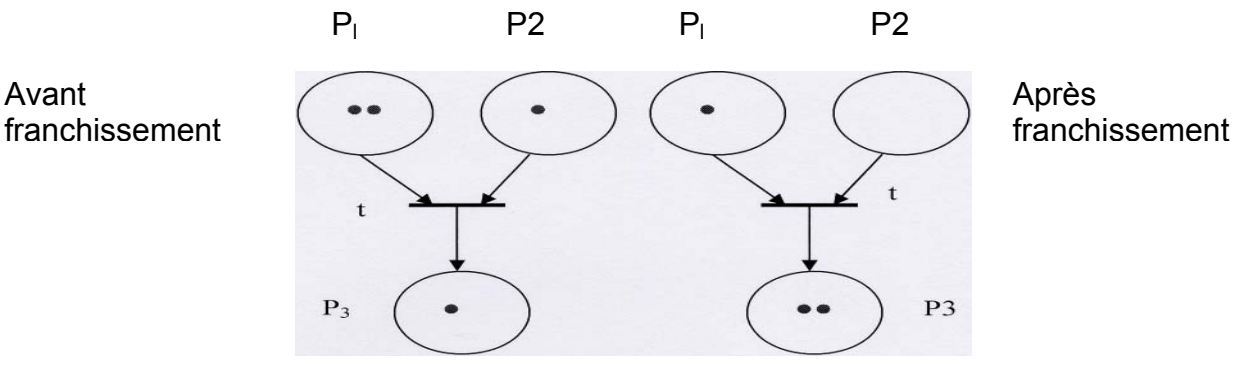


Fig 3 : Exemple de franchissement d'une transition sensibilisée

II-2-2- Quelques définitions particulières [10]

II-2-2-1-Transition source et transition puits

Une transition sans place d'entrée est appelée transition source. Une telle transition est toujours franchissable. Une transition sans place de sortie est appelée transition puits. Si une telle transition est tirée, les jetons sont prélevés dans la place d'entrée suivant les règles habituelles. Mais aucun jeton n'est produit en sorti.

Une transition source est souvent utilisé dans un modèle d'atelier, pour modéliser l'entrée des matières premières dans le système. De même une transition puits est souvent utilisée pour modéliser la sortie de produits finis du système.

II-2-2-2- Circuits élémentaires et boucles

Un circuit élémentaire est un chemin orienté qui part d'un sommet (place ou transition) du RdP et y revient sans jamais rencontrer plus d'une fois le même sommet. Les boucles permettront par exemple de modéliser le fait q'une pièce ne peut entrer dans une machine si une autre est déjà en cours de transformation sur cette machine.

Un RdP est dit **pur**, s'il ne contient pas de boucle.

II-2-3- Propriétés des RdP [10]

Les RdP peuvent être considérés comme des outils mathématiques ayant un certain nombre de propriétés. Lorsque ces propriétés sont interprétées dans le contexte du système modélisé, elles permettent aux concepteurs d'identifier la présence ou l'absence de propriétés fonctionnelles spécifiques au domaine d'application des systèmes étudiés. On peut distinguer deux classes de propriétés : les propriétés comportementales et les propriétés structurelles.

Les propriétés comportementales sont celles qui dépendent de l'état initial, ou du marquage du RdP. Par contre les propriétés structurelles dépendent de la structure (ou topologie) et ne dépendent pas du marquage initial du RdP.

II-2-3-1- Accessibilité

L'un des objectifs principaux de la modélisation de la phase conception d'un système est de savoir, si ce système présente toutes les propriétés désirées conformément aux besoins spécifiés et ne présente aucune propriété indésirable (un blocage mortel ou un fonctionnement dangereux).

Afin de vérifier si le système modélisé par un RdP peut atteindre un état spécifique inclus dans les états du comportement fonctionnel désiré, il est nécessaire de trouver la séquence de franchissements de transitions qui ferait passer le RdP du marquage M_o au marquage M_i où M_i représente l'état spécifique en question et la séquence des franchissements ayant conduit de M_o à M_i représente le comportement fonctionnel autorisé.

On dit que le marquage M est atteignable à partir du marquage M_o, s'il existe une séquence s de franchissements des transitions qui permette d'atteindre M_i à partir de M_o. On note alors :

$$\mathbf{M_o\ [s > M_i;}$$

II-2-3-2- RdP Borné et Sain

Dans les systèmes manufacturiers par exemple, les places sont souvent utilisées pour représenter des zones de stockage des produits ou des outils. Il est donc important pour les concepteurs de vérifier que la stratégie des commandes proposées permet de percevoir le cas de dépassement de capacité des zones de stockage. La propriété des RdP qui permet de vérifier le dépassement de capacité dans un système modélisé est la **bornitude.**

Un RdP est dit **k-borné** si pour tout marquage $\mathbf{M \in R\ (M_o)}$ atteignable à partir du marquage initial Mo, le nombre de jetons dans toute place p est toujours inférieur ou égal à **k** (nombre entier).

Un RdP est **sain,** s'il est **1-borné.**

II-2-3-3- RdP Conservatif

Si dans un système, les ressources ne sont ni créées ni perdues, alors le nombre total de jetons (marquant les places d'un RDP) utilisés pour représenter ces ressources doit rester fixe.

Un RdP est dit conservatif s'il existe un vecteur V=[, v_1, v_2,..., v_r,] avec p le nombre de places et V (p_i) >0 pour chaque $p_i \in P$ tel que la somme des jetons pondérés par les v_i reste constante, quel que soit le marquage M accessible à partir de M_o.

Un RdP est **strictement conservatif,** si la somme des jetons (sur toutes les places) reste constante quel que soit le marquage M atteint à partir de M_o.

II-2-3-4- RdP quasi–vivant et vivant

Cette propriété est étroitement liée à la situation de blocage. Un RdP modélisant un système sans blocage doit être vivant. Ce qui implique que pour tous les marquages M accessibles à partir de M_o, il est possible de tirer n'importe quelle transition du RdP en progressant le long d'une séquence de franchissement.

- Une transition t est quasi–vivante s'il existe un marquage accessible M tel que **M [t >.**

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

Si le réseau n'est pas quasi-vivant, alors il existe au moins une transition t qui n'est jamais franchissable et donc inutile au fonctionnement du système modélisé. Cette propriété désigne la possibilité de franchir au moins une fois la dite transition depuis un marquage initial. Elle n'est utile que pour les systèmes devant passer par une étape d'initialisation, après quoi il n'est plus nécessaire de la franchir.

Un réseau marqué est dit vivant si et seulement, si quel que soit la transition t et quel que soit le marquage accessible M, il existe une séquence (non vide) de transitions telle que **M [s.t >.**

Un réseau vivant est un réseau en fonctionnement permanent, dans lequel l'indisponibilité d'une certaine fonction (transition) signifie une erreur ou une panne. Dans ce cas, il est impératif qu'une procédure d'urgence soit prévue.

II-2-3-5- Réinitialisation et état d'accueil

Un aspect important dans le fonctionnement des systèmes réels, tels que les systèmes manufacturiers est l'aptitude de remédier aux erreurs. Ils doivent être en mesure de retourner des états d'échec vers les états de bon fonctionnement. Ces spécifications sont étroitement liées aux propriétés de réinitialisation et état d'accueil dans un RdP.

Pour tout marquage initial M_o, un RdP est dit réinitialisable, si pour tout marquage $M \in R(M_o)$, M_o est atteignable à partir de M.

Un état M d'un RdP est un état d'accueil, si pour tout marquage $M \in R(M_o)$, M est atteignable à partir de M.

Remarques :

1. Si le marquage initial est un état d'accueil, alors le réseau est rénitialisable.

2. Si un réseau admet un état d'accueil, alors il est facile de vérifier qu'il est vivant et sans blocage. Il suffit de montrer, dans le premier cas, que le réseau est quasi–vivant depuis l'état d'accueil, et dans le second cas, qu'au moins une transition est franchissable depuis ce marquage.

II-2-3-6- Etat puits et réseaux sans blocage

- Un marquage M est appelé état puits, si aucune transition n'est franchissable depuis M.

- Un réseau marqué est dit sans blocage si tout marquage accessible n'est pas un marquage puits.

II-2-4- Méthodes d'Analyse des Réseaux de Petri [10]

De nombreux travaux ont été consacrés à l'analyse des réseaux de Petri, les méthodes les plus utilisées sont les méthodes basées sur le graphe des marquages accessibles ou sur le graphe de couverture.

Ces méthodes sont basées sur l'énumération de tous les marquages possibles qui sont accessibles à partir du marquage initial M_o. En partant du marquage initial M_o, l'ensemble d'accessibilité est obtenu en tirant toutes les transitions sensibilisées dans tous les marquages atteignables à partir de M_o.

Dans l'arbre de couverture, les sommets sont étiquetés par les marquages et les arcs sont étiquetés par les transitions. Le marquage initial représente la racine de l'arbre.

L'ensemble d'accessibilité peut devenir infini, soit lorsque les marquages se répètent (fonction cyclique), soit lorsque le nombre de jetons dans une place peut augmenter indéfiniment.

Pour une représentation finie de l'arbre de couverture, dans le premier cas, tout noeud déjà rencontré sera un noeud terminal et dans le second cas, on utilisera le symbole w pour représenter le marquage d'une place dont le nombre de jetons peut devenir infini.

D'un point de vue pratique, cette approche n'est intéressante que pour des RdP bornés et des RdP de taille (nombre de places et de transitions) relativement réduite.

L'arbre de couverture permet d'étudier un certain nombre de propriétés du RdP par exemple, si on constate la présence d'un symbole w sur un sommet, on peut conclure que le RdP n'est pas borné, autrement il sera borné. Pour un RdP borné, le nombre de sommets de l'arbre de couverture est égal au nombre de marquages accessibles à partir du marquage initial M_o. Dans ce cas, l'arbre de couverture est appelé **Graphe des Marquages Accessibles** (GMA).

Le graphe des marquages accessibles est composé de noeuds qui correspondent aux marquages accessibles, et les arcs correspondent aux franchissements de transitions faisant passer le RdP d'un marquage à un autre.
Le graphe des marquages accessibles permet de vérifier les propriétés (bornitude, vivacité, états d'accueil, ... etc) du système étudié.

II-2-5- Classification des Réseaux de Petri

Différentes modifications ont été effectuées sur le modèle à RdP de base. Ces modifications sont essentiellement de deux types :

1. **Les abréviations:** qui correspondent à des représentations simplifiées, pour rendre plus claire la représentation graphique,

2. **Les extensions:** qui correspondent à des modèles auxquels des règles de fonctionnement supplémentaires ont été rajoutées, afin d'enrichir le modèle de base, et de permettre d'élargir son champ d'application.

Les modèles RdP rencontrés dans les applications peuvent être subdivisés en trois classes principales:

- **Les RdP ordinaires (le modèle de base):** Dans ces réseaux, tous les arcs ont un poids de 1, on y trouve un seul type de jetons, les capacités des places sont supposées infinies. Une transition est franchie **ssi** toute place de son entrée contient au moins un jeton et le temps n'intervient pas.

- **Les abréviations**: On peut toujours faire correspondre un RdP ordinaire à une abréviation. On peut citer les RdP Généralisés, les RdP Colorés et les RdP à capacité finie. Ils ont le même pouvoir de description que les RdP ordinaires.

- **Les extensions**: On peut distinguer deux sous classes dans les extensions:

  - Les RdP autonomes : C'est des réseaux ayant le même pouvoir de description que les machines de Turing : RdP avec arc inhibiteur et les RdP avec priorité.

  - La deuxième sous classe correspond aux RdP non autonomes, qui décrivent le fonctionnement de systèmes dont l'évolution est considérée par rapport à des événements extérieurs et /ou le temps : RdP synchronisés, RdP temporisés et RdP stochastiques.

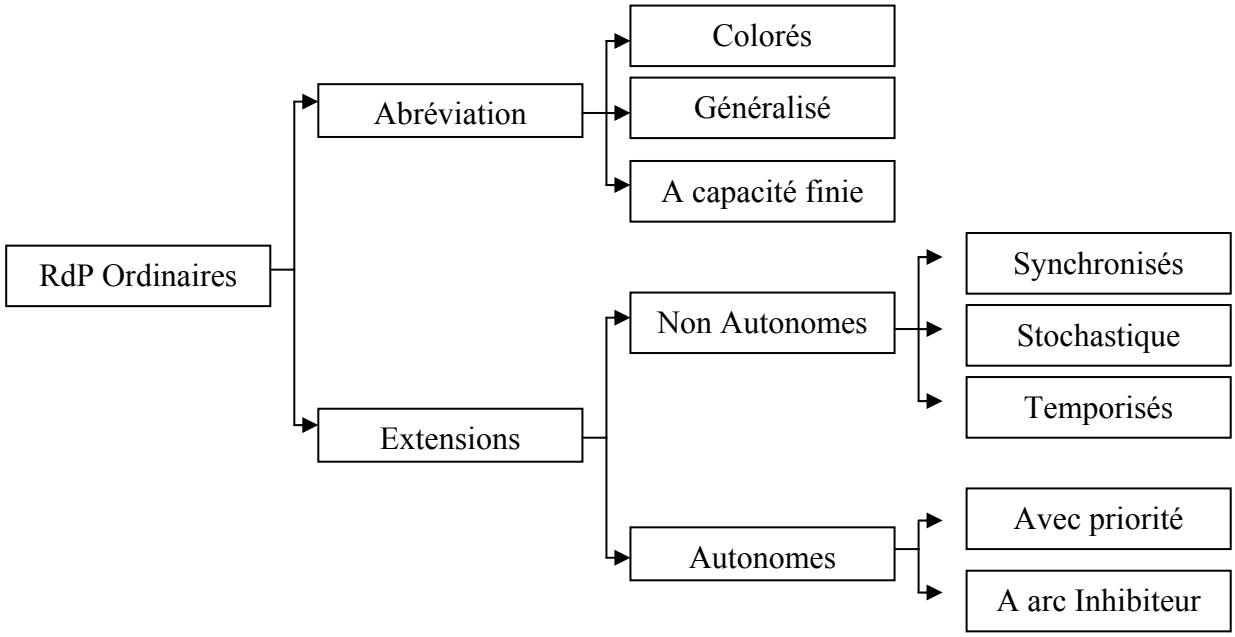


Fig 4: Classification des Réseaux de Petri

II-2-5-1 Quelques abréviations

- **RdP Colorés**

Plusieurs modèles de RdP dans lesquels les jetons ont une identité, ont été proposés. Ils sont appelés **RdP de haut niveau.** Les **RdP prédicat-transition** et les **RdP colorés** sont des **RdP de haut niveau.** Dans les RdP colorés, l'information peut être associée aux jetons. Cette information est appelée *'couleur'* du jeton. Les jetons sont donc identifiés à l'aide de leurs couleurs. Les couleurs des jetons peuvent être assez complexes, telles que la description du contenu de paquets de messages ou le contenu d'une base de données.

Ils sont utilisés pour simplifier la représentation de certains systèmes complexes comportant des symétries dans leur fonctionnement. Toutes les composantes symétriques sont condensées dans une seule. Toute l'information sur une composante est portée par les jetons qui lui correspondent. La distinction entre les composantes est faite à l'aide des couleurs attribuées aux jetons. Nous reviendrons plus en détail sur ce type de RdP dans la section suivante.

- **RdP à capacité finie**

Pour modéliser certains systèmes physiques, il est souvent nécessaire de considérer une limite supérieure pour le nombre de jetons qu'une place peut contenir. C'est le cas par exemple, lorsque la quantité d'une ressource ou la capacité d'un stock est limitée. A chaque place **P**, on associe une capacité Q(P) (le nombre maximal de jetons que peut contenir **P).** On obtient ainsi, un **RdP à capacité finie.**

II-2-5-2 Quelques extensions

II.2-5-2-1 RdP autonomes

- **RdP avec arcs inhibiteurs**

Le pouvoir d'expression d'un RdP de base peut être augmenté en rajoutant aux conditions de franchissement une condition dite de test à zéro, sur certaines places. Cette condition supplémentaire est caractérisée par un type particulier d'arcs appelés arcs inhibiteurs. Un arc inhibiteur connecte une place à une transition (il part d'une place et aboutit à une transition).

- **RdP avec priorité**

Ce type de réseau est utilisé, lorsqu'on veut faire un choix (pour le tir) entre plusieurs transitions qui sont sensibilisées simultanément. On définit une relation d'ordre partiel sur les transitions. On peut dire que c'est un RdP pour lequel le conflit est résolu par l'introduction d'un ordre de franchissement.

II-2-5-2-2 RdP non autonomes

Les RdP non autonomes permettent une approche quantitative. Les réseaux de Petri non autonomes sont obtenus par des extensions permettant de décrire non seulement ce qui se passe, mais aussi quand cela devrait se passer. Ces extensions permettent de modéliser des systèmes dont les franchissements sont synchronisés par des événements externes, et /ou dont l'évolution dépend du temps.

- **RdP Synchronisés**

Dans un tel réseau, un évènement externe (e), est associé à chaque transition. Une transition est franchie, lorsque l'événement qui lui est associé se produit. Si une transition est synchronisée sur l'évènement (e), alors elle peut être franchie dès qu'elle est validée.

- **RdP Temporisés**

Un RdP temporisé permet de décrire un système dont le fonctionnement dépend du temps. Par exemple, un certain temps peut s'écouler entre le début et la fin d'une opération. Un RdP temporisé permet de tenir compte de ce temps. Un tel réseau est utile pour l'évaluation des performances d'un système.

- **RdP Stochastiques**

Dans un RdP temporisé, une durée fixé est associée à chaque place ou à chaque transition du réseau. Ce modèle est adapté à l'étude des systèmes dans lesquels les durées des opérations sont fixes. C'est le cas par exemple, des systèmes de production où le temps de traitement d'une unité par une machine est constant. Il existe cependant, des systèmes qui ne peuvent être modélisés de manière correcte à l'aide de durées constantes. C'est le cas par exemple pour la durée de bon fonctionnement (entre deux pannes) d'une machine. Cette durée peut être modélisée par une variable aléatoire. L'hypothèse communément utilisée est que dans un RdP stochastique, les temporisations sont considérées comme des variables aléatoires exponentiellement distribuées.

