

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Hadj Lakhder - Batna
Département d'informatique

MEMOIRE

Présenté pour l'obtention du diplôme de Magistère en
Informatique

Option : Système Informatique Intelligent et Communicant (SiIC)

Thème

AR-MAS : Approche Multi-Agent Basé-Rôles pour Systèmes Adaptatifs

Présenté par :

Monsieur Sofiane KEMCHI.

Dirigé par :

Docteur Ammar LAHLOUHI.

Soutenu le 28/05/2015

devant le jury composé du :

Président :

Dr Samir ZIDAT.

MCA à l'université Hadj Lakhder – Batna.

Rapporteur :

Dr Ammar LAHLOUHI.

MCA à l'université Hadj Lakhder – Batna.

Examineurs :

Dr Mourad BOUZENADA.

MCA à l'université de Constantine 2-ABDELHAMID Mehri.

Dr Hacene BELHADEF.

MCA à l'université de Constantine 2-ABDELHAMID Mehri.

Remerciements

Je voudrais remercier Dr Ammar LAHLOUHI, Maître de Conférences A à l'université Hadj Lakhder de Batna, qui a encadré mon mémoire de magistère. Je le remercie pour sa disponibilité et les conversations régulières que nous avons eues sur ce travail, sa rigueur et notamment pour son art consommé de la synthèse dès qu'il s'agit d'avoir une vue d'ensemble ou encore d'étudier un point précis. Mais au-delà des qualités scientifiques, ce sont surtout les qualités humaines que je voudrais souligner. Je remercie spécialement les membres de jury.

Finalement, ce travail de mémoire doit beaucoup à mon père, ma mère, mes frères et ma sœur qui m'ont encouragé et soutenu depuis toujours. Je leur adresse un immense merci.

Ce serait inconcevable de ne pas remercier tous les membres de ma belle famille, qui m'ont vivement soutenu tout au long du travail notamment mon beau-père et ma belle-mère pour l'ambiance chaleureuse avec laquelle ils m'ont entouré.

Enfin et pas la moindre, je tiens à exprimer mes profonds sentiments et mes remerciements à ma douce fiancée pour son soutien indéfectible. Un grand merci est destiné à tous ceux qui m'ont, moralement ou pratiquement, soutenu et que ces lignes ont, accidentellement, oublié de mentionner.

A mon père et à ma famille.

RESUME

L'informatique envahie chaque jour de nouveaux domaines d'applications de plus en plus complexes. De telles applications ont des besoins qui varient dans le temps et dans l'espace. Tenir compte de cette variabilité au cours du développement est très difficile. Il faut alors doter de tels systèmes de capacités d'auto-adaptation leur permettant d'adopter des aspects ne pouvant être fixés directement dans le développement initial.

Plusieurs approches ont été proposées pour mettre en œuvre l'adaptation de systèmes. Cependant, le problème encore ouvert et beaucoup de travail reste à faire dans ce domaine. Dans ce projet, nous visons une approche multi-agent pour l'auto-adaptation. La suprématie des modèles multi-agent sur les approches classiques vient des aspects d'autonomie des agents et de leur coopération dans la mise en œuvre des tâches. La première permet des adaptations locales des comportements des agents sans influencer le reste du système. Ce qui simplifie amplement le processus d'adaptation. La seconde permet d'avoir différentes configurations du système multi-agent pour la mise en œuvre d'une tâche donnée. Ce qui offre à un ensemble d'agents plusieurs possibilités pour atteindre ses objectifs.

Plusieurs inspirations ont été utilisées pour le développement de systèmes multi-agent. La plus concrète d'entre elles est celle basée organisation de rôles. Cependant, peu de travaux ont abordé les systèmes multi-agent adaptatifs avec une inspiration basée organisation. Dans notre mémoire on propose une approche pour l'adaptation des systèmes multi-agent basés organisation de rôles, une approche capable de répondre aux contraintes d'un environnement ouvert, dynamique et en constante évolution. Pour cela nous prenons pour base le concept d'adaptation du système multi-agent par une réorganisation d'assignation des rôles aux agents. Comme moyen technique nous utilisons les systèmes multi-agents basés organisation (auto-organisation) des rôles, La fonction globale du système est une fonction émergente guidée par des contraintes d'organisation.

Mots clés : Adaptation, Systèmes multi-agent, Organisation multi-agent.

Abstract

Everyday computer science invade new field of applications more and more complex. Such applications have needs which varied in space and time. It is very difficult to take this variability into account in development action. So we have to endow these systems with auto adaptation capacity to permit them to adopt aspects which can't be fixed directly in initial development.

A lot of approaches have been proposed to implement adaptation of systems. However, many works to do in this field and the problem is still open. In this project we focus a multi agent approach for auto adaptation. The supremacy of multi agent model on classical approach, come from the aspect of autonomy and their cooperation to achieve tasks. The first one allow local adaptation of agent behavioral without influence on the rest of the system this make easy the adaptation process. The second one allows having different configuration of multi agent system to achieve a task this give to agent many possibilities to reach objective.

Many inspirations have been used to develop multi agent system. The most concrete of them are those based role organization. However, few works tackled multi agent system with inspiration based organization. In this memory we propose role based organization approach to adapt multi agent systems, an approach capable to deal with open environment, dynamic and in constant evolution. For this goal we have a reorganization of role allocation to agent as a base and role based organization multi agent system as technical mean. The global function of the system is an emergent function which is guided by organization constraints.

Key words: adaptation, multi agent system, multi agent organization.

Table des matières

Chapitre 1 Introduction générale.....	1
1.1. Contexte du projet.....	2
1.2. Motivations.....	2
1.3. Problématique et objectif.....	4
1.3.1. Problématique.....	4
1.3.2. Objectif.....	5
1.4. Organisation du mémoire.....	6
Chapitre 2 Les systèmes multi-agent.....	8
2.1. Définition d'un agent.....	8
2.2. Définition d'un système multi-agent.....	10
2.3. Origines.....	13
2.4. Classification d'organisation.....	14
2.5. L'interaction.....	19
2.6. L'adaptation.....	20
Chapitre 3 Organisation multi-agent et réorganisation.....	21
3.1. Définition d'organisation multi-agent.....	21
3.2. Structures organisationnelles et organisations concrètes.....	23
3.3. Définition d'auto organisation.....	24
3.4. Définition de la réorganisation.....	24
3.5. Processus de réorganisation.....	26
3.6. Quelques travaux de réorganisation.....	26
a) SMA a organisation émergente.....	27
b) SMA basés sur la coalition.....	28
c) Ingénierie orienté agent.....	30
d) SMA orientés organisation.....	32