II -3- Les Réseaux de Petri colorés (RdPC)

II-3-1- Introduction

Les réseaux de Petri modélisent aisément les synchronisations et d'autres comportements similaires. Néanmoins, ils présentent deux limitations importantes. Il est impossible de transporter une information structurée car les places ne contiennent que des jetons uniformes et il faut définir de manière explicite le comportement d'objets différents mais qui se comportent de manière identique.

De plus les systèmes modélisés par des RdP ordinaires peuvent développer des évolutions excessives, par conséquent produire des réseaux de grande taille difficilement manipulables.

Pour pallier à ces problèmes différentes extensions dont les réseaux de Petri colorés.

Les réseaux de Petri colorés ont été introduits par Jensen [37]. Ils sont utilisés pour simplifier la représentation de certains systèmes complexes comportant des symétries dans leur fonctionnement. Toutes les composantes symétriques sont condensées dans une seule. Toute l'information sur une composante est portée par les marques qui lui correspondent. La distinction entre les composantes est faite à l'aide des couleurs attribuées aux jetons.

Dans les réseaux de Petri colorés, l'information peut être associée aux jetons. Cette information est appelée `couleur ` du jeton. Les jetons sont donc identifiés à l'aide de leurs couleurs. Chaque transition peut être franchie de différentes manières représentées par les différentes couleurs de franchissement que l'on associe à la transition. La relation entre les couleurs de franchissement et le marquage coloré concerné est définie par des fonctions associées aux arcs.

La couleur associée à la marque peut être un n-uplet et peut donc porter une information complexe telle que par exemple la nature et la position d'un objet dans un stock. Il peut y avoir disparition et création de nouvelles couleurs par le franchissement de transition.

II -3-2-Notion de Couleur [10]

Dans un RdP ordinaire, les places ne contiennent que des jetons uniformes, on ne pas distinguer l'information relative à une marque dans une place. Cependant il n'est possible de caractériser qu'un seul type d'objets, en quantité quelconque dans une place donnée. Par contre dans un RdP coloré, l'information est portée par le couple (place, couleur de marque). Une couleur représente une information qui permet de distinguer les éléments entre eux, par exemple deux tâches entre elles ou deux ressources entre elles.

II-3-3-Notion de Fonctions [10]

Les fonctions réalisent une transformation linéaire des couleurs associées à une transition. Une fonction quelconque est définie en calculant l'image de chaque couleur de la transition.

II-3-4- Définitions

II-3-4-1- Définition d'un RdP Coloré [10]

Un RdP coloré est un sextuplet

R= (P, T, Pré, Post, Mo, C) avec:

P = {p_1, p_2,..., p_m}, l'ensemble des places,

T = {t_1, t_2,..., t_n}, l'ensemble des transitions,

C= {c_1, c_2, ..., c_k}, l'ensemble des couleurs,

Pré et **Post** sont des fonctions relatives aux couleurs de franchissement,

M_o est le marquage initial.

II-3-4-2- Le Graphe

Comme dans tout RdP, le graphe d'un RdP coloré comporte deux types de noeuds, les places et les transitions, reliés par des arcs orientés [10].

Les places

Une place est représentée par un cercle. Elle peut contenir des marques colorées.

Les transitions

Une transition est représentée par un trait. A chaque transition est associé un ensemble de couleurs. Par exemple la transition t_1 de la figure 5 peut être franchie par rapport à la couleur <b>, à la couleur <v >, ou à la couleur <o>.

Les arcs

Les arcs orientés relient une place à une transition ou une transition à une place. Le poids de l'arc est une fonction Pré ou Post qui établit une correspondance entre chaque couleur de la transition et les couleurs de la place [(Pré (p_i, t_i / C_k) ou Post (p_i, t_i / C_k]. Par exemple dans la figure 5 Pré (p_1, t_1 / <v>) = F(<v>)=<v> +<b>

Il peut y avoir une transformation de couleur lors du franchissement d'une transition. Ainsi dans l'exemple de la figure 5, le franchissement de t_1 par rapport à la couleur <b> retire une marque de couleur <v> de p_1 et ajoute une marque de couleur <b> dans la place p_2.

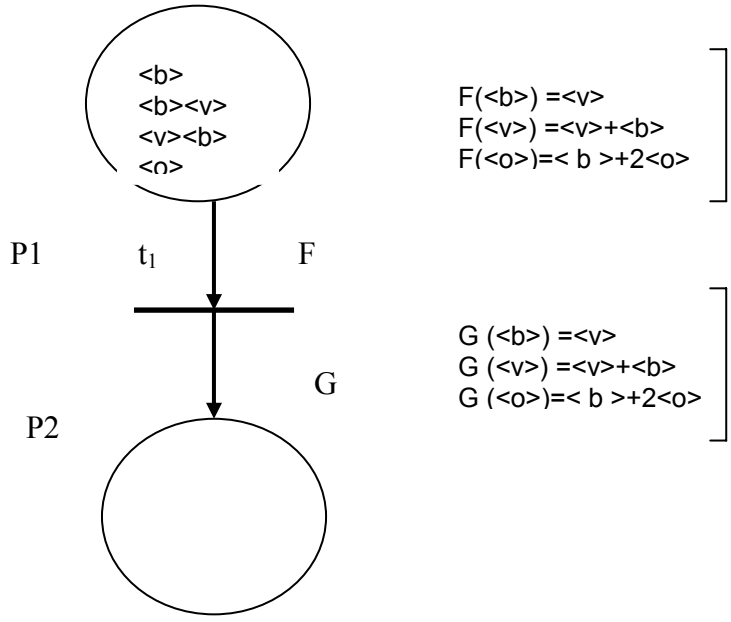


Fig 5 : Réseau de Petri coloré

II-3-4-3-Transition validée

Soit C (t_j,), l'ensemble des couleurs associées à la transition t_j. Cette transition peut être franchie par rapport à une de ces couleurs (le nombre de couleurs dans C (t_j,).

Soit une couleur quelconque de C (t_j,), et soit M le marquage courant du RDP coloré. La transition t_i est validée par rapport à la couleur C_k pour tout marquage M, ssi le nombre de marques contenu dans toute place p_i en amont de t_i est supérieur ou égal à Pré (p_i, t_j / C_k).

$$\mathbf{M\ (p_i) > Pré(p_i, t_j / c_k)}$$

Pré (p_i, t_j / C_k) est l'image de la couleur C_k par la fonction poids de l'arc reliant la place p_i à la transition t_j,

Dans l'exemple de la figure 5, la transition t_1 est validée par rapport à la couleur <b>, car F (<b>) = <v> et la place p_l contient deux marques de couleur <v>. Elle est également validée par rapport à la couleur <v>. En revanche t_1 n'est pas validée par rapport à la couleur <o>, car il faudra une marque de couleur <b> et deux marques de couleur <o> et la place p_i ne contient qu'une seule

II-3-4-4-Règle de franchissement d'une transition

Une transition t_i validée par rapport à la couleur C_k peut être franchie. Son franchissement qui sera noté t_i / C_k consiste à effectuer simultanément les deux opérations suivantes :

1. 1-On retranche à toute place p_i; en amont de t_j une quantité de marques égale à Pré (p_i, t_i / C_k).

2. 2-On ajoute à toute place p_i en aval de t_j, une quantité de marques égale à Post (p_i, t_i / C_k).

Soit M' le marquage obtenu après le franchissement de la transition t_i par rapport à la couleur C_k. Ce marquage se déduit du marquage M par la relation :

$$\mathbf{M'(P_i) = M(p_i) + Post(p_i, t_j / C_k) - Pré(p_i, t_j / C_k)}$$

Dans l'exemple de la figure 5, la transition t_1 est validée par rapport à la couleur <b>. Elle peut être franchie par rapport à cette couleur. Ce franchissement consiste d'une part à retirer une marque de couleur <v> de la place p_{1i} car, F (<b>) = <v>, et d'autre part ajouter une marque de couleur <b > à la place p_2 car G (<b>) = <b>.

Remarque :

Lorsqu'une transition est franchie par rapport à une couleur donnée, c'est cette couleur qui fixe les règles de franchissement. Toutes les images de fonctions associées soit à un arc entrant ou à un arc sortant de cette transition sont calculées à partir de cette couleur.

II-3-5- Propriétés des RdP colorés

Lorsqu'un RdP coloré est conçu, il est alors possible de valider les spécifications qu'il représente. Comme nous l'avons vu, un RdP coloré n'est qu'une représentation condensée du RdP ordinaire ou généralisé obtenu par dépliage. Il s'ensuit que les propriétés des RdP colorés sont les mêmes que celles des RdP non colorés. Nous allons revoir les principales notions et propriétés des RdP non colorés et analyser comment elles se présentent dans un RdP coloré.

II-3-5-1- Marquage

Le marquage est un vecteur dont chaque composante est une somme pondérée de marques colorées.

II-3-5-2- Matrice d'incidence

Comme pour un RdP non coloré, on définit une matrice d'incidence C. l'élément C(i, j) de cette matrice est égal à la différence des deux fonctions Post (p;, t_l) et Pré (p_i, t_j).

$$C = Post(p_i - t_j) - Pré(p_i - t_j).$$

II-3-5-3- Franchissement d'une Transition

Au franchissement d'une transition t_j dans un RdP non coloré correspond, le franchissement d'une transition t_j par rapport à une couleur C_k dans un RdP coloré.

La généralisation est évidente. Un RdP coloré est borné si, pour tout marquage accessible, toute place contient un nombre fini de marques.

Un RdP coloré est dit sain, si pour toute place, on a au maximum une marque de chaque couleur.

II-3-5-5- Vivacité et Blocage

Un RdP coloré est vivant, si son RdP déplié est vivant. Donc si pour tout marquage accessible il est possible de franchir n'importe quelle transition par rapport à n'importe laquelle de ses couleurs de franchissement.

Un RdP coloré est sans blocage, si pour tout marquage accessible, il y'a au moins une transition franchissable pour au moins une de ses couleurs.

II-3-6- Analyse des Réseaux de Petri Colorés

Si le nombre des couleurs est fini dans les RdPC, il est possible d'énumérer les marquages accessibles du réseau de Petri coloré. L'analyse par énumération des marquages accessibles peut se faire exactement comme pour un réseau de Petri ordinaire. Il faut toutefois remarquer que le risque d'explosion combinatoire (en terme d'états) est très probable, puisqu'il est nécessaire de différencier deux jetons de couleurs différentes. Ce qui revient finalement à travailler sur des réseaux de Petri ordinaires de grande taille obtenus en dépliant le réseau coloré (si le nombre de couleurs est bien fini).

II-3-7- Conclusion

Dans ce chapitre, nous avons présenté les notions fondamentales des modèles de réseaux de Petri de base et leurs extensions, tout en montrant l'intérêt de l'utilisation de ces derniers pour la modélisation des systèmes complexes, présentant des propriétés de synchronisation, de parallélisme, de conflits, des problèmes d'exclusion mutuelle et de partage de ressources.

La modélisation par les réseaux de Petri permet d'analyser et de vérifier certains critères qualitatifs. Cette analyse permet en particulier, de s'assurer que les systèmes modélisés fonctionnent correctement et ne tombent pas dans un état de blocage ou dans un état indésirable.

Cependant une question reste posée, comment utiliser les modèles de réseaux de Petri pour obtenir une représentation formelle des systèmes hétérogènes, coopératifs et distribués modélisés par une approche système multi-agents ? Dans le chapitre suivant nous allons présenter quelques modèles de modélisation des systèmes multi-agents à base des réseaux de Petri proposés par la communauté SMA.

Chapitre III
Approches Systèmes Multi-Agents
&
Les Réseaux de Petri

III-1- Introduction

Le modèle réseau de Petri (les réseaux de haut niveau) est destiné à la modélisation, à l'analyse et au prototypage de systèmes dynamiques à activité parallèle. Aussi le domaine des SMA s'apprête bien à cette approche. Deux questions essentielles restent cependant posées. Que modélise – t- on avec un réseau de Petri ? Et quel apport en espère –t-on?

La première approche consiste à se limiter à une partie du système, par exemple, le comportement de l'agent lors de l'affectation des tâches, et le comportement d'agents en interaction. Ainsi El Fallah [20], et Mazouzi [47] ont introduit le formalisme RdPC dans la communauté multi-agents pour la modélisation des interactions, et ce dans un but de répondre aux exigences de la validation des protocoles extensibles et ouverts. Leur approche consiste à utiliser les réseaux de Petri en vue de décrire des enchaînements d'interaction inter-agents, et des structures de contrôle de l'activité d'interaction entre les agents. En effet ils utilisent les réseaux de Petri colorés (RdPC), car le contrôle d'interaction peut être développé de façon excessive et produit des réseaux de très grande taille difficilement manipulables. L'idée sous-jacente des réseaux de Petri colorés est précisément de regrouper des structures similaires de comportements sans perte ni d'information, ni de visualisation graphique de ces structures.

La deuxième idée est de modéliser l'ensemble du système et étudier son comportement en utilisant les méthodes de vérification des réseaux de Petri. On peut trouver cette approche traitée de manière assez similaire dans le formalisme Val proposé par F. Vernadat [57], de Ferber et Magnin [45]. Cependant, ces travaux se sont généralement concentrés sur la représentation de modèles cognitifs et communicants, en faisant l'hypothèse selon laquelle le nombre d'agents reste constant.

Innocent Bakam [01] a proposé une autre approche qui consiste en la formalisation des systèmes multi-agents par les réseaux de Petri (un couplage simulations et approches formelles de modélisation). En effet les simulations d'un système multi-agents permettent d'observer uniquement l'existence de certaines situations atteignables. Il serait pourtant nécessaire dans le contexte de la gestion des ressources renouvelables par exemple de vérifier la non observabilité de certaines configurations du système, étant donnée une situation initiale et sous certaines hypothèses d'évolution. Cependant les simulations d'un SMA ne peuvent pas nous renseigner sur la non existence de situation de blocage, et sur la viabilité du système en fonction des hypothèses d'évolution. Bakam propose un modèle de validation des systèmes multi-agents à base du formalisme réseaux de Petri colorés qui est à la fois un modèle mathématique et un langage graphique permettant la modélisation des systèmes complexes. Cette approche de modélisation prend en compte les interactions spatiales entre les différentes entités du système (modèles d'agents situés).

III-2 Les Systèmes Multi-Agents et les Réseaux de Petri

Les réseaux de Petri proposent des techniques de vérification de propriétés sur le comportement des systèmes. De nombreux travaux ont été effectués sur la conception des systèmes multi-agents par les réseaux de Petri : [01], [20], [45], [57]. Nous allons présenter dans cette section les différents modèles de conception de systèmes multi-agents à base des réseaux de Petri proposés par la communauté SMA.

III-2 -1- Le Modèle VAL

Le processus de développement de systèmes multi-agents destiné à des applications de contrôle de systèmes complexes repose généralement sur un cycle itératif de prototypage et de simulation. Cette approche ne suffit pas lorsque l'utilisateur attend une certaine fiabilité comportementale. Vernadat s'est intéressé à ce type d'applications comportant des caractéristiques de répartition du contrôle et de coopération entre agents autonomes. Si l'apport des approches multi-agents dans ces domaines est indéniable et la faisabilité démontrée à travers un certain nombre de réalisations, la question se pose de définir, un cadre méthodologique de développement et des outils logiciels associés permettant d'une part d'assister ce processus de la conception à la réalisation et d'autre part de permettre la vérification et la validation des applications développées. En effet Vernadat propose un processus (formalisme VAL) de développement et de validation de systèmes multi-agents en associant dans un couplage fort deux approches complémentaires, le prototypage à l'aide d'outils formels (réseaux de Petri Predicat-Transition) et la simulation dans l'environnement de programmation cible. Deux descriptions sont utilisées : un modèle acteur et une représentation par système de transitions. Le modèle acteur dispose d'une implémentation, un système de transitions est bien adapté à la validation.

VAL est un formalisme de description de systèmes d'agents communicants basé sur la programmation logique et les réseaux de Petri (Prédicats/Transitions). Un objectif principal est la modélisation (description et vérification formelle) de systèmes dynamiques d'agents communicants dont le nombre (le type) d'agents et leur organisation évoluent dynamiquement au cours du temps.

Le formalisme VAL propose donc une démarche de conception mixte basée sur deux approches complémentaires : prototypage à l'aide d'outils formels et la simulation dans un environnement de programmation cible (PRAL).

Le modèle de description [57] est basé sur l'approche acteur, utilisant le langage acteur PRAL. Ce langage qui repose sur un modèle de calcul concurrent, permet de concevoir une application comme un ensemble d'acteurs communiquant par envoi de messages asynchrones avec leurs accointances. Un acteur au sens PRAL est un agent élémentaire autonome. Il lui est associé un automate qui décrit les états dans lesquels l'acteur est susceptible d'évoluer. Un acteur PRAL est alors défini par son interface, ses accointances et l'ensemble de ses comportements.

Le formalisme VAL est une extension des réseaux de Petri Prédicat–

transition, dans lesquels les transitions portent des prédicats. Ce formalisme prend en compte de manière explicite la communication et le dynamisme. En plus des pré-conditions et des post-conditions des réseaux de Petri classiques, les transitions comprennent des classes de synchronisation, les messages attendus, les messages à émettre ainsi que les instances d'agents à créer.

Les acteurs PRAL sont traduits dans le formalisme VAL afin d'exploiter les potentialités de vérification de celui –ci. La vérification concerne les propriétés attendues du système. Elle se fait par construction du graphe d'accessibilité (l'espace d'états accessibles) du système.

III-2 -2- Le Modèle BRIC

Magrin et Ferber [45] proposent la définition d'un SMA par la composante modulaire de type BRIC (Block-like Representation of Interactive Components) et réseaux de Petri (réseaux de Petri à arcs inhibiteurs). BRIC est un langage [26] permettant de concevoir et de réaliser des systèmes multi-agents à partir d'une approche modulaire. Ainsi les agents et l'environnement sont des composants et un SMA consiste en un ensemble de composants reliés.

Un composant BRIC est une structure logicielle caractérisée extérieurement par un certain nombre de bornes d'entrée et de sortie qui lui servent d'interface avec les autres composants. De manière interne, un composant peut être défini, soit par un ensemble de composants ou par un réseau de Petri. Un composant peut aussi être une boite noire pour représenter les fonctionnalités externes à l'agent ou au système.

La modélisation de l'environnement [26] en BRIC se fait sous la forme d'un composant qui reçoit les influences produites par les agents, les combine avec l'état courant de l'environnement et renvoie aux agents les informations sur le nouvel état (consommation, perception).

Le formalisme BRIC peut être utilisé pour modéliser un système multi-agents communicant dans lequel les agents sont des composants dont les bornes d'entrée/sortie sont liées aux bornes d'un module d'acheminement des messages. Ce dernier se contente de récupérer tous les messages et de les envoyer aux agents destinataires à partir de l'adresse indiquée dans le message [45].

Les réseaux de Petri possèdent de remarquables propriétés qui en font un outil de vérification formelle très apprécié. Cependant, les réseaux complexes deviennent illisibles. L'idée de les encapsuler dans des composants permet donc à la fois d'améliorer leur lisibilité et leur réutilisation [46].

III-2 -3- Représentation et manipulation de Plans à l'aide des réseaux de Petri

La résolution distribuée de problèmes au moyen des systèmes multi-agents nécessite le plus souvent la construction de plans globaux à partir de plans locaux élaborés individuellement par les agents. Donc chaque agent élabore un plan (plan local) de résolution à partir de ses connaissances et peut le modifier ou l'enrichir (plan global) en interagissant avec l'environnement ou d'autres agents. Dans ce contexte, El Fallah à proposé un formalisme de représentation de plans basé sur les réseaux de Petri hiérarchiques de haut niveau (RHHN) [18], tout en mettant en œuvre les avantages qu'offre ce formalisme à savoir, ses possibilités d'abstraction, les transformations que l'on peut effectuer et les méthodes de vérification et d'évaluation associées.

D'après El Fellah, le formalisme des RHHN est destiné à la modélisation de systèmes distribués. Il permet de spécifier des activités parallèles ou reparties. Appliqué à la spécification de systèmes multi-agents, ce formalisme offre les avantages suivants :

- Expression naturelle et graphique des synchronisations des activités, parallèles que sont les exécutions des tâches des agents,
- Spécification de la communication entre agents,
- La prise en compte des flux d'informations,
- Ordonnancement des actions (relation d'ordre causale ou temporelle) des plans de résolution, - modification des plans de résolution par fusion de plans,
- Affectation dynamique des tâches,
- Partage d'informations à travers les places des agents,
- Analyse qualitative et quantitative des réseaux de Petri modélisant des plans de résolution.

De plus, l'aspect hiérarchique des RHHN permet de considérer un système distribué à des niveaux d'abstraction multiples et de raffiner des transitions abstraites (une transition peut être une abstraction d'un sous-réseau). Par conséquent la notion de hiérarchie offre la possibilité de masquer des parties du réseau par abstraction et de les démasquer par dépliage en cas de besoin. Le dépliage d'un objet (place ou transition) génère un sous réseau d'un niveau hiérarchique immédiatement inférieur. Un plan est une abstraction de la résolution du fait qu'il ne contient que les informations pertinentes aux interactions des agents. Cependant, toutes ces informations n'étant pas nécessaires à chaque instant de la résolution, la hiérarchie fournit le moyen de ne présenter que la vision nécessaire dans un contexte précis de résolution.

- **Représentation d'un plan de résolution par les réseaux de Petri**

Les RHHN ont été utilisés pour représenter un plan de résolution où l'état d'un réseau est déterminé à partir du marquage des places et des transitions activées avec une nouvelle sémantique décrite ci-dessous:

Transition : Une transition modélise une action du plan de résolution et son franchissement correspond à l'exécution de l'action modélisée. On distingue deux types d'action :

Une action abstraite qui est une action décomposable en actions élémentaires. Elle représente une vue abstraite du plan de résolution (sous-réseau) et peut être raffinée.

Une action élémentaire au plus bas niveau de la hiérarchie est irréductible et sera affectée à un agent par un mécanisme d'instanciation. On associe à une transition un domaine constitué d'informations relatives à son exécution (les agents capables de l'accomplir et les données en entrée/sortie). La transition sera instanciée (affectée) par l'agent responsable de son exécution dans le cas d'une action élémentaire et par l'agent responsable de l'exécution du plan dans le cas d'une action abstraite.

Place : Une place modélise une information relative au processus de résolution ou au problème, ou à un état du processus de résolution. Dans le premier cas, elle représente le partage d'informations entre les actions d'un même agent ou d'agents distincts. Dans le second, elle représente le séquencement des actions et permet la création d'actions parallèles. Nous associons à une place un domaine symbolisant la nature de l'information, la nature de la synchronisation etc.

Arc : Un arc relie une place à une transition et inversement. La valuation d'un arc exprimée sous forme de fonctions représente :

- le flux d'informations en provenance ou à destination d'un agent (sélection pertinente des informations plus ou moins complexes telles que des méthodes),
- un changement d'état au moyen de fonctions de changement d'états.

La démarche basée, en partie sur l'élaboration de plans globaux à partir de plans locaux est appliquée aux systèmes multi-agents et peut être utilisée lors d'un processus de résolution afin de valider structurellement les plans de résolution à des niveaux d'abstraction multiples. Le formalisme de représentation de plans dans un système multi-agents basé sur les réseaux de Petri proposé par Amal El Fallah est indépendant du domaine d'application. Sa structure hiérarchique autorise différentes abstractions de la résolution d'un problème. Les compositions de réseaux de Petri fournissent un mécanisme efficace de fusion de plans; ce mécanisme supporte des modalités différentes qui peuvent fournir plusieurs plans globaux à partir des mêmes données. Enfin la vérification et l'évaluation des plans ainsi formés repose sur les méthodes issues de la théorie des réseaux de Petri.

III-2 -4- Modélisation des Interactions par les RdPC

Ce modèle a été proposé par H. Mazouzi dans ses études relatives à l'ingénierie des protocoles d'interactions [47]. Cependant ils proposent un modèle à base de réseaux de Petri colorés permettant de modéliser toute conversation entre agents quel que soit leur nombre avec peu de places et de transitions.

Dans la modélisation des interactions par les réseaux de Petri colorés, le domaine des couleurs peut correspondre aux identités des agents, aux types de messages...etc.

Les transitions représentent les traitements et les évènements d'envoi et de réception de messages.

Les fonctions sont construites avec des variables de même type que les jetons circulant dans le réseau de sorte à assurer un contrôle global cohérent reflétant ainsi une conversation rationnelle.

- **Le Modèle RdPC pour Les Protocoles d'Interactions**

D'une façon générale, les places représentent les états des agents dans le cadre d'une conversation.

Les transitions modélisent des unités de traitement telles que des envois et réceptions de messages et le traitement des messages reçus par un agent. Le traitement local d'un message par un agent récupère les paramètres en entrée (jetons de couleur) et fournit en sortie d'autres jetons conformément aux règles de franchissement.

Les arcs représentent les mécanismes d'envoi de messages et permettent les changements d'état d'une conversation correspondant au marquage courant dans le réseau.

Il y a toujours un initiateur de conversation. Un agent ayant le rôle d'initiateur commence la conversation en émettant le premier message. Un marquage initial est défini de telle sorte que le rôle qui initie la conversation (et seulement ce rôle) puisse franchir la toute première transition.

Afin de structurer les actes de communication d'un protocole (modéliser le contrôle), les notions de place et de transition seront utilisées pour représenter les points de synchronisation.

- **Exemple de Modélisation du Protocole FIPA- Inform**

Le protocole **FIPA –Inform** [31] et un acte communicatif simple pour passer une information (position) d'un agent à un autre, donc il fait interagir deux agents.

L'agent initiateur envoie un message Inform à un autre agent (action représentée par $\mathbf{t_1}$) et l'agent destinataire reçoit le message ($\mathbf{t_2}$) et traite ($\mathbf{t_3}$). La conversation se termine quand les deux agents franchissent les transitions de fin

respectives **(t_4** et **t_5).** (Voir la figure 6).

Détaillons les domaines et les fonctions de couleur. Les domaines sont construits à partir de deux ensembles de base A (agent initiateur) et B (agent destinataire).

- **Transitions :**

  $t_1 = t_5 = A$
  $t_2 = t_3 = t_4 = B$

- **Places**

  $p_1 = p_2 = B$
  $p_3 = p_4 = p_5 = B$

- **Fonctions**

  $X(A) = A$
  $Y(A) = Y(B) = B$

Interprétation

Lors du franchissement de la transition **t_1** instanciée par l'unique jeton A. La fonction (X) fournit un jeton A et la fonction (Y) fournit un jeton B. Par la suite dans les franchissements, les fonctions (X) et (Y) jouent le rôle de la fonction identité faisant circuler les jetons A et B dans leurs sous réseaux respectifs

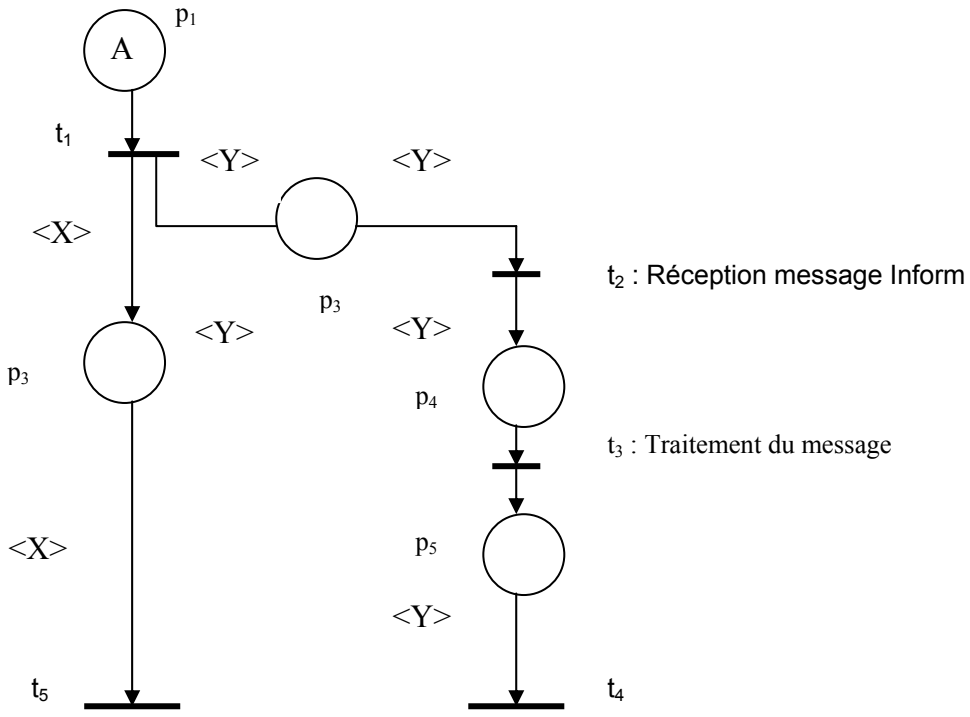


Fig 6 : Modélisation par un RdPC du protocole FIPA-Inform.

➢ Propriétés des Réseaux de Pétri colorés

Une des principales caractéristiques des réseaux de Petri est qu'ils permettent d'étudier certaines propriétés des systèmes concurrents. Nous allons présenter dans cette section les propriétés de ces derniers en vue d'étudier les propriétés qualitatives des systèmes, en particulier dans les protocoles d'interaction dans les systèmes multi agents.

1- Accessibilité d'un marquage

Dans un protocole d'interaction, l'accessibilité garantit la contrôlabilité de l'interaction. Elle permet de conduire une conversation à un état désiré à partir de l'état initial.

2- Caractère borné est sain

Dans les protocoles d'interaction les places véhiculent souvent des instances d'agents et des messages. Le caractère borné garantit que le nombre d'instances d'objets au niveau des places restera limité pour un état initial donné, c'est à- dire que les messages sont constamment traités et ne s'accumulent pas dans le réseau

Un réseau de Petri non borné pour un protocole peut représenter une croissance incessante d'envois de messages sans aucun traitement associé, de déclenchement d'un traitement à l'infini, alors que dans la réalité, ceci n'est guère possible dans le cadre des conversations inter- agents.

3- Vivacité et absence de blocage

Dans les protocoles d'interaction, l'absence de blocage garantit qu'étant donné un état initial, au moins une opération peut être exécutée quel que soit l'état atteint au cours d'une conversation. La vivacité garantit que pour un état initial donné, toutes les opérations peuvent toujours s'exécuter quelles soient les décisions admissibles possibles des conversations.

4- Réinitialisation et état d'accueil

Dans les conversations, nous ne souhaitons pas qu'un protocole d'une conversation puisse revenir à l'état initial du dialogue. Par contre nous souhaitons souvent qu'une conversation puisse revenir à un état donné (état d'accueil) pour des raisons diverses comme par exemple les multiples cycles d'une négociation lors d'une vente aux enchères.

Dans les protocoles d'interaction, la réinitialisation implique qu'une conversation peut être amenée à son état initial ou à un état accessible donné quelconque (état intermédiaire d'un dialogue).

- **Analyse et Validation des Protocoles**

Dans le cadre des protocoles d'interaction, la validation du modèle RDPC consiste à montrer qu'il se comporte correctement. Les méthodes de vérification se traduisent généralement par l'énumération des marquages accessibles. Par conséquent, il est nécessaire d'avoir des outils efficaces pour la construction et l'analyse par exemple des états du protocole modélisé. Lorsque les couleurs et les fonctions présentent une certaine symétrie, il est possible de définir des classes d'équivalences entre les marquages accessibles. L'explosion combinatoire est alors réduite.

Quelques propriétés de base représentant les conditions fortes de la validité des spécifications des protocoles doivent être contrôlées. Ces propriétés sont essentiellement :

- Absence d'inter blocage : les agents impliqués dans une conversation atteignent tous leur état de fin respectif.
- Absence de séquences infinies : tous les agents franchissent leurs transitions finales respectives.
- La vivacité : tous les états sont atteignables à partir de l'état initial et l'utilisation de tous les messages du protocole est assurée.
- Le caractère borné du modèle : le protocole ne peut engendrer un processus avec un nombre infini d'états.

III-2 -5- Modèle RdPC des SMA pour la gestion des ressources renouvelables

Bakam [01] a proposé une démarche de modélisation des systèmes multi-agents par les réseaux de Petri colorés. En effet, il considère un ensemble d'agents (ou d'objets) représentant des ressources naturelles renouvelables. Ces ressources peuvent être la faune, les pâturages, les périmètres irrigués, la forêt naturelle. Les ressources sont exploitées par l'homme. Cette exploitation influence leur dynamique qui est donc fonction de la renouvelabilité et de la pression de prélèvement qui y est exercée.

La figure 7 illustre le modèle multi-agents pour la gestion des ressources naturelles et renouvelables définit par Bakam.

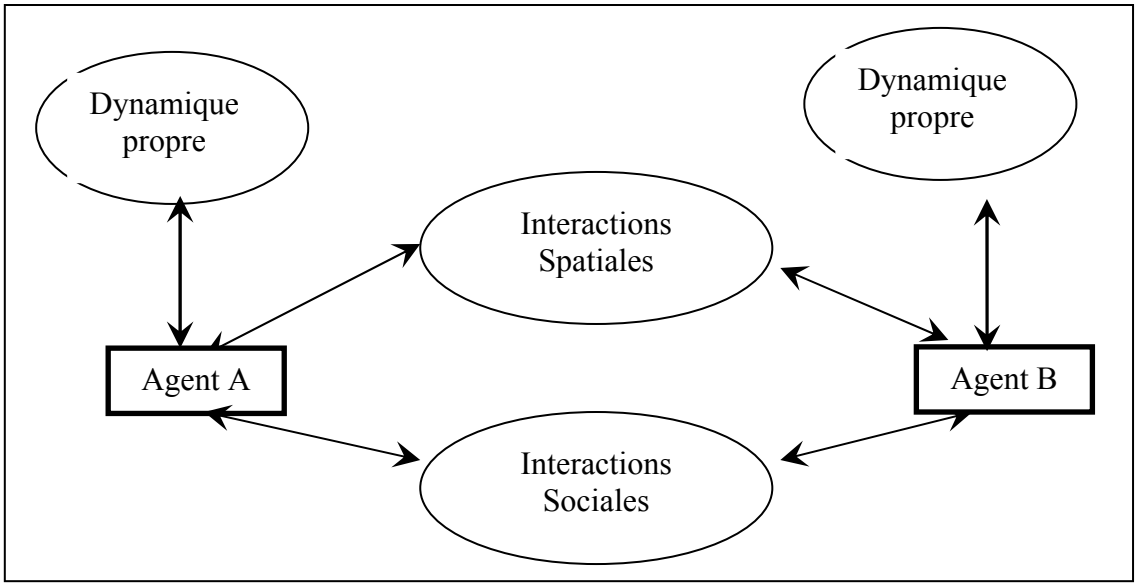


Fig 7: Structure du SMA pour la gestion des ressources renouvelables

Des simulations de ce modèle ont été effectuées sur la plate forme Cormas [01] pour explorer le comportement du système sur de longues durées. Cependant, ces simulations qui correspondent à des évolutions particulières du système ne renseignent pas sur la non existence de situation de blocage par exemple, ou l'extinction de la population que les systèmes de gestion de la faune cherchent à éviter.