Table de matières

Chapitre 4 Approches pour la conception des systèmes multi-agent adaptatifs.....	35
Résumé.....	35
4.1. Introduction.....	35
4.2. Classification.....	36
4.3. Approches multi-agent pour systèmes adaptatifs.....	38
4.4. Comparaison.....	62
4.5. Conclusion.....	63
Chapitre 5 Problème posé et solution proposée.....	64
5.1. Problématique.....	64
5.2. Approches existantes.....	64
5.3. Contribution.....	65
5.4. Scénario de validation.....	67
5.5. Conception.....	68
5.5.1. Développement des tâches de systèmes.....	69
5.5.1.1. Moyens de description et processus de développement..	69
5.5.2. Développement des modèles d'organisations.....	71
5.5.2.1. Organisation élémentaire.....	71
5.5.2.2. Organisations composées.....	73
5.5.2.3. Moyens de description.....	74
5.5.3. Développement des modèles de systèmes multi-agent.....	84
5.5.3.1. Moyens de description.....	84
5.5.4. Modélisation multi-agent d'un environnement de services pour touristes (commerce électronique).....	85
5.5.4.1. Organisation d'application de services pour touristes.....	85
5.5.4.2. Développement d'un modèle multi-agent d'application de services pour touristes.....	85
Chapitre 6 Mise en œuvre.....	91
6.1. Implémentation.....	91
6.1.1. Environnement d'implémentation Netlogo.....	91
6.1.2. Choix et prise de décision	92
6.1.3. Etude du comportement.....	92
6.1.4. L'adaptation dans notre environnement.....	93
6.1.5. Expérimentations et résultats.....	94
6.1.5.1 Cas d'étude.....	94
6.1.5.2 Emergence d'affectation des rôles.....	96
6.1.5.3 Discussions.....	97

Table de matières

Conclusion et perspectives.....	98
I. Conclusion.....	98
II. Perspectives.....	99
Bibliographies.....	100

Liste des Figures

Fig.1 représentation d'un agent en interaction avec son environnement.....	10
Fig.2 représentation imagée d'un agent en interaction avec son environnement et les autres agents.....	11
Fig.3 positionnement des SMA.....	14
Fig.4 la relation micro-macro dans les systèmes multi-agents.....	15
Fig.5 (a) SMA émergents ; (b) SMA basés sur les coalitions ; (c) Ingénierie orientée agent ; (d) SMA orientés organisation. Le concepteur/observateur peut être le développeur/utilisateur (cas exogène) ou un ensemble d'agents (cas endogène).....	17
Fig.6 organisation et structures organisationnelles.....	24
Fig.7 adaptation par auto organisation d'un système en interaction dynamique.....	27
Fig.8 phases du processus de la méthode Cassiopée.....	30
Fig.9 le processus de développement Gaia et les méthodes manipulées.....	32
Fig.10 schémas modèle d'un rôle.....	39
Fig.11 schémas du rôle 'CoffeeFiller'.....	40
Fig.12 définition du protocole Fill.....	40
Fig.13 modèle d'organisation.....	43
Fig.14 plan arborisant définissant le rôle physicien.....	47
Fig.15 modèle d'organisation.....	48
Fig.16 représentation d'un ensemble de tâches d'organisation.....	51
Fig.17 diagramme de transition.....	52
Fig.18 exemple de trafic avec cameras.....	55
Fig.19 scénario illustratif d'une abstraction basic du modèle	56
Fig.20 Interaction a l'aide d'un rapport de dominance.....	66

Liste des figures

Fig.21 description informelle de la tâche du système d'application de services pour touristes.....	77
Fig.22 Description formelle de la tâche d'une application de services pour touristes. La description des éléments du CPN est donnée dans le tableau3..	77
Fig.23 Description informelle de la tâche de l'organisation d'application de services pour touristes.....	78
Fig.24 Description formelle de la tâche de l'organisation d'application de services pour touristes. La description des éléments de ce CPN est similaire à celle de la figure 22 donnée dans le tableau 3.....	79
Fig. 25 Représentation graphique de la relation de communication entre les représentants des sous organisations de l'organisation d'application de services pour touristes TSOA de la table 14.....	79
Fig.26 Description informelle de la tâche d'un système de réservation.....	79
Fig.27 Description formelle de la tâche du système de réservation. La description des éléments du CPN est donnée dans le tableau 4.....	81
Fig.28 Description informelle de la tâche de l'organisation de réservation.....	81
Fig.29 Description formelle de la tâche de l'organisation de réservation. La description des éléments de ce CPN est similaire à celle du CPN de la figure 27 donnée dans le tableau 4.....	82
Fig.30 Représentation graphique de la relation de communication entre les rôles de l'organisation des brokers agent.....	83
Fig.31 Description formelle de la tâche du modèle de rôles BM	86
Fig.32 Description formelle de la tâche du modèle de rôles BCM	87
Fig.33 Représentation graphique des liens de communication entre les agents du système multi-agent de l'équipe de broker agent.....	89
Fig.34 Environnement de notre application de service Pour touristes.....	91
Fig.35 Choix et prise de décision.....	92
Fig.36 établissement des liens en rouge entre les rôles en bleus.....	94

Liste des figures

Fig.37 affectations des rôles (1, 2, 3) en bleus aux agents broker en rouge en établissant les liens correspondant entre les agents.....95

Fig.38 comparaison de la courbe des liens entre les agents et la courbe des liens entre les rôles.....95

Fig.39 courbe du taux de migration des rôles au cours d'évolution.....96

Liste des Tables

Tableau 1 opérateur d'expression de service.....39

Tableau 2 tableau comparatifs.....63

Tableau 3 Description des éléments du CPN de la figure 22.....78

Tableau 4 Description des éléments du CPN de la figure 27.....81

Chapitre 1

Introduction générale

L'organisation a fait l'objet de plusieurs recherches scientifiques en économie, sociologie, science politique...etc. Une organisation fournit une structure de contraintes influant sur le comportement des agents (ensemble de rôles), qui peut focaliser leurs décisions et l'action de certains agents. [1]

Les concepts de sociétés méthodologiques et d'organisation ont beaucoup intéressé les chercheurs en informatique, notamment la réalisation des systèmes méthodologiques par l'application des systèmes multi-agent. Un système multi-agent peut être autrement vu qu'un système en interaction, mais aussi comme une organisation où chaque agent est un composant appartenant à cette organisation. La notion d'organisation a été adoptée dans les SMA comme un avantage, inspirée de l'organisation sociale ou humaine. Contrairement d'autres individus en interaction, sans aucune organisation, l'être humain a créé l'organisation pour être plus efficace et plus puissant. [2]