En plus, les simulations ne permettent pas de vérifier certaines configurations tells que la viabilité du système, en fonction des conditions initiales et des hypothèses d'évolution [01]. Afin de remédier à ces problèmes et d'étudier le système de manière plus formelle et plus globale, l'utilisation de méthodes formelles de spécification des systèmes multi-agents s'est avérée nécessaire. Bakam a proposé une démarche de modélisation par les réseaux de Petri colorés [01] qui permettent de représenter et d'analyser le comportement des entités autonomes en interaction**.**

- **Démarche de Modélisation par les Réseaux de Petri Colorés**[01]

Les agents sont des jetons qui sont sujets à plusieurs dynamiques : l'évolution interne, les interactions entre agents et les interactions avec l'espace.

L'espace quant à lui est représenté par un objet qui résume dans son champ d'information, la situation spatiale des autres agents ainsi que les relations spatiales qui les lient.

Bakam [01] a proposé dans son approche de modélisation :

- l'utilisation des modules ou des sous réseaux comme des classes d'agents et les jetons comme des agents, cela permet de modéliser l'apparition ou la disparition d'agents en termes de création ou suppression de jetons.

- la modélisation des systèmes multi-agents situés en représentant l'environnement comme un objet particulier du système, particulier par sa structure et ses liens avec les autres objets.

1- Les agents

Un agent est représenté par un jeton dans un sous réseau. C'est-à-dire que le sous réseau correspond à la classe de l'agent, tandis que le jeton dans une place du sous réseau correspond à l'état particulier de cet agent. Les transitions dans le sous-réseau sont les processus de changement d'état de l'agent.

Le sous réseau possède une place particulière dans laquelle le jeton représentant l'agent se trouve chaque fois qu'il doit être l'objet d'une dynamique spatiale. C'est-à-dire une modification de l'information qu'il a sur l'espace et de l'information spatiale globale.

2- Dynamique spatiale

Les transitions qui représentent les processus de dynamique spatiales sont validées par des arcs provenant à la fois de la place qui contient le jeton représentant l'agent situé et de celle qui contient le jeton représentant l'espace. Cette transition est pré conditionnée par :

- la présence dans les places en entrée des marques agent et espace (condition des RdP ordinaires),

- la satisfaction par le couple (agent, espace) des conditions du modèle qui garantissent l'exécution du processus.

3- Action de perception

Le processus de perception prend en entrée l'objet x qui se trouve dans un état exprimant la requête d'une information sur l'espace E. Ce processus retourne l'objet x' qui est le résultat du changement d'état de l'objet x. Ce changement concerne à la fois l'acquisition de l'information sollicitée et le positionnement de l'état à une valeur exprimant la réalisation de l'action de perception. L'objet E représentant l'espace n'est pas modifié lors de la perception

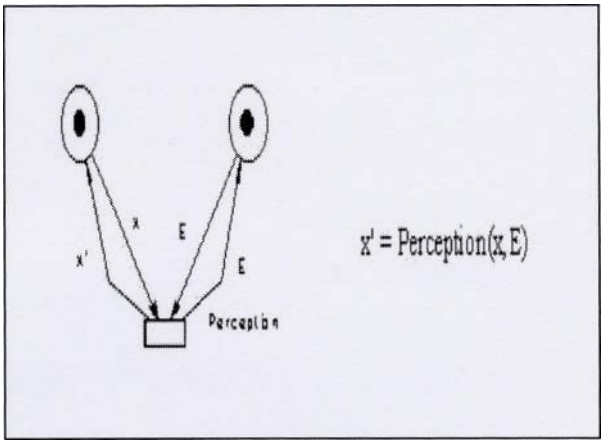


Fig 8 : Modélisation de la perception

4- *Action de modification de l'espace*

Le processus de modification de l'espace prend en compte l'objet qui exerce cette action et l'espace lui-même. Après la modification, le nouvel état de l'espace E' est calculé en fonction de E et de l'objet x. De même l'objet x' est créé comme transformation de x dans le processus (modification de l'intention d'action et résultat de l'action).

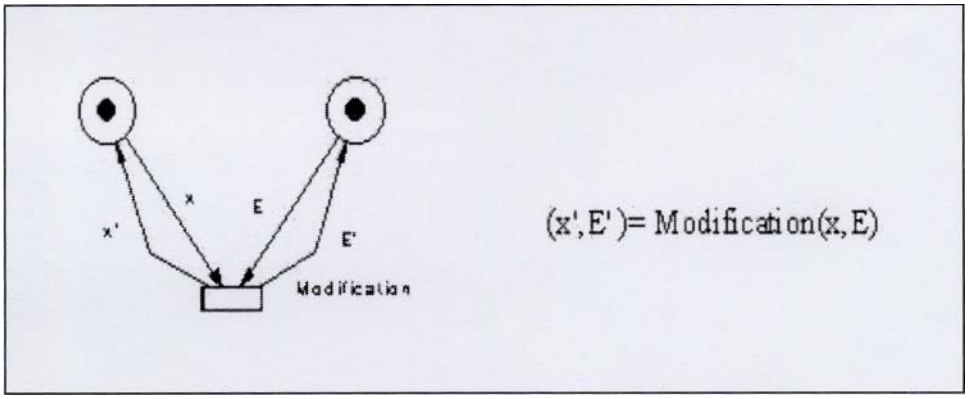


Fig 9 : Modélisation des actions sur l'espace

5- Action de déplacement

Le déplacement consiste à modifier le paramètre de l'agent qui représente sa position dans l'espace, ainsi que l'information sur les positions des agents dans l'objet espace.

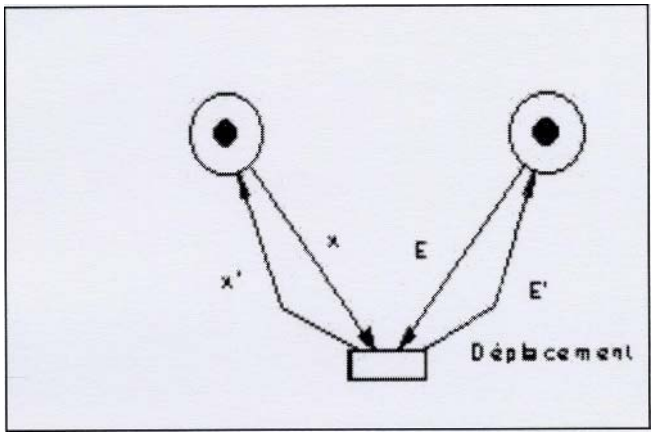
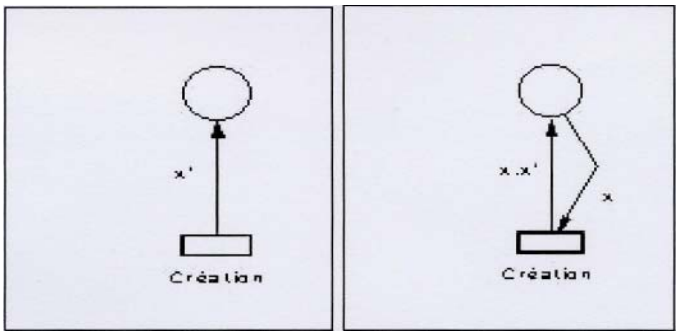


Fig 10 : Modélisation du déplacement

7- Création d'agent

Elle peut être soit un processus indépendant (du système) (fig a), soit la conséquence de l'action d'un agent du système (fig b).



(a) (b)

Fig 11 : Modélisation de la création de nouveaux agents

La situation (a) modélise la création de nouveaux agents dans le système par le modélisateur. Cela fait partie des scénarios dans lesquels un acteur extérieur (ici le modélisateur) veut modifier en cours de simulation certains paramètres du système. Lorsque la création d'un nouvel agent est une conséquence de l'évolution du système, le processus de création est représenté comme sur (b). L'agent (ou l'objet) x est l'initiateur de ce processus et est donc porteur de l'information sur la nature de l'agent à créer. C'est cette information qui est récupérée dans *x* et utilisée par la transition *Création* pour générer le nouvel agent x'. Dans le cas où x devrait être maintenu dans le système, il est retourné en même temps que x'.

- **Application : Modélisation de l'activité de Chasse** [01]

-Description informelle du Système

Le système est composé de chasseurs et d'animaux d'une espèce donnée. Ces animaux se déplacent de façon aléatoire sur un territoire et ils peuvent être attrapés par les pièges que les chasseurs ont pris soin de disposer le long de certaines pistes de chasse. Les animaux se multiplient suivant une fonction de croissance. La disposition des pièges est le résultat d'un processus cognitif des chasseurs que nous ne décrivons pas dans le modèle.

- **Représentation par un Réseau de Petri Coloré**

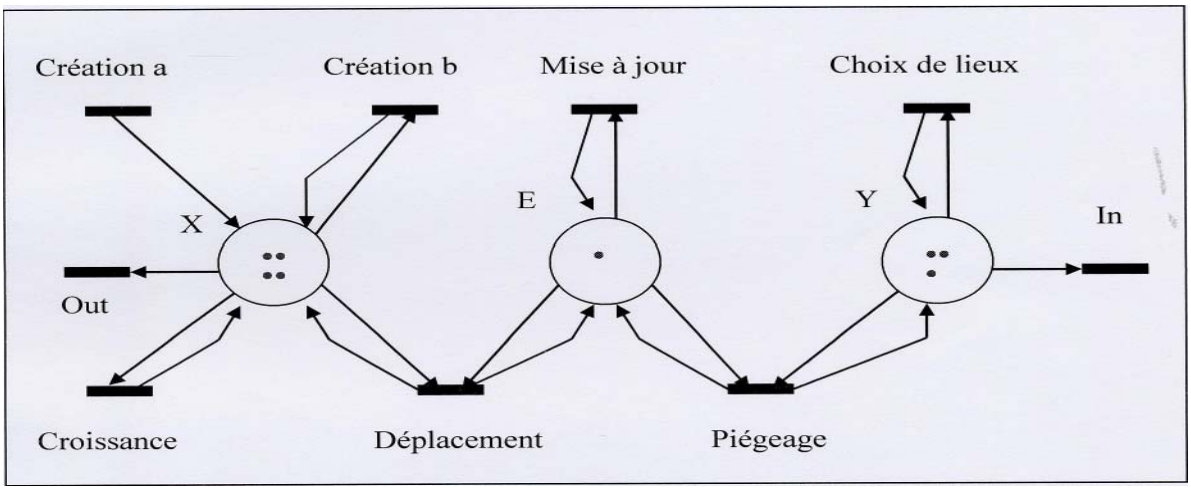


Fig 12 : Modèle de réseau de Petri coloré de la chasse

Les jetons

X = (id, age état, pos).

E = {(pos, elt, état)}.

Y= (id, posPiege, état).

Les transitions

- Croissance (id, age, etat, pos) = (id, age+l, C(etat), pos) où C(etat) est une transformation de l'état par le processus de croissance. Par exemple le passage de #jeune à #vieux ou de #vieux à #mort.
- x' = Déplacement(x,E) = (id, age, D(etat), D(pos)).
- Out(id, age, #mort, pos) = true.
- Mise à jour(E) : Rétablissement de cohérence sur les informations spatiales après certaines modifications.

- In : Création de nouveaux agents dans le système.
- Piégeage : Positionnement des pièges sur l'espace.
- Choix de lieu : Choix des emplacements de pièges.
- Création a : Création de nouveaux agents dans le système par le modélisateur.
- Création b : Création d'un nouvel agent par évolution du système.

➢ **Analyse du Modèle Réseau de Petri Coloré de la chasse**

❖ **Viabilité du système**

Il existe une correspondance entre la viabilité du système et la vivacité (ou la quasi-vivacité) du réseau de Petri. Un réseau est vivant si quel que soit son évolution, une transition donnée est toujours franchissable, cela revient à dire que quel que soit l'évolution du système, on peut toujours aboutir à l'exécution d'une action. On peut donc s'assurer que l'évolution du réseau ne conduit pas à des situations de blocage.

❖ **Faisabilité d'une action**

Dans l'exemple de la modélisation de l'activité de chasse, Il est intéressant de savoir si les conditions d'application d'une règle de gestion sont satisfaites de façon permanente pendant la simulation. Cette propriété, par rapport aux réseaux de Petri, revient à vérifier la vivacité du réseau par rapport à une transition ou un groupe de transitions données.

❖ **Disponibilité des ressources**

Le problème fondamental de la gestion des ressources renouvelables étant l'identification de la stratégie de gestion qui garantit la viabilité de ces ressources (pas de disparition de ressources). Les simulations à elles seules ne permettent pas de vérifier cette propriété [01].

La représentation par les RdP permet d'aborder cette question. Les ressources étant représentées par des marques dans une place du réseau, il faudra démontrer la non-existence d'un marquage pour lequel cette place est vide afin de conclure sans simulation que la ressource sera préservée quel que soit l'évolution du système.

III-2-6- Un Réseau de Petri coloré pour une application Multi-Agents

Danny Weyns [58] a présenté la modélisation des applications multi-agents par un réseau de Petri Coloré. Le but de ses recherches a été d'obtenir une meilleure compréhension de sociabilité dans les systèmes multi-agents, afin de développer graduellement un modèle conceptuel générique pour les agents sociaux situés dans un SMA. En particulier il a utilisé packet- world pour l'étude du comportement social des agents. Son approche est basée sur la combinaison des expériences avec la modélisation conceptuelle.

Premièrement Weyns a défini un modèle de base, puis il ajoute des nouvelles qualifications sociales d'une manière modulaire, afin d'avoir une vue conceptuelle claire sur l'évolution des agents et de l'environnement. Une vue conceptuelle définit les concepts dont ont besoin les agents afin d'acquérir de nouvelles capacités sociales, l'infrastructure nécessaire dans l'environnement pour soutenir les capacités des agents et comment ses concepts se relient entre eux. Weyns présente d'abord un réseau de Petri coloré pour un modèle de base du paquet-world. Ce modèle se compose des agents qui peuvent seulement agir l'un sur l'autre par des objets passifs dans l'environnement. L'interaction et la coordination entre les agents et l'environnement sont assurées par une Infrastructure de base incorporée dans le modèle de base.

Les réseaux de Petri sont très utilisés pour décrire et analyser des processus concurrents. En plus de leurs représentations graphiques intuitives. Ce sont des candidats potentiels pour la modélisation conceptuelle des systèmes complexes. Le comportement d'un système modélisé avec un réseau de Petri coloré peut être analysé, non seulement au moyen d'une simulation mais également sur une base formelle. Il est remarquable que les réseaux de Petri coloré offrent la plupart des ingrédients pour aborder la complexité des systèmes multi-agents.

Le Packet-World

Soit une grille carrée de taille ''S'', contenant un nombre de paquets colorés et des agents. Le rôle d'un agent est de collecter les paquets et les mettre dans leur destination. La grille contient une destination pour chaque couleur. La figure 14 illustre un exemple de Packet-world, de taille 8 avec trois agents. Dans le Packet-world, les agents peuvent interagir avec l'environnement suivant plusieurs chemins : Un agent peut faire un pas vers une des cases voisines libres qui l'entourent. Si un agent ne porte aucun paquet, il peut ramasser un des paquets à partir de l'une des cases qui l'entoure.

- Un agent peut déposer le paquet sur l'une des cases libres ou sur la destination de ce paquet.
- Un agent doit attendre lorsqu'il n'a rien à faire. Il est important de noter que l'agent n'a qu'une vue partielle de son environnement et qu'il n'a aucune interaction avec les autres agents mais seulement avec son environnement. L'agent doit avoir des informations sur lui même pour qu'il puisse agir. Dans ce modèle l'agent a des connaissances concernant sa position dans la grille et la possession ou non d'un paquet.

Ces informations seront représentées dans cette modélisation par des marques de type ‘‘Views’’, il peut avoir aussi des informations sur l’environnement qui ont été collectées dans le passé, et qui peuvent être vraies ou pas, ces informations sont appelées les croyances et on les modélise par des marques de type ‘‘Belief’’.

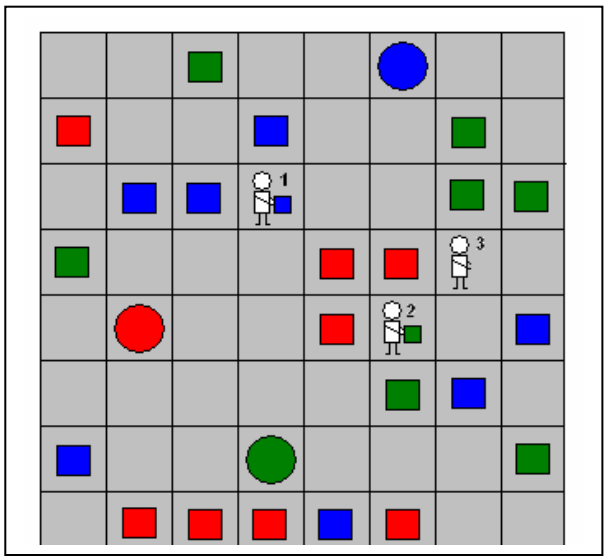


Fig 13: Packet-world

➢ **Le Réseau de Petri associé** [59]

La figure 14 représente la modélisation du système ‘‘Packet-world’’ par un réseau de Petri coloré où l’ensemble des transitions et l’ensemble des places sont décrits comme suit :

- Look for P : Cette place contient des jetons de type Agent, elle permet la recherche d’un paquet.
- Look for D : Cette place contient aussi des jetons de type Agent et elle permet la recherche d’une destination.
- Ready : initialement cette place contient un jeton de type Agent. Elle permet le commencement de l’exécution par le franchissement de la transition ‘‘Start up’’ qui fait passer le jeton ‘‘Agent ’’ aux places ‘‘Look for P’’et ‘‘Identify’’.
- Les actions de l’agent sont modélisées par des transitions. Chaque transition consomme au moins un jeton de type ‘‘Agent ’’ et un jeton de type ‘‘Views’’. Si le jeton ‘‘ Agent’’ se trouve dans la place ‘‘Look for P’’, une des transitions ‘‘Step P’’, ‘‘Pick’’ et ‘‘Skip P’’ peut être franchie ; et s’il se trouve dans la place ‘‘Look for D’’ une des transitions ‘‘Step D’’, ‘‘Put’’ et ‘‘Skip D’’ peut être franchie.
- Les informations concernant l’environnement sont représentées par la place ‘‘Percept’’. Lorsqu’une nouvelle information arrive, l’agent doit s’identifier pour l’avoir. Cette action est représentée par la transition ‘‘Update View’’.
- La place ‘‘Consume packet’’ représente les paquets pris par les agents.
- La place ‘‘Consume position’’ représente les positions occupées par les agents.
- Si un agent a l’intention de faire un pas, il doit mettre un jeton de type ‘‘Move’’ dans la place ‘‘Perform Step’’. Ce jeton contient l’identité de l’agent et les coordonnées de la case désirée. De même l’agent envoie ce jeton à la place

‘‘Perform Put’’ lorsqu’il veut déposer un paquet, à la place ‘‘Perform Pick’’ lorsqu’il veut ramasser un paquet et à la place ‘‘Perform Skip’’ lorsqu’il veut sauter à une case.