D'une manière générale, l'organisation est un modèle permettant aux agents de coordonner leurs actions au cours de la résolution d'une ou de plusieurs tâches. Elle définit d'une part, une structure (ex., une hiérarchie) comprenant un ensemble de rôles qui doivent être attribués aux agents et un ensemble de chemins de communication entre ces rôles. Elle définit d'autre part un régime de contrôle (ex., une relation maître/esclave) qui dicte le comportement social des agents. Enfin, elle définit des processus de coordination qui déterminent la décomposition des tâches en sous-tâches, l'allocation des sous-tâches aux agents, et la réalisation des tâches dépendantes de façon cohérente. [3]

Il est à noter qu'il existe deux points de vue au niveau organisationnel : ACPV (agent-centred point of view) et OCPV (organisation centred point of view). Si on prend les concepteurs du système de point de vue ACPV, ici ils s'intéressent principalement aux composants du système à construire à savoir les agents. En spécifiant et concevant des comportements locaux et des interactions pair à-pair adéquates, la fonctionnalité globale du système est le résultat des interactions complexes et des dynamiques au sein de la société d'agents. On parle donc d'une organisation émergente. Sinon d'un point de

vue OCPV les concepteurs spécifient, d'une part l'organisation et les schémas de coordination dans leur intégralité, et d'autre part les comportements des agents vis-à-vis de cette organisation. [4]

1.1 Contexte du projet

Notre travail peut être classé comme appartenant à l'ingénierie orientée agent AOSE (Agent Oriented Software Engineering). Correspondant à la communauté COIN (Coordination, Organisation, Institutions and Norms in agent systems). C'est la classe (c) dans la classification prise en considération dans la fig. 5 chapitre 2.

Puisqu'il est centré sur l'organisation OCPV (Organisation-Centred Point View), l'organisation qui est spécifiée dès le départ dans la phase de conception. Les agents au cours de leurs évolutions doivent respecter certaines contraintes (considéré comme obligatoire ou simple proposition), découlant de la définition même de l'organisation régissant leurs interactions.

L'organisation va donc servir de modèle au SMA adaptatif, pour la coordination des actions des agents composant le système dans un environnement ouvert et dynamique. Une adaptation réalisée par une nouvelle réorganisation des rôles aux agents du système.

1.2 Motivations

Dans un environnement dynamique (ex. le contrôle aérien, entreprise ...) les systèmes complexes sont appelés à interagir avec de multiples entités.

L'environnement devient de plus en plus complexe et dynamique ce qui rend difficile la tâche de programmer, de manière explicite, comment notre système doit-il répondre aux multiples événements imprévisibles.

L'interaction est la propriété la plus significative dans un système complexe, plus il y'a des composants qui interagissent avec l'environnement plus notre système est difficile à développer.

La disponibilité d'une grande capacité computationnelle a favorisé la délégation des tâches aux systèmes informatiques, se qui requière un haut niveau d'intelligence, un tel système doit impliquer :

- Habilité à agir de façon autonome (sans une intervention extérieure).
- Habilité à représenter de la meilleure façon l'interaction avec les humains ou autres système.

- Habilité à coopérer et à négocier avec d'autres systèmes.

Tout cela a mené à l'émergence de l'ingénierie orientée agent. Récemment il est devenu clair que les systèmes traditionnels ne sont pas appropriés pour interagir et évoluer dans un contexte dynamique. Implémenter l'interaction de manière explicite revient à gaspiller du temps et beaucoup d'efforts, cela nous ramène non seulement à développer des applications qui sont difficiles à implémenter, mais en plus qui sont difficiles à maintenir et à changer.

Dans certaines situations, il s'avère impossible que les systèmes traditionnels répondent aux besoins à cause de [5] :

- L'environnement du système est dynamique, rendant inopérant l'énumération exhaustive des situations que le système rencontrera.
- Le système est ouvert et donc dynamique car il est constitué d'un nombre variable de composants.
- La tâche à réaliser par le système est tellement complexe qu'une conception parfaite ne peut être garantie par les concepteurs.
- La manière de réaliser la tâche assignée au système est difficile, voire impossible, à appréhender dans sa globalité par le concepteur.

Par contre les SMA ont prouvé qu'ils sont très adaptés pour la conception d'applications complexes, interagissant avec un environnement en changement permanent.

Chaque agent est une entité computationnelle autonome. Capable de prendre en considération son contexte d'évolution, la communication avec d'autres agents comme lui. Produisant ainsi un système puissant par son encapsulation.

Un agent peut être défini en termes de buts, connaissances et sa capacité sociale. Après, il doit trouver, de manière autonome, sa propre fonction (rôle) dans son environnement de fonctionnement pour lequel il a été conçu. D'où toute l'importance à concevoir des SMA adaptatifs, applicables dans divers domaines (aéronautique, médical,...) pour la résolution de problèmes complexes dans un environnement dynamique et ouvert.

1.3 Problématique et objectif

1.3.1 Problématique

En ce qui concerne l'auto-organisation des rôles, il existe déjà des travaux dans ce sens dont on peut s'inspirer : les réseaux de Kohonen (Self Organization Map), l'équilibrage de charge, le modèle de fluide...etc. Après notre approche, doit être auto-adaptative. Nous allons justement prendre les cartes de Kohonen (Self Organization Map) comme modèle pour l'auto-organisation des tâches aux agents de notre système.

ACG : Agent Connection Graph $ACG = (A, E^A)$

A : Ensemble d'agent. $||A|| = m$.

E^A : Ensemble de connections entre agents.

$g(a, b)$: Délai de communication entre agents.

Le délai de communication peut être utilisé pour définir une distance métrique entre deux agents choisis aléatoirement.

$$d : A \times A \rightarrow R^+$$

Où «**d**» indique la plus courte distance métrique entre deux agents. Par analogie les tâches peuvent être modélisées par un graphe :

TIG : Task Interaction Graph $TIG = (T, E^T)$

T : Ensemble de tâches. $||T|| = n$.

E^T : Ensemble de communication entre tâches.

$a(i, j)$: Intensité de communication (échange).