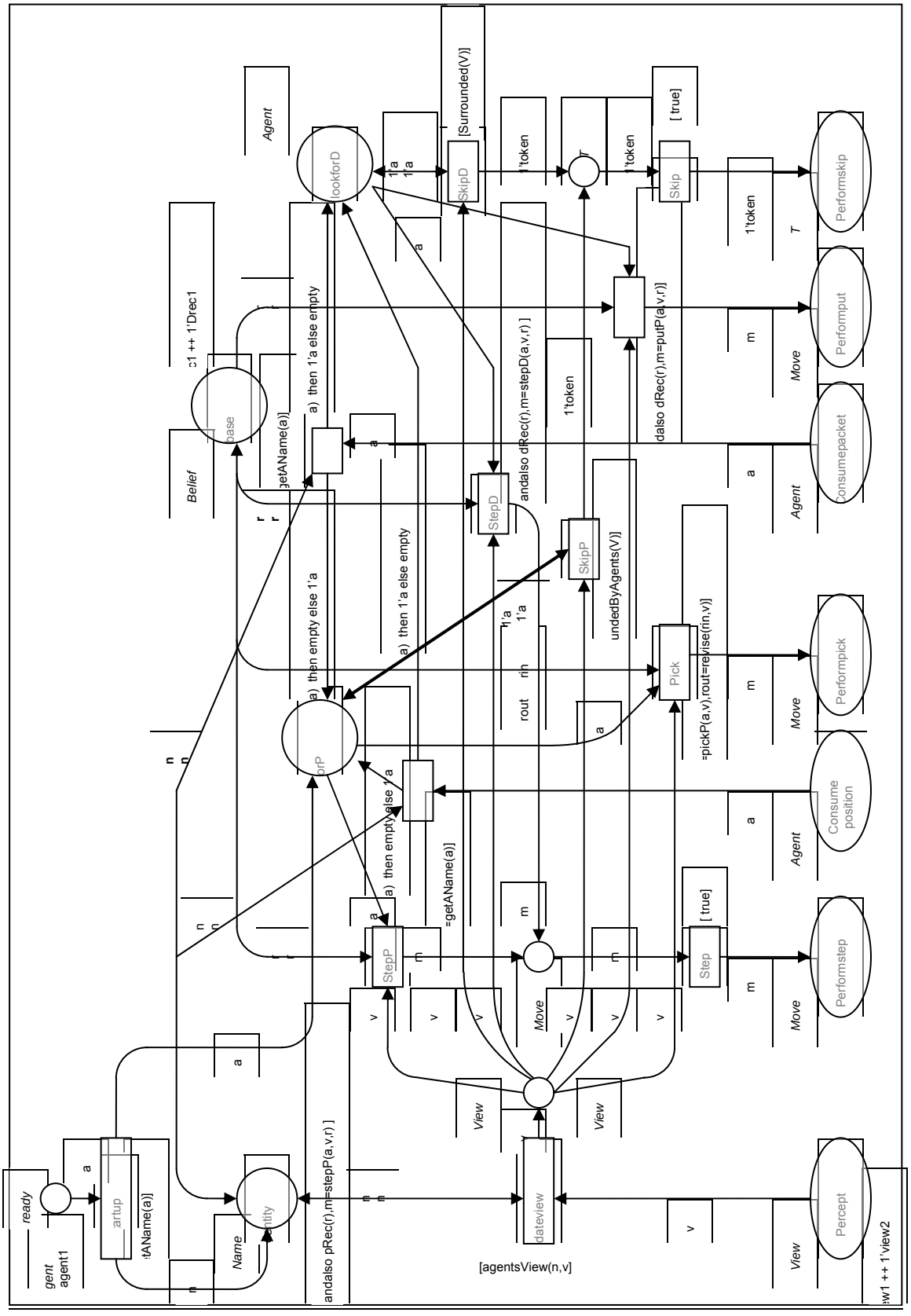


Fig 14 : Le RdP associé au packet world

III-3- Conclusion

L'intérêt des systèmes multi-agents est reconnu depuis longtemps. Cependant, leur utilisation se heurte à quatre obstacles majeurs : leurs notions de base ne sont pas caractérisées de façon claire, les méthodologies de leurs développement sont à l'état artisanal, les environnements permettant d'assister les développeurs sont pratiquement inexistants et l'utilisation des méthodes et des outils de vérification et validation nécessite une mise en œuvre fine et raffinée.

C'est pour ses raisons que beaucoup de chercheurs ont investi dans le domaine des SMA, plus particulièrement l'utilisation des réseaux de Petri comme outil de formalisation et modélisation.

Ainsi le formalisme Bric [45] et le formalisme Val [57] ont été introduits pour la formalisation des modèles à agents cognitifs et communicants. El Fallah s'est intéressée à la représentation et manipulation des plans à l'aide des réseaux de Petri, et ce dans le contexte de la résolution distribuée des problèmes par les agents. Par contre Weyns a focalisé ses recherches dans l'étude, la cohérence et la vérification des comportements sociaux des agents.

Mazouzi [47] s'est intéressé aux comportements d'agents en interaction, en introduisant un modèle réseaux de Petri colorés pour la formalisation des protocoles d'interaction.

Quant au modèle proposé par Bakam [01], il permet d'analyser et de valider les résultats obtenus par simulation des systèmes multi-agents dédiés à la gestion des ressources renouvelables.

D'après les différents modèles que nous avons présentés dans ce chapitre, chaque chercheur a utilisé les réseaux de Petri pour formaliser un aspect des SMA. Ainsi on retrouve le formalisme Bric et Val pour les modèles d'agents cognitifs et communicants, le modèle de représentation de plan de résolution par des agents, la modélisation des conversations entre des agents communicants, la simulation des systèmes multi-agents pour des agents situés et la modélisation d'une application le Packet –World.

Dans le Chapitre suivant, nous présentons une modélisation des systèmes multi-agents à l'aide des réseaux de Petri. Cette modélisation engendrera les principaux aspects des SMA à savoir la définition d'une architecture d'agents, les interactions à travers les différentes formes (coopération, coordination et négociation).

Chapitre IV
La Modélisation SMA

IV -1- Introduction

Le principal intérêt des SMA est qu'ils proposent de modéliser des systèmes complexes à l'aide d'une société d'entités appelées agents. Chaque agent a ses propres compétences, mais il a besoin d'interagir avec les autres pour résoudre les problèmes qui dépendent de son domaine de compétence et éviter les conflits. L'objectif de ces systèmes est donc de trouver une solution à des problèmes globaux ou de simuler des comportements complexes à l'aide d'un ensemble d'agents. Donc les systèmes multi-agents permettent de distribuer des agents, entités communicantes, autonomes, réactives et dotées de compétences.

Pour réaliser un SMA selon ces critères, il faut doter chaque agent, des trois propriétés suivantes : indépendance, communication et intelligence. Pour modéliser de tels agents, il nous faut définir leur architecture (fonctions internes et interactions cognitives), ainsi que la structuration des connaissances nécessaires pour leurs différentes activités.

Un système multi-agent se distingue d'une collection d'agents indépendants par le fait que les agents interagissent en vue de réaliser conjointement une tâche ou d'atteindre simultanément un but particulier. Les agents peuvent interagir en communiquant directement entre eux ou par l'intermédiaire d'un autre agent ou même en agissant sur leur environnement. Chaque agent peut être caractérisé par trois dimensions : **ses buts**, **ses capacités** à réaliser certaines tâches et **les ressources** dont il dispose. Les interactions des agents d'un SMA sont motivées par l'interdépendance des agents selon ces trois dimensions : leurs buts peuvent être compatibles ou non; les agents peuvent désirer des ressources que les autres possèdent; un agent peut disposer d'une capacité nécessaire à un autre agent pour l'accomplissement d'un de ses plans d'action.

Dans ce chapitre nous proposons une description d'agent et de ses systèmes, qui comporte notamment une description formelle des éléments de l'architecture des agents cognitifs et réactifs sous forme de réseaux de Petri, en plus une formalisation des différentes formes d'interactions à savoir la coopération, la coordination d'actions et la négociation.

IV -2- Modélisation d'un Agent

Pour la définition de l'architecture d'un agent, nous nous somme inspirés d'une part du schéma de modélisation proposé par Ermine dans [22]. Il reprend le schéma classique OID (Opérations, Information, Décision) [42] auquel il intègre un quatrième paramètre permettant la circulation des connaissances, Et d'autre par de l'approche proposée par Fougéres dans [30], où il définit une structure des agents qu'il a définit permet de représenter sur un modèle unique, à la fois l'activité et le comportement décisionnel des agents cognitifs.

L'approche que nous proposons permet de représenter un modèle qui décrit à la fois :

1- l'activité cognitive des agents: *[Perception, Interprétation/Décision, raisonnement,]* dont le processus de raisonnement est orienté vers la prise en compte de ses états mentaux, ses capacités cognitives et la disponibilités des ressources.

2- l'activité réactive des agents *[Action/Réaction]*, dont la prise de décision se base sur des données brutes en provenance des senseurs.

En plus la démarche proposée met en exergue les différents aspects qu'entour les agents a savoir :

- Ce que l'agent connaît et perçoit de son environnement,
- Les états possibles envers lesquels l'agent peut vouloir s'engager;
- Les états envers lesquels l'agent s'est engagé, et envers lesquels il a engagé des ressources,
- Les capacités, et les compétences de l'agent pour résoudre un problème ou un conflit,
- Gérer, emmener, résoudre les problèmes des différentes formes d'interactions (la coopération, la coordination et la négociation).

IV -2-1- Architecture d'un Agent

L'architecture générale d'un agent cognitif et réactif que nous avons proposé repose sur un réseau de Petri coloré (Figure 15). Cette architecture respecte les quatre propriétés, à savoir l'indépendance, la réactivité, la communication et l'intelligence. Celle-ci inspirée de la théorie de modularité est composée de cinq modules gérant les connaissances, la perception, la reconnaissance de la situation, la communication, le raisonnement et la décision de l'agent.

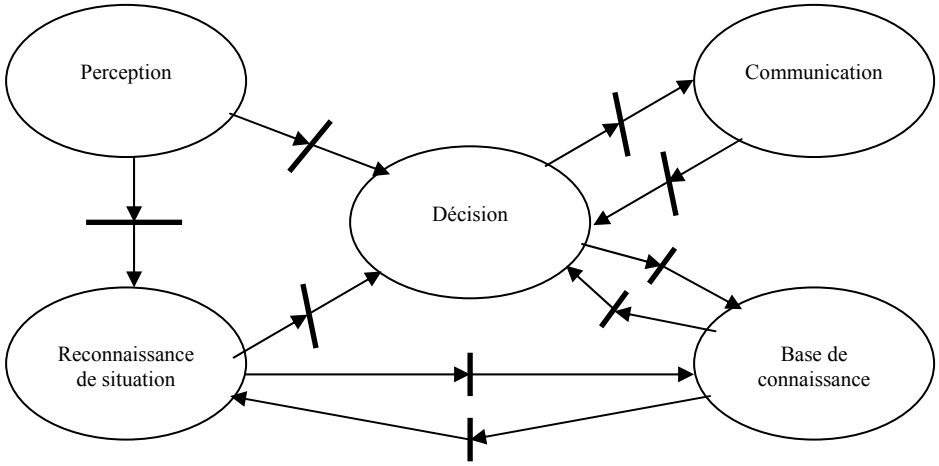


Fig 15: Architecture modulaire d'un agent

Lorsqu'un agent perçoit une situation dans l'environnement, il essaie de la reconnaître. Si la situation lui est familière, il peut enclencher un processus de planification, de coordination ou de résolution de conflit afin de résoudre le problème. Il peut aussi reconnaître la situation en terme d'action et donc, passe à l'exécution de la tâche. Lorsque l'agent perçoit des situations qu'il connaît très bien, il peut faire intervenir son comportement réactif en passant directement à l'action (Perception Exécution). Si l'agent ne peut pas résoudre un problème (compétences insuffisantes), il engage un processus de coopération pour demander de l'aide aux autres agents.

Si l'agent ne reconnaît pas la situation perçue, il engage un comportement cognitif, pour déterminer la façon d'agir et les éléments nécessaires pour la prise d'une décision rationnelle. L'agent doit entreprendre son action en fonction des connaissances et des ressources dont il dispose, tout en conservant son intégrité, l'organisation dont il fait partie et ses buts. Son action sera le fruit d'une délibération raisonnée et qui sert directement à satisfaire ses buts.

IV -2-2- Modélisation des modules

L'architecture des agents que nous avons proposés repose principalement sur trois modules à savoir : le module communication, le module perception et le module décision. Pour l'analyse et la vérification du fonctionnement et du comportement des modules communication et perception, nous proposons dans la section suivante une modélisation de ces derniers par les réseaux de Petri colorés.

IV -2-2- 1- Le module de communication

Les communications, dans les systèmes multi-agents sont à la base des interactions et de l'organisation sociale. Pour communiquer, les agents expriment leurs intentions selon un langage de communication. La forme générale du message peut avoir une apparence de la forme (demander, affirmer, accepter, refuser, etc). Dans notre modélisation le processus de communication est initié à la réception d'un message qui sera stocké dans une file d'attente en vue d'un traitement. L'agent lit le message, l'interprète, le décode et le traite. Lorsque l'agent formule le besoin d'un envoi de message, une demande est effectuée, puis le message est créé selon un lagunage de communication, et transmis au mécanisme de synchronisation en vue d'une transmission.

- **Modélisation du module communication par un réseau de Petri coloré**

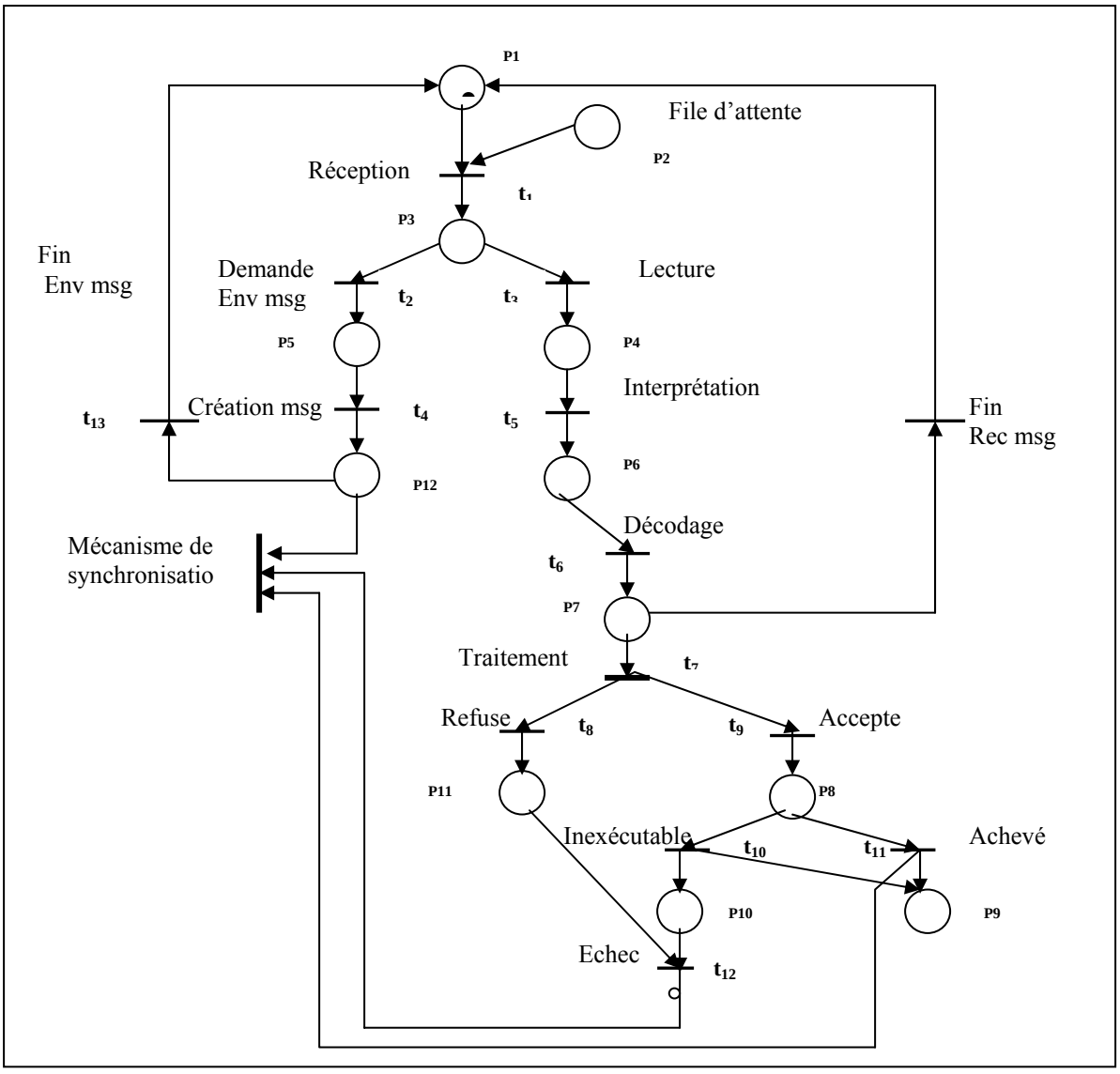


Fig 16 : Modélisation par RdPC du module communication

Interprétation

Le processus se déclenche à la réception d'un message émanant d'un autre agent. Le récepteur consulte sa file d'attente et vérifie son existante et son ordre Après le franchissement de la transition t_1 correspondant à la réception, le message sera lu, interprété et décodé (franchissement des transitions t_3, t_5, t_6). Le franchissement de la transition t_7 correspond au traitement du message. A cette étape deux possibilités sont envisagées :

1- Soit le message est accepté, et il sera exécuté, l'opération est achevée (franchissement de la transition t_{11}) et l'agent récepteur sera dans un état de fin de communication.