$\pi : A \rightarrow P$: est une application de cartographie (organisation) qui vise à minimiser le cout de communication définie comme le temps de communication.

$$cc = \sum_{(i,j) \in E^T} a(i, j). d(\pi(i), \pi(j))$$

Tout en préservant les communications et les relations entre voisins, on a donc besoin d'une cartographie qui préserve la topologie et les réseaux de Kohonen (SOM) garantissent justement la préservation de la topologie «homeomorphism ». La cartographie par les réseaux de Kohonen peut être injective « une tâche pour chaque agent » ($m \geq n$), comme elle peut être contractive plusieurs tâches par agent ($m < n$) se qui requière dans le deuxième cas un équilibrage de charge. [6]

Les travaux de Heiss se base sur les cartes de Kohonen mais en élargissant la communication qui ne va plus se limiter, comme dans le cas des réseaux de Kohonen, seulement aux voisins directs. Rendant ainsi la communication possible entre tous les nœuds, avec un facteur inversement proportionnelle à la distance qui les éloignent. [6]

Ceci dit les résultats de Kohonen présentent des limites quand au manque d'adaptation. En effet les cartes de Kohonen restent figées dans le temps, alors que notre système multi-agents à une topologie dynamique : de nouvelles connexions entre agents qui naissent, d'autres qui disparaissent, un agent qui vient de quitter le système, d'autres qui arrivent... etc.

Mais notre approche, doit être capable d'adapter le système aux nouveaux changements par une nouvelle réorganisation de répartition des rôles aux agents du système. Une réassignation des rôles différente de la répartition initiale des rôles, en réponse aux nouveaux évènements qui peuvent surgir aux cours de l'évolution du système et son interaction avec l'environnement. L'organisation des agents n'est pas fixe mais se transforme suivant les besoins du système et sa confrontation à l'environnement, l'organisation est dite dynamique. L'intelligence donc de notre approche réside dans ca faculté d'auto-adaptabilité (elle est dite robuste) aux changements de contexte.

1.3.2. Objectif

Notre objectif est de réaliser une approche pour l'adaptation des systèmes multi-agents basés sur l'organisation des rôles. Une approche capable de répondre aux contraintes d'un environnement ouvert, dynamique et en constante évolution. Nous considérons une organisation préexistante installée dans le système multi-agent, qui doit être spécifiée durant la phase de conception (avant la phase d'implémentation).

Nous sommes dans le contexte de l'ingénierie de système méthodologique (AOSE methodology : Agent Oriented Software Engineering methodology). Le

but donc est de réaliser un système adaptatif et autonome. Un système qui sera capable de résoudre des problèmes auxquels les systèmes actuels n'arrivent pas à trouver de réponse, essentiellement dans un contexte ouvert et en changement permanent.

Pour cela, nous prenons pour base le concept d'adaptation du système multi-agent par une réorganisation d'assignation des rôles aux agents. L'organisation du système est décrite en termes d'affectation de rôles aux agents « carte d'affectation », la modification ou la réorganisation de la répartition des rôles aux agents est considérée comme un mécanisme d'adaptation.

Le changement de répartition des rôles des agents composant un SMA (changement des rôles qu'un agent assume), implique le changement de sa tâche et par conséquent son adaptation. L'adaptation permet de mettre en œuvre une nouvelle organisation (qui est la tâche du SMA). Nous prenons comme moyen technique les systèmes multi-agents basés organisation (auto-organisation) des rôles.

La fonction globale du système est une fonction émergente guidée par les contraintes d'organisation. Une organisation du SMA implique une fonction globale de ce SMA, un changement d'organisation implique donc un changement de la fonction globale du SMA. Le résultat est un système qui s'auto-organise par ces parties (auto organisation).

1.4. Organisation du mémoire

Ce mémoire s'organise en cinq (5) chapitres comme suit :

Chapitre 2 'Les systèmes multi-agents' nous définissons les notions d'agents et de systèmes multi-agents, leurs origines et la classification des organisations des systèmes multi-agents.

Chapitres 3 'Organisation multi-agent et réorganisation' nous abordons la définition de l'organisation dans les différentes littératures et domaines : sociologie, science de gestion et en fin dans les systèmes multi-agents. Nous nous intéressons après à la problématique de réorganisation, comme moyen d'adaptation, dans les systèmes multi-agents basé organisation des rôles. Finalement nous examinons les méthodes existantes décrivant l'auto-organisation de société d'agent.

Chapitre 4 'Approches pour la conception des systèmes multi-agent adaptatifs' cela consiste d'abord en une exploration horizontale des approches les plus pertinentes qui portent sur la réalisation des systèmes multi-agent adaptatifs ('Modification de la structure relationnelle' on peut citer le travail TSE de Kota et al. 'Interaction humaine pour le choix de la décision d'adaptation' on peut citer le travail d'Eric Matson et al. 'La séparation entre le code opérationnel et fonctionnel de l'agent' il y'a Danny weyns et al qui ont proposés l'approche MACODO. 'Allocation et réallocation des rôles' on trouve le travail T-MAS de Centeno et al et, le travail OMACS de Scott A. DeLoach et al et ceux de Juan M. Alberola et al). Puis une exploration verticale (plus détaillée et plus approfondie). Nous avons également proposé une classification des différents travaux en se basant sur quelques critères de comparaison.

Chapitre 5 'Problème posé et solution proposée' ou après avoir expliqué la problématique et les différentes approches existantes, nous présentons notre contribution avec une application pour la validation de notre approche.

Pour la phase de conception nous avons choisi la méthodologie MASA-Method [65] réalisée par Dr LAHLOUHI comme processus méthodologique. C'est une méthodologie complète selon la classification proposée dans [66]. Cette classification est basée sur un processus de développement de SMA comprenant d'autres étapes que celle d'implémentation des SMA.

Chapitre 6 'Mise en œuvre' nous donnons la justification du choix de l'environnement d'implémentation, aussi des explications plus techniques son présentées. En fin nous mettons en évidence d'une façon plus claire l'adaptation dans notre environnement.

La conclusion mettra en avant les avantages et les perspectives de notre approche.

Chapitre 2

Les systèmes multi-agent

2.1. Définition d'un agent

[7] 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,
- ✓ 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,
- ✓ 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.