2- Soit le message est refusé (franchissement de la transition t_8), l'agent passe à l'état échec (franchissement de la transition t_{12}). Un message d'échec sera envoyé à l'émetteur via le mécanisme de synchronisation.

Lorsque l'agent émet le vœu d'envoyer un message à un autre agent, une demande d'envoi de message sera envoyée au module communication, le message sera crée (franchissement de la transition t_4) puis envoyé au mécanisme de synchronisation, et le message de fin d'envoi sera envoyé l'agent émetteur (franchissement de la transition t_{13}).

IV -2-2-2- Module de Perception

Un agent peut avoir une perception locale ou globale de son environnement, avec une possibilité d'intégrer dans ses connaissances de nouvelles informations en fonction de ses besoins. Généralement la perception s'effectue à travers les capteurs dont disposent les agents. Ses derniers peuvent apercevoir et acquérir de nouvelles connaissances en agissant ou en se déplaçant dans leur environnement. Deux types de perception peuvent être distinguées :

- La perception passive où l'agent reçoit des données sans avoir à réaliser d'action pour les obtenir.

- La perception active qui résulte d'une action volontaire. Dans ce cas l'agent va à la recherche d'une information. Il peut s'agir d'une détection de son environnement physique via des données issues des capteurs, de l'envoi d'une demande d'information ou d'identification en direction des autre agents, ou simplement de l'évaluation de la modification d'une variable ou du contenu d'un fichier.

.

- **Modélisation du module communication par un réseau de Petri coloré**

La capacité perspective d'un agent est utilisée afin de représenter l'état de son environnement, c'est à dire mettre à jour sa représentation interne de son environnement. Dans l'architecture de l'agent que nous avons définie, le module qui met en œuvre cette capacité est le module de Perception. Afin de déterminer son fonctionnement nous présentons une modélisation de ce dernier par un réseau de Petri coloré (figure 17).

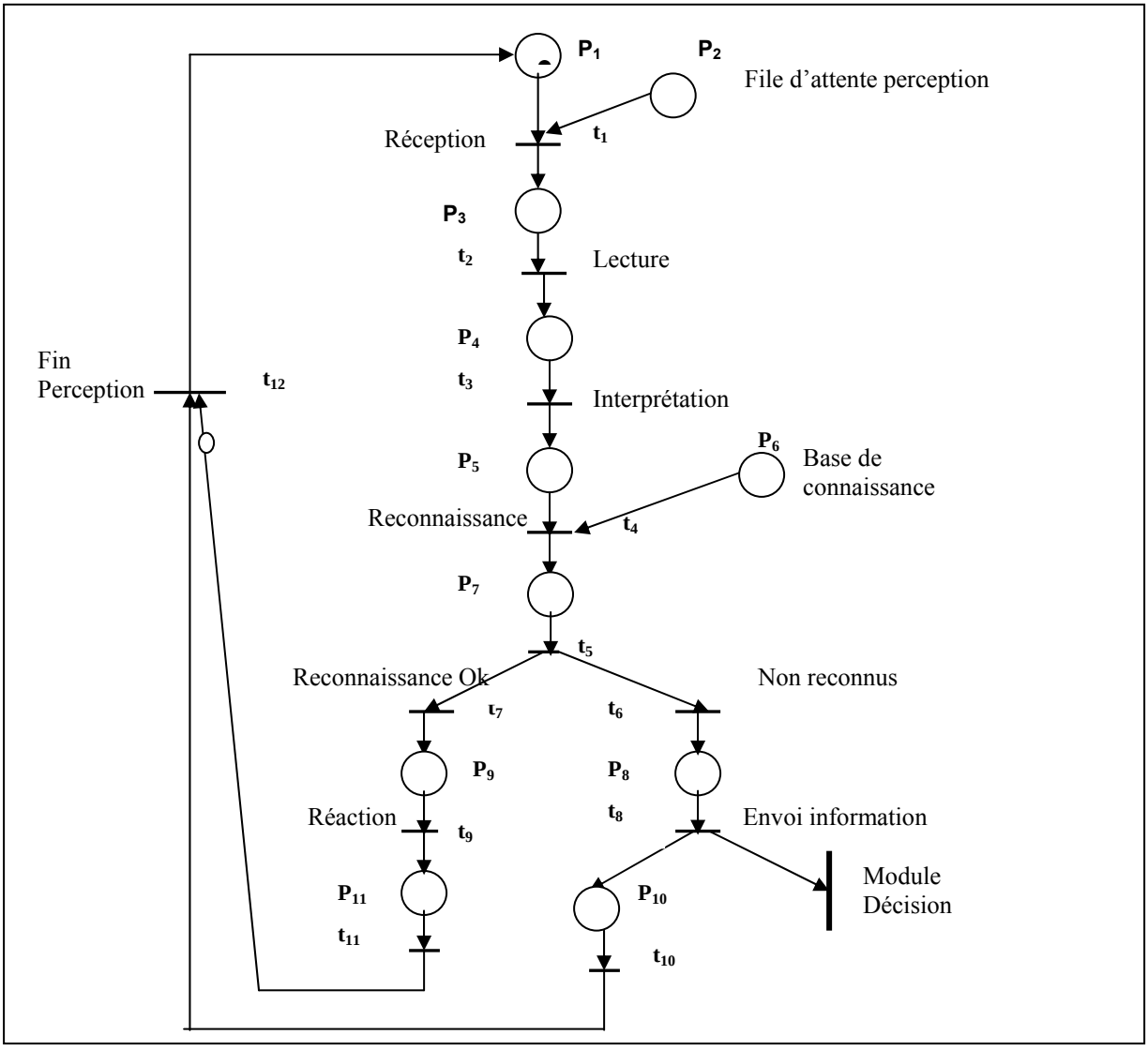


Fig17 : Modélisation par RdPC du module Perception

Interprétation

Le processus se déclenche à la réception d'une perception de l'environnement, l'agent consulte sa file d'attente et vérifie si elle existe. Après le franchissement de la transition t_1 correspondant à la réception, la perception sera lue, interprétée (franchissement des transitions t_3, t_2,). L'agent consulte sa base de connaissances et essaye de reconnaître ce qu'il a perçu (franchissement de la transition t_4.). A cette étape deux possibilités sont envisagées :

1- L'agent reconnaît la perception et il engage le processus d'un agent réactif (réaction automatique en conséquence).

2- L'agent ne reconnaît pas la perception, il envoie les informations perçues au module décision en vue d'une exploitation (franchissement de la transition t_8) et un message de fin de perception est envoyé (franchissement de la transition t_{12}).

IV -3- Modélisation des Interactions

Problématique des interactions

Une des principales propriétés de l'agent dans un système multi-agents est celle d'interagir avec les autres. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein de la société qui a pour effet de modifier le comportement d'un autre agent. Elles donnent aux agents la possibilité de participer à la satisfaction d'un but global. Cette participation permet au système d'évoluer vers un de ses objectifs et d'avoir un comportement intelligent quel que soit le degré de complexité des agents qui le composent.

En général, les interactions sont mises en oeuvre par un transfert d'informations directement entre agents ou via l'environnement; soit par perception ou par communication. Par la perception, les agents ont connaissance d'un changement de comportement d'un tiers au traver de l'environnement. Par la communication, un agent fait un acte délibéré de transfert d'informations vers un ou plusieurs autres agents. L'interaction peut être décomposée en trois phases non nécessairement séquentielles [08] :

- la réception d'informations ou la perception d'un changement,
- le raisonnement sur les autres agents,
- une émission de message(s) ou plusieurs actions (plan d'actions) modifiant l'environnement. Cette phase est le résultat d'un raisonnement de l'agent sur son propre savoir-faire et celui des autres agents.

Au sein des systèmes multi-agents, la communication est souvent l'un des moyens utilisés pour échanger des informations entre agents (plans, résultats partiels, buts, etc). La capacité de communiquer et, par conséquent, de coopérer des agents, s'appuie sur un fond commun d'aptitudes aussi complexes que la perception, l'apprentissage, la planification et le raisonnement. On retrouve ainsi dans la résolution de problèmes, des situations d'interaction et des stratégies coopératives non déterministes, difficiles à interpréter et parfois non complètement reproductibles. L'équilibre entre l'autonomie des agents, dotés d'un comportement plus ou moins intelligent, et la convergence vers un objectif global caractérise l'interaction.

L'interaction constitue donc un niveau d'abstraction supérieur à la notion de communication et d'action. Le problème est alors de savoir combiner ces éléments (communications et actions) afin de coordonner et de contrôler les échanges entre plusieurs agents pour avoir un comportement collectif cohérent du système.

Plusieurs types d'interaction ont été définis et analysés à travers diverses composantes [26]. Ferber donne une classification des situations d'interaction abordées selon un point de vue d'un observateur extérieur. Cette classification représente les différentes situations d'interaction que l'on retrouve en fonction des objectifs des agents (compatibles ou incompatibles), des ressources dont ils disposent (suffisantes ou insuffisantes) et de leurs compétences pour la résolution d'un problème.

Ferber aboutit ainsi à une typologie des situations d'interactions : indépendance, collaboration simple, encombrement, collaboration coordonnée, compétition individuelle pure, compétition collective pure, conflit individuel, conflits collectifs.

Dans le chapitre 1 nous avons mis en évidence, l'influence de trois critères (objectifs, compétences, ressources) sur la situation d'interaction induite. En l'occurrence, la compatibilité des objectifs induira des situations de coopération dès que les ressources ou les compétences sont insuffisantes.

De cette typologie des situations d'interaction découlent plusieurs formes d'interaction, cela peut aller de la collaboration qui consiste à établir qui fait quoi et avec quels moyens, et de quelle manière, cela sous entend qu'il faut trouver des solutions aux différents problèmes que constitue la collaboration par répartition des tâches, la coordination, jusqu' à la résolution des conflits qui supposent une négociation pour lever ces conflits. Dans cette partie, nous proposons des modélisations par les réseaux de Petri colorés des différentes formes d'interactions, d'une part entre les agents coopératifs qui peut se traduire par l'allocation des tâches et la coordination d'actions et d'autre part entre les agents antagonistes qui se traduit par la négociation.

IV -3-1- Les Interactions entre Agents Coopératifs

Au fur et à mesure que croît l'intérêt pour les applications composées d'agents s'entraidant pour l'atteinte d'un but commun, et que de plus en plus d'agents sont construits dans le but de coopérer avec d'autres agents, il devient important de bien comprendre les principes qui sous-tendent la coopération.

Comme nous l'avons vu dans le chapitre 1, d'une part la coopération peut être sous la forme d'une collaboration simple. Dans le cas des agents ayant des buts communs, des ressources suffisantes et des compétences insuffisantes, cette forme de collaboration consiste en une simple addition de compétence ne nécessitant pas d'action supplémentaire de coordination entre les agents. Cette situation s'exprime sous la forme d'allocation de tâches et partage de connaissances. A travers cette forme d'interaction, les agents se font partager des connaissances pour aboutir à la résolution d'un problème.

D'autre part, la coopération peut se présenter sous la forme d'une coordination d'actions dans le cas des agents ayant des buts communs, des ressources insuffisantes et des compétences insuffisantes. La coordination des agents consiste à modéliser explicitement le travail d'équipe qu'effectuent les agents. Cette approche s'avère très pratique dans des environnements dynamiques où les membres de l'équipe peuvent échouer dans leurs tâches. Dans de telles situations, l'équipe doit pouvoir évaluer ses performances et être en mesure de se réorganiser en conséquence. Il s'agit d'une collaboration complexe qui suppose que les agents doivent coordonner leurs actions afin de disposer de la synergie de l'ensemble de leurs compétences. C'est la plus complexe des situations de coopération puisqu' elle ajoute aux problèmes de l'allocation des tâches, des aspects de coordination dûs aux ressources limitées.

- **L'allocation des tâches**

La coopération permet de procurer des avantages en termes d'efficacité quantitative et émergence qualitative comme nous l'avons vu dans le chapitre 1. Néanmoins, elle pose des problèmes de répartition du travail entre les agents composant le système. Ces problèmes ne sont pas faciles à résoudre ; car ils font intervenir de nombreux paramètres tels que la capacité cognitive et d'engagement des agents, la compétence des individus, la nature des tâches, le coût de transmission. De ce fait, la répartition des tâches et des ressources constitue l'un des domaines majeur des systèmes multi-agents. Répartir des tâches, des informations et des ressources revient à répondre à la question : qui doit faire quoi et avec quels moyens, en fonction des buts et des compétences des agents et des contraintes contextuelles (types et quantités de ressources, nature de l'environnements, etc.).

Dans ce paragraphe, nous présentons une synthèse des modèles illustrant les différents modes d'allocation de tâches entre les agents et nous présenterons une modélisation par les réseaux de Petri colorés de la forme d'allocation de tâche

distribuée qui présente de nombreux avantages par rapport aux autres formes comme nous le verrons dans la section suivante.

Modes d'allocation des tâches

L'allocation des tâches passe par la définition des mécanismes par lesquels les agents peuvent mettre leurs compétences en commun afin de réaliser un travail collectif. Il s'agit donc de décrire la manière d'allouer les tâches. Les tâches qui réclament plus de moyens, de travail ou de savoir-faire, qu'un seul agent n'est capable de fournir et d'accomplir, doivent être décomposées d'abord en plusieurs sous-tâches puis être réparties parmi les agents que constitue le système multi-agents. Ces deux opérations sont évidemment liées car d'une part, la décomposition des tâches doit souvent prendre en compte les compétences des agents, ce qui engendre une répartition facile et cohérente [26].

La gestion de la répartition des tâches peut s'effectuer, soit en centralisant le processus d'allocation, soit en le distribuant à l'ensemble des agents concernés.

Dans un mode d'allocation, centralisé la répartition passe par la définition d'agents spéciaux, les médiateurs, qui gèrent l'ensemble du processus d'allocation en centralisant les demandes des clients et les offres de services des fournisseurs, afin de mettre en correspondance ces deux catégories d'agents.

L'inconvénient majeur de ce mode est qu'il constitue un goulet d'étranglement, par conséquent, il diminue considérablement les performances du système, dès que le nombre des agents et des demandes augmente. En effet, le nombre de messages que le médiateur doit gérer croît comme le carré de N, où N est le nombre d'agents. En plus, un système centralisé est très sensible aux défaillances. En effet, si le médiateur, tombe en panne, tout le système est en état de disfonctionnement d'où la nécessité d'un mécanisme de secours.

Lorsque l'on souhaite que le système puisse répartir ses mécanismes d'allocation de tâches, il est préférable de passer à un mode d'allocation des tâches distribuées.

Dans un mode d'allocation distribué, chaque agent s'occupe individuellement d'obtenir les services des fournisseurs qui peuvent lui être utiles pour la réalisation de ses projets. On distingue deux mécanismes d'allocation distribuée :

Le mode d'*allocation par réseau d'accointances,* où les agents possèdent une représentation des compétences des autres agents et des capacités dont ils disposent. Il n'est pas nécessaire que les agents connaissent les capacités de tous les autres agents pour pouvoir résoudre leur problème, mais seulement que le réseau formé par l'ensemble des accointances soit fortement connexe (il existe un chemin allant de n'importe quel agent vers n'importe quel autre agent) est évidemment cohérent (les représentations sur les compétences des accointances d'un agent correspondent bien aux compétences effectives des agents).

Dans ce cas, d'une part, l'allocation des tâches peut être directe entre des agents ayant des liens et des relations directes. L'agent qui veut accomplir une tâche essaye à tour de rôle chacun des agents qu'il connaît et dispose de la compétence désirée jusqu'a ce que l'un d'eux accepte. Si aucun agent n'accepte, on atteint un état spécial qu'il faudra traiter. Soit on considère qu'il s'agit d'un défaut du système, soit on prend des mesures tendant surmonter cet obstacle en affectant autoritairement une tâche a un agent ou en relançant les opérations par un mécanisme d'appel d'offre permettant de choisir l'agent qui a le moins de raisons de refuser cette tâche.

D'autre part, l'allocation des tâches peut être par délégation. Cette technique permet de relier des fournisseurs qui ne se connaissent pas directement. Si un fournisseur n'est pas capable d'effectuer une tâche, il peut la renvoyer à un autre agent. L'avantage du mode d'allocation distribué réside dans sa simplicité, il suppose simplement que les clients ont la capacité de connaître suffisamment de spécialistes pour que les tâches qu'ils ne peuvent pas effectuer eux même puissent être réalisées par d'autres. Cependant, ce mode s'avère beaucoup trop limité, car il ne peut mettre en contact des fournisseurs qui ne seraient pas en contact direct avec les clients. Il est donc nécessaire d'introduire un mécanisme qui permette à plusieurs agents de réaliser un contact indirect.

Dans ce contexte, nous avons proposé une modélisation par réseaux de Petri colorés (figure 18) qui prend en charge à la fois l'allocation des tâches distribuée directe, et qui remédie au problème cité précédemment (lorsque aucun agent n'accepte d'exécuter la tâche).

Le mode *d'allocation par appel d'offre* est plus connu en intelligence artificielle distribuée sous le nom de *réseau contractuel.* Il présente l'avantage d'une très grande dynamicité et s'avère particulièrement facile à mettre en œuvre.

Ces modes s'appliquent à des systèmes d'agents cognitifs.

Il existe une autre forme d'allocation moins connue généralement et plus caractéristique des systèmes réactifs dans lesquels les communications s'effectuent par des propagations de stimuli. Ce point ne sera pas détaillé dans ce travail.

A- Modélisation de l'allocation des tâches par RdPC

Dans les systèmes d'allocation par réseau d'accointances, on suppose que chaque agent dispose d'une table de compétences des agents qu'il connaît, sous la forme : $\{C_1: [A_{11}, \ldots, A_{1k}], \ldots, C_n :[A_{n1}, \ldots, A_{nl}]\}$; où les C_i sont les compétences requises pour réaliser une tâche T_i et les A_{ij} sont les agents qui savent effectuer ce type de tâches.

Dans le modèle d'allocation des tâches distribué que nous proposons, un agent peut faire exécuter une tâche qu'à un agent qu'il connaît directement. Le mécanisme de répartition est très simple. L'agent qui veut faire accomplir une tâche essaie à tour de rôle chacun des agents qu'il connaît et qui dispose de la compétence désirée jusqu'à ce que l'un d'eux accepte.

Les messages internes que le client échange avec les autres agents sont définis par :

- **Allouer (T)** est une demande vers le module d'allocation pour qu'il trouve quelqu'un pour faire la tâche T.
- **Impossible All (T)** est une réponse du gestionnaire d'allocation qui indique qu'il n'existe aucun agent qui accepte de faire la tâche T.
- **Engagé faire (Y, T)** est une réponse du module d'allocation indiquant que l'agent Y s'est engagé à faire la tâche T.

Les messages externes sont définis par :

- **Demander (T, self)** est une demande de l'agent à un fournisseur que la tâche T soit réalisée.
- **Accepter (T, Y)** est la réponse positive du fournisseur Y à la demande du client.
- **Refuser (T, Y)** est la réponse négative du fournisseur Y à la demande du client.

Description du modèle RdPC associé

D'après la figure 18. Le processus se déclenche par l'envoi d'une demande provenant de l'agent ayant une tâche à allouer et en consultant la table des compétences C des autres agents via la transition t_2 ; Après le franchissement de cette transition deux possibilités se présentent. Soit l'agent demandeur trouve un ensemble E d'agents susceptibles de réaliser la tâche demandée, dans ce cas la transition t_3 est tirée, et le premier de la liste des fournisseurs (agent) sera sollicité pour l'exécution de la dite tâche, par conséquent la transition t_{10} sera tirée, ce qui engendre l'émission d'un message de la forme Demander (T, self) de l'agent client lui indiquant la réalisation de la tâche T.

Si le fournisseur sait exécuter la tâche T, il envoie un message de la forme Accepter (T, Y) qui signifie une réponse positive du fournisseur Y pour la réalisation de la tâche T. Sinon le fournisseur envoie un message de la forme RefuserF (T, Y) comme réponse négative du fournisseur Y à la demande d'allocation de la tâche T.

Si les transitions t_4 et t_7 sont franchies, cela signifie qu'aucun agent de l'ensemble de la liste dont dispose le client ne peut réaliser la tâche T, par conséquent pour que le système ne se bloque pas (pas de défaillance du système), le client doit s'assurer que dans tous les cas, il y aura au moins un agent (fournisseur) qui acceptera de réaliser la tâche. Pour surpasser cet obstacle, l'agent déclenchera le processus d'évaluation des exposés des motifs par appel d'offre afin de choisir l'agent qui a le moins de raisons de refuser cette tâche. Le mécanisme d'évaluation des exposés des motifs par appel d'offre sera décrit dans le paragraphe suivant.

B- Mécanisme d'évaluation des exposés des motifs par appel d'offre

Le réseau des exposés des motifs est fondé sur la notion d'appel d'offre. Il reprend le protocole de l'élaboration de contrats dans les marchés publics. La relation entre le client, ou l'administrateur, et les fournisseurs, ou l'offrant, passe par l'intermédiaire d'un appel d'offres et une évaluation des propositions des arguments du rejet de la proposition d'allocation d'une tâche donnée par un client envoyé par les fournisseurs. L'appel d'offres s'effectue en quatre étapes :

1. Dans la première étape, l'administrateur envoie une requête contenant une description de la tâche qu'il voudrait voir effectuée et une demande concernant les différentes raisons du rejet de la demande au sujet de la réalisation de la tâche T à tous les agents du système.

2. Dans la deuxième étape, les offrants élaborent une proposition contenant les exposés des motifs concernant les raisons du rejet de la demande au sujet de la réalisation de la tâche T et l'envoie à l'administrateur.

3. L'administrateur reçoit et étudie les exposés des motifs selon le degré et la nature des raisons ayant trait au rejet de la demande, il attribue le marché à l'offrant ayant des raisons moins prioritaires.

4. Enfin, dans la dernière étape, l'offrant à qui le marché est attribué devient ainsi le contractant, envoie un message à l'administrateur lui indiquant qu'il est toujours d'accord pour accomplir la tâche requise et qu'il s'engage de ce fait à la réaliser ou bien qu'il ne peut s'acquitter du contrat, ce qui relance l'évaluation des offres et l'attribution du marché à un autre agent (et on revient à l'étape 3).

Un agent peut assurer les deux rôles à la fois, celui de l'administrateur et de l'offrant. Cependant, il est préférable de décomposer cet algorithme en distinguant la définition du comportement de l'administrateur de celui de l'offrant. Les appels d'offre seront reconnus dans le réseau de manière unique par une clé formée de la description de la tâche, de l'administrateur responsable de la diffusion de l'appel et d'un numéro d'ordre local à ce dernier.

Pour la réalisation d'un mécanisme d'appel d'offre on retrouve les types de messages utilisés dans les autres systèmes d'allocation distribués à savoir : Allouer, la réponse Impossible Faire et l'engagement de l'offrant Engagé Faire. En plus les types de messages suivants :

- **Appel Offre(T,X)** est un message de l'administrateur X à un offrant potentiel pour lui demander d'envoyer les raisons du rejet de la proposition pour la réalisation de la tâche T.

- **Argument (T,A)** est la réponse d'un offrant Y à un appel d'offre concernant la tâche T. Il retourne alors un argument A (les raisons du rejet de la demande concernant la réalisation de la tâche T), qui comprend entre autres le nom de l'offrant.

- **Attribuer (T,C,X)** est le message de l'administrateur X à un offrant pour lui signifier qu'il lui attribue le contrat C portant sur la tâche T.

- **Accepter (T,Y)** est une réponse positive de l'offrant Y à l'attribution du contrat par l'administrateur.

- **Refuser (T,Y)** est la réponse négative de l'offrant à l'attribution du contrat par l'administrateur, ce qui relance l'évaluation et le processus d'attribution

- Description du modèle RdPC sous-jacent du mécanisme d'évaluation des exposés des motifs par appel d'offre

L'administrateur envoie l'appel d'offre à l'ensemble des agents du système représentés par leur identificateur dans la place P_{13}. Les fournisseurs (agents), envoie un message de type Argument (T ,R) où l'on décrémente le nombre d'agents et l'on mémorise l'ensemble des offres (argument) représenté par la lettre A dans la place P_{16} via la transition t_{13}. Lorsque tous les agents répondent (card(A) =0), l'administrateur sélectionne par t_{16} l'agent (représenté par le jeton Y) qui a le moins de raisons de refuser cette tâche à qui il va attribuer le contrat (le jeton C). L'offrant peut répondre positivement par l'envoi du message Accepter (T,Y) à l'administrateur. Ce dernier lui répond par un message d'engagement (via la transition t_{14}. Comme il peut refuser le contrat en envoyant le message Refuser(T,Y). Dans ce cas l'administrateur doit choisir un autre agent (offrant) s'il existe, sinon il envoie un message du type Impossible Faire(T). Voir la figure 18.

Mécanisme d'évaluation des exposés des motifs par appel d'offre:

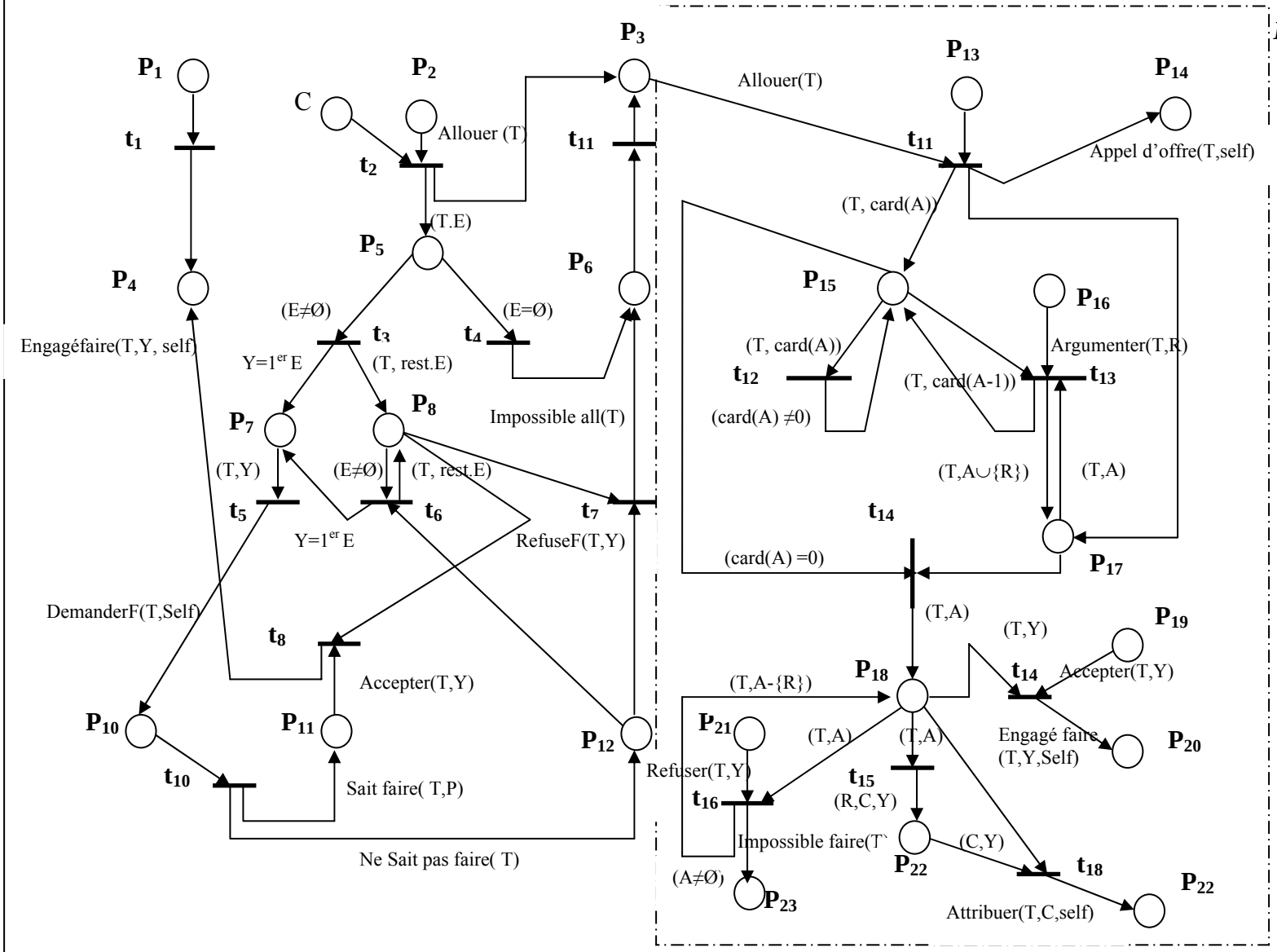


Fig18 : Modélisation par RdPC de l'allocation distribuée des tâches

IV -3-2- Les Interactions entre Agents Antagonistes

Les interactions entre les agents antagonistes se basent principalement sur la négociation. On peut voir que la définition de la négociation est aussi imprécise que celle du concept d'agent. On arrive malgré tout à retrouver certaines caractéristiques importantes de la négociation et identifier certains cas où cette dernière est indispensable dans le cadre des interactions dans les systèmes multi-agents. Les principales situations conflictuelles et compétitives que les agents antagonistes peuvent rencontrer sont [26]:

- Compétition individuelle pure : C'est le cas des agents ayant des buts incompatibles, des ressources suffisantes et des compétences suffisantes. Les agents doivent lutter ou négocier pour atteindre leurs buts. La compétition suppose que les agents ont tous les mêmes ressources à leur disposition et qu'ils sont placés dans des situations initialement identiques. On parle de compétition pure, lorsque les ressources ne constituent pas l'enjeu du conflit. Il n'y a pas de problèmes spécifiques d'interaction liés à ce type de situation.

- Compétition collective pure : On suppose que les agents ont des buts incompatibles, des ressources suffisantes, et des compétences insuffisantes. Lorsque les agents n'ont pas de compétences suffisantes, ils doivent se regrouper au sein d'une association pour parvenir à atteindre leurs objectifs.

- Conflit individuel pour des ressources : C'est le cas où les buts sont incompatibles, les ressources insuffisantes et les compétences sont suffisantes. Lorsque les ressources ne peuvent pas être partagées, on se trouve dans une situation caractéristique de conflits dont les ressources sont en jeu, chacun voulant les acquérir pour lui seul.

- Conflits collectifs pour les ressources : Dans ce dernier cas, les agents ont des buts incompatibles, les ressources disponibles sont insuffisantes, et les compétences sont aussi insuffisantes. Ce type de situation combine la compétition collective avec les conflits individuels pour les ressources. L'association lutte les unes contre les autres pour obtenir le monopole d'une ressource.

La principale technique utilisée pour gérer ces situations conflictuelles et compétitives qui pourraient mettre en péril des comportements coopératifs entre les agents antagonistes est la négociation.

❖ La Négociation

On peut définir la négociation comme le processus d'améliorer les accords sur des points de vue communs ou des plans d'actions, voir le partage de ressources grâce à un échange structuré d informations pertinentes.

La négociation s'appuie généralement sur l'établissement de contrat d'accord entre les agents. Les agents coordonnent leurs activités grâce à l'établissement de contrats pour atteindre des buts spécifiques. Un agent, propose son contrat (une tâche, un problème, partage de ressources) aux autres agents concurrents afin d'étudier et d'évaluer les clauses du contrat. Les agents qui possèdent les ressources appropriées, et l'information requise envoient des soumissions qui indiquent leurs capacités à s'engager à respecter les engagements des contrats.