Chacun des termes de cette définition est important. Une entité physique est quelque chose qui agit dans le monde réel : un robot, un avion ou une voiture sont des exemples d'entités physiques. En revanche, un composant logiciel, un module informatique sont des entités virtuelles, car elles n'existent pas physiquement. Les agents sont capables d'agir, et non pas seulement de raisonner comme dans les systèmes d'IA classique. L'action, qui est un concept fondamental pour les systèmes multi-agents, repose sur le fait que les agents accomplissent des actions qui vont modifier l'environnement des agents et donc leurs prises de décision futures. Ils peuvent aussi communiquer entre eux, et c'est d'ailleurs la l'un des modes principaux d'interaction existant entre les agents. Ils agissent dans un environnement, sauf, comme nous le verrons, pour les agents purement communicants pour lesquels toutes les actions se résument à des communications. Les agents sont doués d'autonomie. Cela signifie qu'ils ne sont pas dirigés par des commandes venant de l'utilisateur (ou d'un autre agent), mais par un ensemble de tendances qui peuvent prendre la forme de buts individuels à satisfaire ou de

fonctions de satisfaction ou de survie que l'agent cherche à optimiser. On pourrait dire ainsi que le moteur d'un agent, c'est lui-même. C'est lui qui est actif. Il a la possibilité de répondre par l'affirmative ou le refus à des requêtes provenant des autres agents. Il dispose donc d'une certaine liberté de manœuvre, ce qui le différencie de tous les concepts semblables, qu'ils s'appellent "objets", "modules logiciels" ou "processus". Mais l'autonomie n'est pas seulement comportementale, elle porte aussi sur les ressources. Pour agir, l'agent a besoin d'un certain nombre de ressources : énergie, CPU, quantité de mémoire, accès à certaines sources d'informations, etc. Ces ressources sont à la fois ce qui rend l'agent non seulement dépendant de son environnement, mais aussi, en étant capable de gérer ces ressources, ce qui lui donne une certaine indépendance vis-à-vis de lui.

L'agent est ainsi à la fois un système ouvert (il a besoin d'éléments qui lui sont extérieurs pour survivre) et un système fermé (car les échanges qu'il a avec l'extérieur sont très étroitement réglementés).

Les agents n'ont qu'une représentation partielle de leur environnement, c'est-à-dire qu'ils n'ont pas de vision globale de tout ce qui se passe. C'est d'ailleurs ce qui se passe dans les réalisations humaines d'envergure (la fabrication d'un Airbus par exemple) dans lesquelles personne ne connaît tous les détails de la réalisation, chaque spécialiste n'ayant qu'une vue partielle correspondant à son domaine de compétence. L'agent est ainsi une sorte "d'organisme vivant" dont le comportement, qui se résume à communiquer, à agir et, éventuellement, à se reproduire, vise à la satisfaction de ses besoins et de ses objectifs à partir de tous les autres éléments (perceptions, représentations, actions, communications et ressources) dont il dispose.

Récemment, Jennings, Sycara et Wooldridge [8] 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 sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne;
- *flexible* : l'agent dans ce cas est :
 - ✓ *capable de répondre à temps* : l'agent doit être capable de percevoir son environnement et 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 "bon" moment;
 - ✓ *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 ces agents à accomplir les leurs.



Fig.1 Représentation d'un agent en interaction avec son environnement.

2.2 Définition d'un système multi-agent

Un système multi-agent est un système distribué composé d'un ensemble d'agents. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités du raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de *coopération*, de *concurrence* ou de *coexistence*. [9] [10] [11]

D'après Ferber [JFE, 1995] On appelle système multi-agent (ou SMA), un système composée des éléments suivants:

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

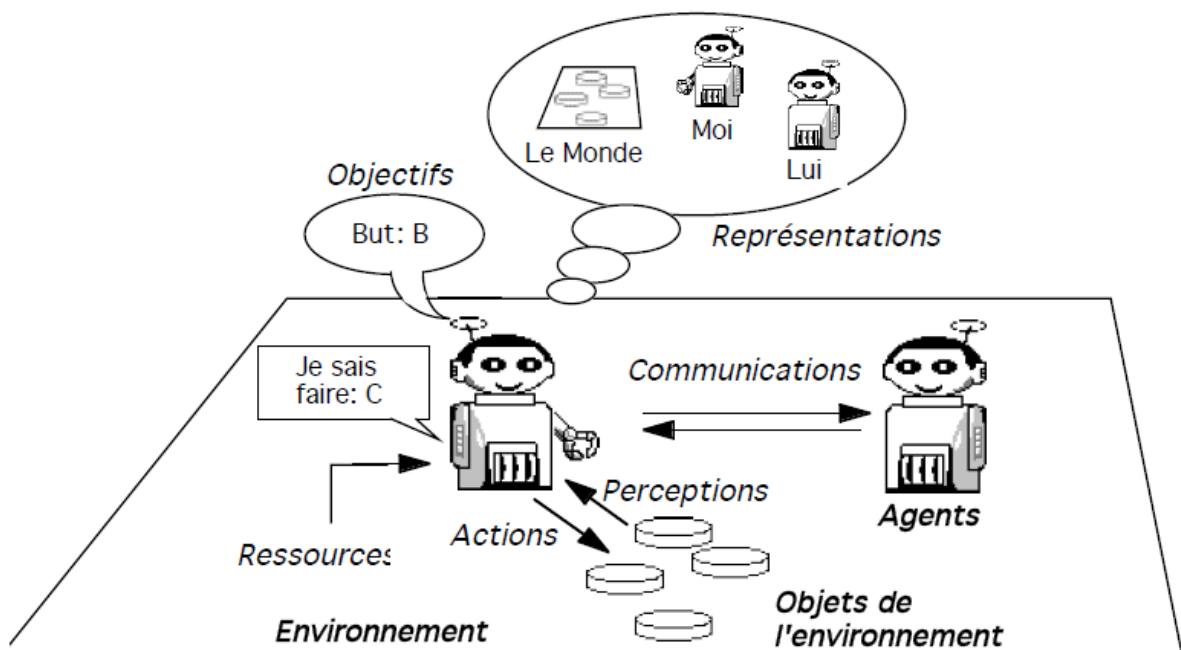


Fig.2 Représentation imagée d'un agent en interaction avec son environnement et les autres agents

Un SMA est généralement caractérisé par :

1. chaque agent à des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent à un point de vue partiel;
2. il n'y a aucun contrôle global du système multi-agent;
3. les données sont décentralisées;
4. le calcul est asynchrone.

Très courants en intelligence artificielle distribuée. Leur domaine de prédilection est la coopération de modules logiciels dont la fonction est de résoudre un problème ou d'élaborer une expertise (interprétation de signaux ou conception d'un produit par exemple) à partir de modules spécialisés, comme dans le cas d'un système de contrôle distribué, ou E est défini par la structure du réseau sous-jacent. Ces systèmes se caractérisent par le fait que les interactions sont essentiellement des communications intentionnelles, et que le mode de travail ressemble à celui d'un organisme social (groupe de travail, entreprise, administration, etc.).

Lorsque les agents sont situés, E est généralement un espace métrique, et les agents sont capables de percevoir leur environnement, c'est-à-dire de reconnaître les objets situés dans l'environnement en fonction de leurs capacités perceptives, et d'agir, c'est-à-dire de transformer l'état du système en modifiant les positions et les relations existant entre les objets.

Nous verrons que la plupart des systèmes multi-agents réactifs considèrent que la notion d'environnement est fondamentale pour la coordination des actions entre plusieurs agents. Par exemple, dans un univers de robots, les agents A sont les robots, E est l'espace géométrique euclidien dans lequel se meuvent les robots et O se compose évidemment des agents, mais aussi de l'ensemble des objets physiques placés ici et là, et que les robots doivent éviter, prendre ou manipuler. Les opérations Op sont les actions que les robots peuvent faire en se déplaçant, en bougeant les autres objets ou en communiquant, et R est l'ensemble des relations qui unissent certains agents à d'autres, telles que des relations d'acointances (certains agents en connaissent d'autres) et les relations de communicabilité (les agents peuvent communiquer avec certains agents mais pas nécessairement avec tous).

De ce fait, la dualité agent/environnement est au cœur des systèmes multi-agents situés. En agissant à partir de ses perceptions de l'espace physique et

des communications directes qu'il reçoit, l'agent se définit comme l'image duale de son environnement, c'est-à-dire par ce qui le distingue de ce qui l'entoure. Inversement, l'environnement d'un agent est caractérisé par tout ce qui n'est pas lui. Il est donc impossible de définir la notion d'agent situé indépendamment de celle d'environnement, les deux étant intrinsèquement liées et constituant deux aspects complémentaires d'un univers multi-agent. La réalisation d'un système multi-agent situé passe ainsi par la définition simultanée de la structure des agents et de celle de leur environnement, les actions de ceux-là devant s'exécuter au sein de celui-ci. Lorsque les agents sont purement communicants ou purement situés, la définition que nous avons donnée des systèmes multi-agents peut être adaptée.

2.3 Origines

- Intelligence artificielle classique : modélisation du comportement d'une seule entité « intelligente »
- Intelligence artificielle distribuée (IAD)
 - ✓ 1973 : Tableaux Noirs (Erman, système HEARSAY II)
 - ✓ Construction d'une solution commune à un problème
 - ✓ Centralisation des informations
 - ✓ Entités opportunistes de traitement
 - ✓ Evolué avec l'introduction de contrôles
 - ✓ 1977 : Acteurs de Hewitt
 - ✓ Fondé sur des entités communicantes (réception et envoi de messages)
 - ✓ 1978 : Première définition d'agents autonomes (Minsky)
- Systèmes multi-agents.

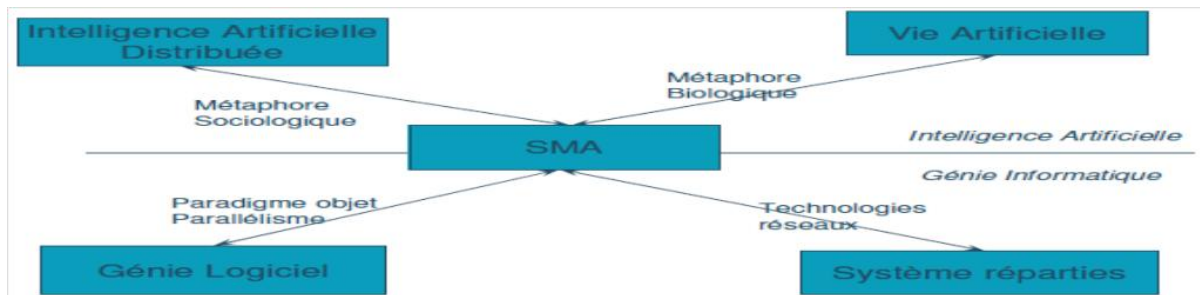


Fig.3 positionnement des SMA.

2.4 Classification d'organisation

Nous allons considérer la classification de ferber En reprenant la classification proposée par G. Gurvitch (Gurvitch 1963), maintenant traditionnelle en sociologie (Rocher 1968), on peut distinguer trois niveaux d'organisation dans les systèmes multi-agents:

1. Le niveau microsocial, où l'on s'intéresse essentiellement aux interactions entre agents et aux différentes formes de liaison qui existent entre deux ou un petit nombre d'agents. C'est à ce niveau que la plupart des études ont été généralement entreprises en intelligence artificielle distribuée.
2. Le niveau des groupes, où l'on s'intéresse aux structures intermédiaires qui interviennent dans la composition d'une organisation plus complète. A ce niveau, on étudie les différenciations des rôles et des activités des agents, l'émergence de structures organisatrices entre agents et le problème général de l'agrégation des agents lors de la constitution d'organisations.
3. Le niveau des sociétés globales (ou populations) où l'intérêt se porte surtout sur la dynamique d'un grand nombre d'agents, ainsi que sur la structure générale du système et son évolution. Les recherches se situant dans le cadre de la vie artificielle se situent assez souvent à ce niveau.

Cette classification, suggère notamment qu'il est possible d'analyser et de concevoir des organisations artificielles, soit à partir d'une spécification globale de la société que l'on veut obtenir et des propriétés recherchées, soit

au contraire à partir de la définition fine des agents, la société finale étant la conséquence - recherchée ou non - des interactions entre les agents.

La première approche est descendante et caractéristique d'une démarche d'ingénieur cherchant à obtenir un système qui réponde à un besoin. La seconde est plus expérimentale: elle s'accommode mieux d'un travail exploratoire dans lequel on tente d'obtenir des propriétés émergentes.

Le travail portant sur les organisations artificielles se situe donc au centre d'une dualité irréductible agent/organisation : toute organisation est le résultat d'une interaction entre agents, et le comportement des agents est contraint par l'ensemble des structures organisatrices, comme le montre la figure 4.