- **Modélisation de la négociation par un RdPC**

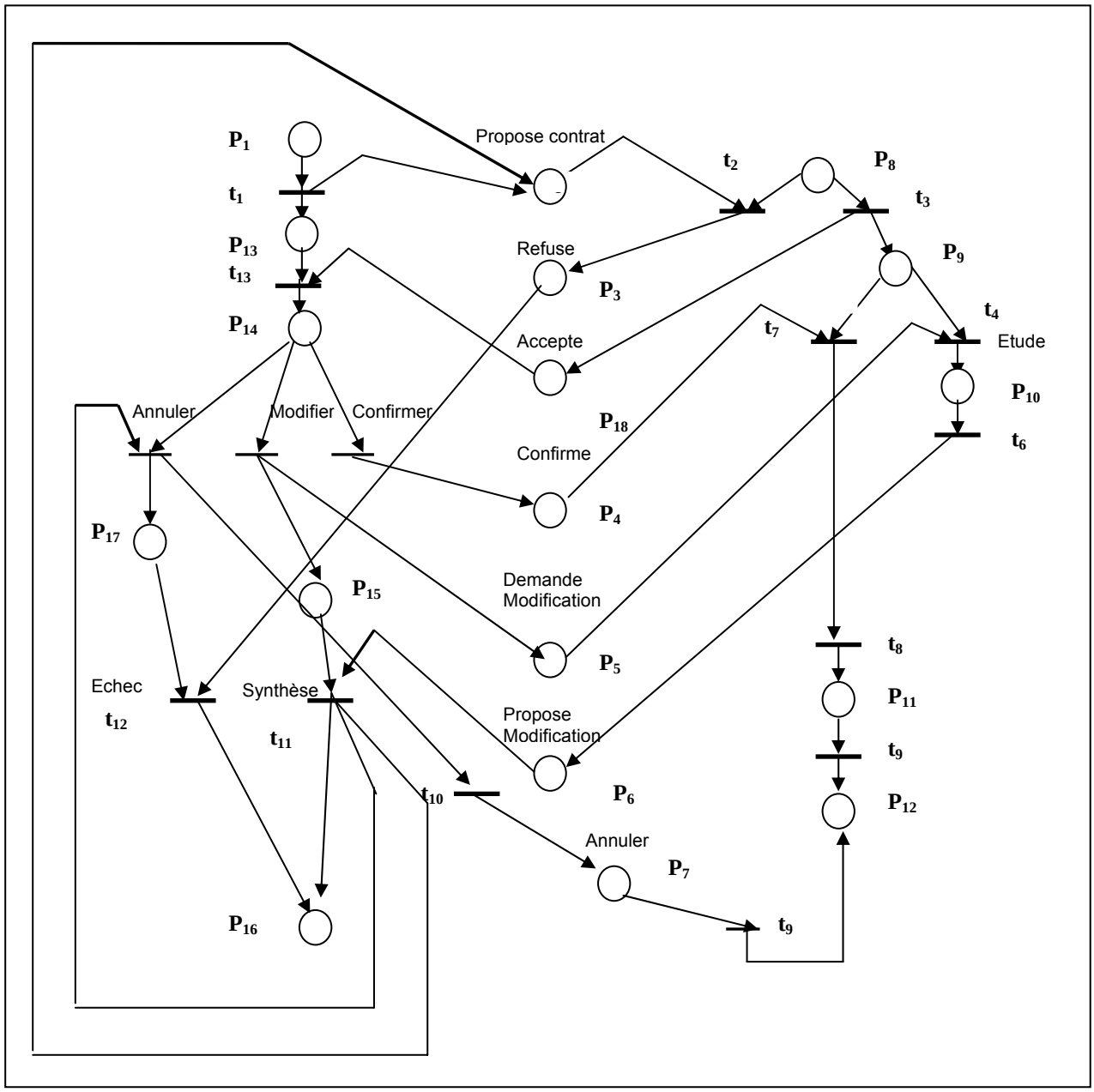


Fig 19 : Modélisation de la négociation par un réseau de Petri coloré

Interprétation

Les places $\mathbf{P_1}$ et $\mathbf{P_8}$ décrivent les états initiaux dans lesquels se trouvent les agents avant le début du processus de négociation. Les places $\mathbf{P_{16}}$ et $\mathbf{P_{12}}$ représentent les états de fin de la négociation. A partir de l'état $\mathbf{P_1}$, l'agent initiateur du contrat envoie une proposition de contrat à un autre agent (message propose contrat) et passe à l'état $\mathbf{P_2}$ qui représente une attente de réponse. Dès la réception du message, l'agent B l'examine et décide de l'accepter ou de la rejeter (message accepte ou refuse). S'il ne veut pas de ce contrat, il envoie un refus à l'agent initiateur qui passe à l'état $\mathbf{P_{16}}$. Sinon il accepte le contrat, alors il envoie un message d'acceptation. Pendant ce temps, l'agent B se trouve en attente à l'état $\mathbf{P_9}$. Après l'écoulement du temps d'attente et la réception du message d'acceptation, l'agent initiateur prend une décision :

1- Soit il confirme le contrat auprès de l'agent B en lui envoyant un message confirme, l'agent B est lié à l'agent initiateur par un contrat d'engagement, l'agent B se place alors à l'état $\mathbf{P_{11}}$ (sous contrat) puis à l'état $\mathbf{P_{12}}$ fin de négociation ;

2- Il annule le contrat alors l'agent initiateur, passe à l'état échec puis à l'état $\mathbf{P_{12}}$ fin de négociation ;

3- Il demande de proposer une modification pour ce contrat (message demande modification) et l'agent initiateur passe à l'état $\mathbf{P_{15}}$ en attente d'une proposition de modification. L'agent B reçoit le message, passe à l'état $\mathbf{P_{10}}$ (prépare une proposition de modification de contrat) et envoie sa proposition. L'agent initiateur reçoit la proposition, effectue une synthèse, la proposition est soit annulée, le cas échéant, l'agent initiateur passe à l'état échec $\mathbf{P_{17}}$, puis à l'état $\mathbf{P_{16}}$ fin de négociation, ou bien alors il propose un nouveau contrat.

IV -4- Conclusion

Dans ce chapitre nous avons présentés :

Premièrement, l'architecture des agents dotés à la fois d'un comportement cognitif et réactif. Cette architecture repose principalement sur trois modules à savoir : le module de communication, le module de perception et le module décision. Pour l'analyse et la vérification du fonctionnement et du comportement des modules de communication et de perception, nous avons utilisés les réseaux de Petri colorés.

Deuxièmement, nous avons proposés une formalisation des interactions par les réseaux de Petri colorés, d'une part entre les agents coopératifs qui se traduisent par l'allocation des taches distribuées. Cette formalisation remédiée au problème du mode d'allocation de tache distribuée (aucun agent n'accepte d'exécuter la tache allouée par un autre agent) et d'autre part les interactions entre les agents antagonistes qui se traduit par la négociation.

Nous avons utilisés dans nos modèles les réseaux de Petri colorés, car ils sont bien adaptés pour représenter le parallélisme et la synchronisation, une des caractéristiques clé des systèmes multi-agents. En plus, la notion de couleur que ces réseaux fournissent permet de distinguer l'information relative à une marque dans une place du RdPC, ce qui caractérise plusieurs types d'objets. Par conséquent les réseaux de Petri colorés représentent et transportent des informations divers et structurées que les agents manipulent et échangent.

Conclusion et Perspectives

L'intérêt des systèmes multi-agents est reconnu depuis longtemps. Cependant, leur utilisation se heurte à quatre obstacles majeurs : leur notions de base ne sont pas caractérisées de façon claire. Les méthodologies de leur développement sont à l'état artisanal. Les environnements permettant d'assister les développeurs sont pratiquement inexistants et l'utilisation des méthodes et les outils de vérification et validation nécessite une mise en œuvre fine et raffinée.

En plus, la puissance d'expression des modèles multi-agents qui permettent de représenter des entités autonomes en interactions, dotées de comportement et pouvant évoluer dans un environnement, est mise à l'épreuve, lorsqu'il faut analyser un système et identifier ses propriétés globales et générales qui sont généralement étudiés par simulation. Cependant l'utilisation de méthodes formelles de spécification pour représenter les modèles multi-agents s'avère nécessaire.

Comme les simulations multi-agents ne permettent pas de prévenir par exemple la non observabilité de certaines configurations non désirées d'un système donné, voir la non existence de situation de blocage, par conséquent, les conclusions sur les évolutions des systèmes modélisés ne peuvent être que partielles. De ce fait, le recours à des modèles formels, pour la vérification et validation des résultats des simulations multi-agents relève d'une très grande portée.

C'est pour ces raisons que beaucoup de chercheurs ont investi dans le domaine des SMA, et particulièrement l'utilisation des réseaux de Petri comme outil de formalisation et de modélisation.

Les modèles formels tels que les réseaux de Petri (réseaux de Petri colorés) permettent de représenter aisément le parallélisme (une des caractéristiques principales des systèmes multi-agents) et la synchronisation et ils fournissent de nombreuses techniques d'analyse et de validation (l'examen du graphe des marquages accessibles....).

Notre démarche a consisté à montrer comment certains des concepts fondamentaux des Systèmes multi-agents peuvent être exprimés par une formalisation abstraite à l'aide des réseaux de Petri. Nous avons présenté une modélisation qui s'articule autour des principaux aspects des SMA à savoir la définition d'une architecture d'agents à base d'un RdP et la formalisation des différentes formes des interactions tells que la coopération, la coordination et la négociation.

Les perspectives de ce travail sont nombreuses, nous citons :

1-L'amélioration des modèles proposés en réduisant le nombre de messages échangés entre les agents lors des interactions, afin d'augmenter les performances d'un système modélisé par un SMA.

2- Définir et concevoir une plate forme de simulation d'agents qui repose sur l'architecture que nous avons proposé et qui prend en considération les incertitudes et les erreurs tells que les panne qui peuvent se produire aux niveaux des agents, les imprécisions et l'incomplétude des informations perçues, la non résistance aux bruits et aux perturbations de l'environnement,……etc.

3- Adapter l'architecture que nous avons proposée à un modèle de confiance qui permet de déterminer le niveau de confiance qu'un agent a envers un autre agent.

Bibliographie

[01] Bakam.I, ***Des systèmes multi-agents aux réseaux de Petri pour la gestion des ressources naturelles*** : le cas de la faune à l'Est du Cameroun, Thèse de doctorat 2002.

[02] Bousquet. F***, Modèles multi-agents pour la gestion de l'environnement: vers un couplage simulations et approche formelles de modélisation,*** Acte du colloque SMAGET, Clermont Ferrand 998. 1998.

[03] Bousquet. F *&* Barreteau. O *&* Mullon. C, Weber. J, ***Modélisation d'Accompagnement: Systèmes Multi-Agents et Gestion des Ressources Renouvelables,*** Actes du Colloque international "Quel environnement au 21ème siècle ? Environnement, maîtrise du long terme et démocratie", GERMES, Paris, France, sous presse 1999.

[04] Books.R, Intelligence ***with out representation, Artificial Intelligence***, 1991.

[05] Brooks R., A robot that walks, ***emergent behaviour from a carefully evolved network,*** Neural computation, 1:253-62, 1989.

[06] Cardon A. & Guessoum Z, ***Systèmes multi-agents adaptatifs,*** Actes des Huitièmes Journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-Agents , Novembre 2000.

[07] Chaib-draa. B, ***Distributed Artificial Intelligence,*** An overview. In A. Ken, J. G. Williams, C. M. Hall, and R.Kent, editors, Encyclopedia Of Computer Science And Technology, volume 31, pages 215-243. Marcel Dekker, Inc, 1994.

[08] Chaib-draa. B, ***Industrial applications of distributed AI Communications of the ACM,*** 38(11) :49-53, 1995. [25] B. Chaib-draa. Interaction between agents in routine, familiar and unfamiliar situations. *International Journal of Intelligent and Cooperative*
*Information System*s, 1(5):7-20, 1996.

[09] Chevrier V, ***Etude et mise en oeuvre du paradigme multi-agents,*** De l'atome à Gtmas, Thèse de l'Université de Nancy I, 1993.

[10].David. R, H. Alla, **Du Grafcet aux Réseaux de Petri**, Hermes 1997.

[11] Demazeau Y, ***From interactions to collective behaviour in agent-based systems,*** Proceedings of the first European Conference on Cognitive Science'95:117-132,1995.

[12] Demazeau. Y & Müller, **Decentralized Artificial Intelligence Chapter From reactive to intentional agents**, Elsevier Science Publishers. 1991.

[13] Durfee E.H. & Montgomery T.A, **A hierarchical protocol for Coordinating Multi Agents, Behaviours**. AAAI-90, 1990.

[14] Deneubourg J, L., Goss S., Francks N.R, Sendova-Franks A,
The dynamics of collective sorting : robot-like ants and ant-like robots*,* Dans Meyer J-A et Wilson S. Eds,

[15] A. Drogoul, J.-D. Zucker, LIP6 1998/041, ***Rapport de Recherche LIP6,*** *30 pages - Octobre/October 1998.*

[16] Durfee E.H, Lesser V.R et Corkill D.D,
Coorperative distributed problem solving, in the handbook of artificial intelligence*,*
vol IV, A Barr, P.R Cohen et E.A Feigenbaum (Ed), p.83-148, Addison-Wesley,1989.

[17] Durfee E.H. and Montgomery T.A., MICE, ***A Flexible Testbed for Intelligent Coordination Experiments. Artificial Intelligence Laboratory****,* Proceedings of the Ninth Workshop on Distributed AI, 1989.

[18] El Fallah S, **Représentation et manipulation de plans à l'aide des réseaux de Petri,** Acte des 2è journées Francophones IADS-SMA. Vorionl994.

[19] El Fallah. S A. Haddad *&* Taghilit. M, **Introduction aux systèmes de planificatio*n.*** Support de cours du DEA de l'université de Paris Dauphine. 1997.

[20] El Fallah S, Haddad A, & Mazouzi. H, Observation **répartie et analyse des interactions dans un système Multi-Agents,**Actes des JFIADSMA 98. Editions Hermès, Nancy l998.

[21] Erceau J, ***Intelligence Artificielle Distribuée et Systèmes Multi-Agents de la théorie aux applications,*** 23ème Ecole Internationale d'Informatique de l'AFCET, Neuchâtel, 1993.

[22] Ermine J.-L, ***Les systèmes de connaissance*s**, 2 *ème* éditions, Hermès Science Publications, Paris, 2000.

[23] Esmahi L, ***Coopération dans les systèmes multi-agents : une analyse et un point de vue,*** JFIADSMA'01, p31-53, Montréal, Canada, 2001.

[24].Fernand.N , Demaseau.Y & Beijs.O, **Systèmes multi-agents et résolution de problèmes spatialisés**, Revue d'intelligence artificielle, 1998.

[25] Ferber. J,***Many agents simulation and artificial life,*** Chapter simulating reactive agents. IOS Press.

[26] Ferber. J, **Les systèmes multi-agents vers une intelligence collective**, Inter Edition, Paris, 1995.

[27] Ferber J. , Gutknecht.o, ***A meta-model for the analysis and design of organizations in multi-agent systems****,* Proceedings of ICMAS '98, IEEE Press.

[28] Ferber J., Multi-Agent Systems, ***An introduction to distributed artificial***

intelligence, Addison-Wesley, ISBN 0-201-36048-9, 1999.

[29] Ferguson. A Touring Machines, ***An Architecture for Dynamic, Rational, mobileAgent*s,** PhD thesis, Clare Hall, University of Cambridge, UK, November 1992. (Also available as Technical Report No. 273, University of Cambridge Computer Laboratory).

[30] Fougères.A ***Une architecture cognitive d'agents communicants dans des systèmes d'information complexes*,** 2000.

[31] **FIPA 2002** Foundation for Intelligent Physical Agent. Site web www.fipa.org.

[32] Gallone J-M., Charpillet F. & Chevrier V, ***Un modèle d'agent pour un raisonnement contraint par le temps,*** Dans: Actes Journée PRC-IA sur les Systèmes Multi-Agents, Paris, Décembre 1994.

[33] Gasser L., Braganza C., & Herman N, ***MACE: A flexible testbed for distributed AI Research,*** in Distributed Artificial Intelligence, Michael N. Huhns, Ed. San Mateo, CA: Morgan aufmann Publishers, pp. 119-152, 1987.

[34] Georgeff and A. L. Lansky, ***Reactive reasoning and planning***.In *The Proceedings of AAAI-8*7, pages 677-682, Seattle, 1987.

[35] Gessoum. Z, *Dima :* **Une plate forme multi-agents en Smaltalk**, Revue Objet. 1998.

[36] Gessoum. Z, ***Les protocoles d'interaction dans les SMA,***
Support de cours du DEA de l'université de Paris 6, 2002.

[37] Jensen. k, **Coloured Petri Nets: Basic Concepts, analysis methodes** and Practical use, Volume 1 Springer Verlay London..

[38] Jennings N.R., Sycara, & Wooldridge M, ***Autonomous Agents and Multi-Agent Systems****, 1998.*

[39] Jenings J.R, ***Commitments and Conventions : The foundation of coordination in Multi Agent systems,*** Theknowledge Engineering review.

[40] Labidi. S & Ledjouad. W , **De l'intelligence Artificielle Distribuée aux Systèmes Multi Agents**, Technical report 2004, INRIA, Sofia Antipolis, 1993.

[41] Labed A, ***RDP Temporisés, RDP Hybrides, Automates Temporisés Etude comparative en vue de la vérification et de l'évaluation de performances des SEDT,*** Thèse de doctorat, 2002.

[42] Le Moigne J.-L, ***La modélisation des systèmes complexe*s**, Dunod, 1990.

[43] Maes P, ***Intelligent Software,*** Scientific American, volume 273, numéro 3, septembre 1995, pp. 84-86.

[44] Malone T.W., **What is coordination theory ,** in national science foundation coordination theory workshop, MIT, 1988.

[45] Magnin. L, Ferber. J, **Conception de systèmes multi-agents par composante modulaires et réseaux de Petri**, Actes des journées du PRC-IA, Montpellier, 1993.

[46] Matthieu. A, ***Un modèle Componentiel dynamique pour les systèmes multi-agents organisationnels***, Thèse de doctorat, 2003.

[47] Mazouzi H, ***Ingénierie des protocoles d'interaction : des systèmes distribués aux systèmes multi agents,*** Thèse de doctorat. 2001.

[48] Mesle, R, **ICHTYUS : Architecture d'un système multi-agents pour l'étude de structures agrégatives,** Rapport de DEA, LAFORIA, Université Paris 6, 1994.

[49] Morin E, ***La nature de la nature, Le Seuil,*** *1977.*

[50] Newell A, ***Unified Theories of Cognition, Harvard Univ,****. Press, Cambridge, Mass., 1990.*

[51] Parunak V.D***, Agent-based emulation vs. Numerical Evaluation,***
A users'guide. Multi-agent Based Simulations, 1998.

[52] Russell, Stuart J. & Peter Norvig **Artificial Intelligence: A Modern Approach**, Englewood Cliffs, NJ: Prentice Hall,1995.

[53]. Singh M. P, ***Multi-agents Systems : A Theoretical Framework for Intentions, Know-How,and Communications*** *(LNAI Volume 799).* Springer-Verlag : Heidelberg, Germany,1994.

[54] Steels, L, ***Building agents with autonomous behavior systems***,
In L. Steels & R. Brooks, eds, `The`artificial life' route to `artificial ntelligence'. Building situated embodied agents.' Lawrence Erlbaum Associates, New Haven, 1994.

[55] Sycara K, ***Multi-agents compromise via negociation, in distributed artificial intelligence,*** L.Gasser et M.Huhns (Ed), Pitman, 1989.

[56] Vally J.Dy & Courdier.R, ***Uniformisation des mécanismes de conception de SMA*** Acte des journées Francophones JFIAD 2002

[57] Vernadat. F, Lanusse. A et Azema. P, Modélisation **par réseaux de Petri d'un langage acteur: Application à la vérification des systèmes multi-agents**, Acte des 2ème journées Francophones IADS-SMA. Voiron, 1994.

[58] Jeennings N. R.& Wooldridge M, ***Agent-Oriented Software Engineering» in Handbook of Technology.*** Edition. J. Bradshaw AAAI/MIT Press. 2000.

[59] Weyns. D, ***A Colored Petri Net for a Multi-Agent Application*****,**
Computer science department Katholieke Universiteit Leuven, Belgium, 2003.