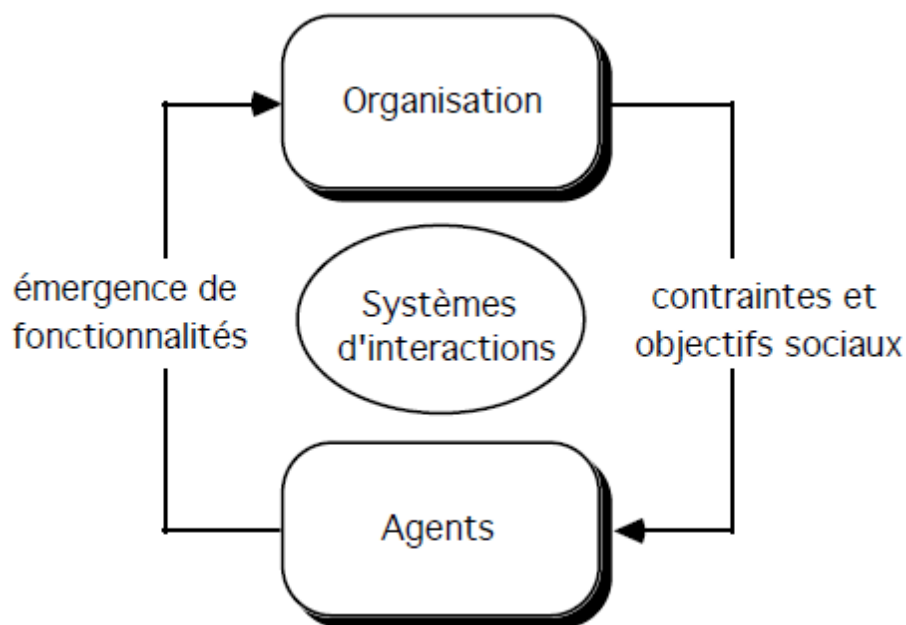


Fig.4 La relation micro-macro dans les systèmes multi-agents.

De ce fait, une société n'est pas nécessairement une entité "donnée", c'est-à-dire un objet préexistant à l'action de ses membres. Si des organisations telles que des institutions publiques ou des entreprises possèdent une individualité propre qui se distingue de l'ensemble de celles de ses membres, il n'en est pas nécessairement ainsi de structures collectives plus simples tels que des groupes de travail ou des hordes d'animaux. Même des sociétés réputées complexes, telles que des colonies d'abeilles ou de fourmis, ne doivent pas nécessairement être considérées comme des individus à part entière si l'on veut en comprendre l'organisation et les phénomènes de régulation et d'évolution qui y règnent. En termes de systèmes multi-agents, cela signifie qu'une organisation peut émerger de la rencontre d'actions

individuelles, sans qu'il soit nécessaire de définir un objet particulier (un élément de l'ensemble O) qui le représente.

Le rapport qui existe entre agent et organisation est donc une réalité mouvante. Sauf dans les cas où l'organisation est décrite intégralement par le concepteur du système, la répartition du travail, la coordination des actions et l'assignation des rôles résultent des activités de ses membres.

Une organisation n'est donc pas seulement une structure statique telle qu'on peut l'analyser si l'on considère un système multi-agent à un moment donné, mais un processus organisateur dont la structure n'est qu'un effet résultant. Cette conception rejoint celle de M. Crozier et E. Friedberg sur les organisations humaines (Crozier et Friedberg 1977), où ces dernières sont considérées comme procédant des systèmes d'actions des acteurs de ces organisations, c'est-à-dire des stratégies de prises de pouvoir, de repli et de négociations définissant des "ordres locaux" de régulation (Friedberg 1993).

Entre les individus qui en sont les acteurs et les organisations qui en résultent, il existe donc des processus essentiels par lesquels cette dualité s'exprime. Il s'agit des phénomènes d'interaction dont la logique dépend à la fois des capacités des agents, mais aussi des caractéristiques et des contraintes du système dans son ensemble. C'est d'ailleurs en mettant l'accent sur les interactions et, plus exactement, en analysant les systèmes d'interactions qui existent entre les agents que les systèmes multi-agents se distinguent des considérations systémiques ou sociologiques plus classiques.

En effet, alors que nombre d'approches considèrent les systèmes d'un point de vue global, holiste même, et dont l'analyse procède de manière descendante, les systèmes multi-agents prennent le parti de l'émergence, c'est-à-dire de l'action et de l'interaction, comme élément moteur de la structuration de la société dans son ensemble. Les systèmes d'interactions constituent ainsi l'objet d'étude essentiel des systèmes multi-agents et servent à articuler le rapport qui existe entre l'agent d'une part et l'organisation dans sa globalité d'autre part. [7]

Il existe donc deux points de vue au niveau organisationnel :

- 1) ACPV (Agent-Centred Point of View) → correspond à la communauté SASO (Self-Adaptive and Self-Organizing systems), qui étudie les organisations sous l'angle phénomène émergent dans les systèmes complexes.

- 2) OCPV (Organisation-Centred Point View) → correspond à la communauté COIN (Coordination, Organisation, Institutions and Norms in agent systems), qui s'intéresse à l'ingénierie des mécanismes effectifs de coordination et de régulation dans le cadre de la conception des SMA ouvert et complexes.

Si on prend les concepteurs du système de point de vue ACPV, ici ils s'intéressent principalement aux composants du système à construire à savoir les agents. En spécifiant et concevant des comportements locaux et des interactions pair-à-pair adéquats, la fonctionnalité globale du système est le résultat des interactions complexes et des dynamiques au sein de la société d'agents. On parle donc d'une organisation émergente et des phénomènes ascendants regroupés sous le terme général d'auto organisation. [4]

Sinon d'un point de vue OCPV les concepteurs spécifient d'une part l'organisation et les schémas de coordination dans leur intégralité, et d'autre part les comportements des agents vis-à-vis de cette organisation. Les agents peuvent être considérés les contraintes définies par l'organisation, comme de simple proposition ou, comme étant obligatoire pour la coordination de leurs actions (la spécification de l'organisation impose un certain comportement ou performance). [4]

Et dans les deux cas la notion d'**organisation dynamique** est capitale. A partir de la figure 1 suivante nous pouvons positionner quelques SMA et approches existantes

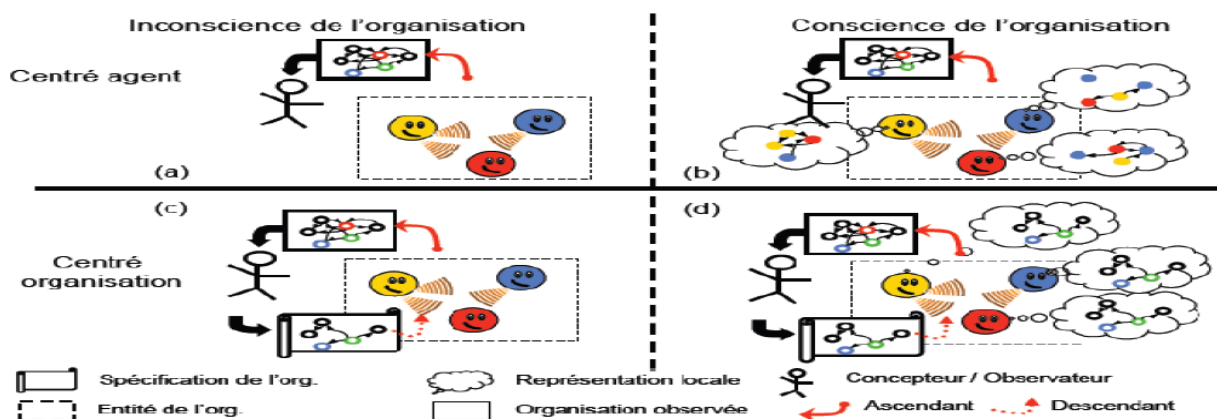


Fig.5 (a) SMA émergents ; (b) SMA basés sur les coalitions ; (c) Ingénierie orientée agent ; (d) SMA orientés organisation. Le concepteur/observateur peut être le développeur/utilisateur (cas exogène) ou un ensemble d'agents (cas endogène).

D'après les travaux de l'équipe « G. Picarda J. F. Hübnera O. Boissiera M.-P. Gleizesb » [4] en considérant l'architecture des agents, nous pouvons affiner ces deux points de vue en suivant l'axe orthogonal sur la capacité des agents à se représenter et à raisonner sur l'organisation. Dans la première colonne de la Fig.5, les agents ne connaissent rien à propos de l'organisation.

- Dans le cas (a) les agents ne représentent pas l'organisation, bien que l'observateur puisse identifier une organisation émergeant des interactions entre agents. Dans un sens, ils ne sont pas conscients d'appartenir à une organisation. Ils peuvent néanmoins avoir une représentation locale, interne et partielle des autres agents.
- Dans le cas (c), l'organisation existe comme un schéma spécifié et formalisé, construit par le concepteur des agents, mais les agents ne savent rien de cette organisation et ne peuvent donc raisonner sur ce sujet. Ils s'y conforment simplement comme si les contraintes organisationnelles étaient codées en dur en eux – par exemple, le SMA résultant d'un déroulement d'une méthode de développement orientée agent où le code des agents est généré à partir de spécifications organisationnelles. [12]

Dans la seconde colonne, nous considérons les cas où les agents ont une certaine représentation ou conscience de l'organisation dans laquelle ils évoluent.

- Dans le cas (b), chaque agent possède une représentation des schémas de coopération qu'il suit afin de décider que faire – par exemple, les réseaux de dépendances pour la formation de coalitions [13]. Cette représentation locale est obtenue par les perceptions, les communications ou des raisonnements explicites (comme le raisonnement social dans [13]) car il n'y a, a priori, dans une vue centrée agent, aucune représentation globale explicite de l'organisation définie. Les agents sont capables de raisonner sur l'organisation et de l'utiliser afin d'initier des coopérations avec d'autres agents. Toutefois, les frontières entre ces différentes catégories sont floues et perméables. Dans la littérature, certaines approches organisationnelles pour les SMA correspondent à des cas de la Fig.5, alors que d'autres s'étendent sur plusieurs cases.

Les propositions concernant les approches de réorganisation pour les organisations formelles peuvent combiner les cas (b) et (d) dans le sens où les

agents utilisent leurs propres mécanismes internes pour adapter l'organisation imposée par le système. Les manipulations ascendantes ou descendantes d'une organisation peuvent être réalisées, soit de manière endogène (par les agents appartenant à l'organisation eux mêmes), ou de manière exogène (par une entité, humain ou agent, hors du système).

2.5 L'interaction

Pour un agent, interagir avec un autre constitue à la fois la source de sa puissance et l'origine de ses problèmes. C'est en effet parce qu'ils coopèrent que des agents peuvent accomplir plus que la somme de leurs actions, mais c'est aussi à cause de leur multitude qu'ils doivent coordonner leurs actions et résoudre des conflits. Pour un agent, l'autre est à la fois le pire et la meilleure des choses. Traiter le problème de l'interaction, c'est se donner les moyens non seulement de décrire les mécanismes élémentaires permettant aux agents d'interagir, mais aussi d'analyser et de concevoir les différentes formes d'interaction que des agents peuvent pratiquer pour accomplir leurs tâches et satisfaire leurs buts. Tout d'abord, les agents doivent être capables, par le biais de la communication, de transmettre des informations, mais surtout d'induire chez l'autre un comportement spécifique.

Communiquer est donc une forme d'action particulière qui, au lieu de s'appliquer à la transformation de l'environnement, tend à une modification de l'état mental du destinataire. Par exemple, demander à un autre d'exécuter une tâche tend à provoquer chez l'autre une intention d'accomplir cette tâche et constitue donc une manière de satisfaire un objectif sans réaliser la tâche soi-même. Les différentes formes d'interaction sont la collaboration et la coordination d'actions. La première s'intéresse à la manière de répartir le travail entre plusieurs agents, qu'il s'agisse de techniques centralisées ou distribuées, et la seconde analyse la manière dont les actions des différents agents doivent être organisées dans le temps et l'espace de manière à réaliser les objectifs. Enfin, lorsque des conflits apparaissent, il est important de pouvoir en limiter les effets.

Les techniques de négociation servent ainsi à satisfaire les parties impliquées en établissant des compromis ou en dépassant la nature du conflit. La coopération est la forme générale d'interaction la plus étudiée dans les systèmes multi-agents [7]. Contrairement au sens commun qui considère que la coopération demeure l'apanage des êtres capables d'avoir un projet explicite, donc des agents cognitifs, il est possible de parler de coopération de manière aussi bien réactive que cognitive si l'on envisage uniquement le

résultat des actions et non les intentions des agents. De manière simplifiée, le problème de la coopération peut se ramener à déterminer qui fait quoi, quand, ou, avec quels moyens, de quelle manière et avec qui, c'est-à-dire en fait à résoudre les différents sous-problèmes que constituent la collaboration par répartition de tâches, la coordination d'actions et la résolution de conflits. Nous résumerons cela par la formule:

$$\text{Coopération} = \text{collaboration} + \text{coordination d'actions} \\ + \text{Résolution de conflits}$$

2.6 L'adaptation

[7] On peut voir le problème de l'adaptation structurelle et comportementale d'un ensemble d'agents de deux manières différentes : soit comme une caractéristique individuelle des agents, et on parle alors d'apprentissage, soit comme un processus collectif mettant en jeu des mécanismes reproductifs, ce que l'on appelle évolution.

Bien que les deux approches soient généralement traitées de manière distincte, il est possible de les considérer globalement comme un processus à la fois individuel et collectif d'adaptation. Ces problématiques font l'objet de recherches importantes dans les domaines de l'intelligence artificielle distribuée et de la vie artificielle, le premier se consacrant plus naturellement à l'apprentissage, par des techniques symboliques ou neuronales, et le second à l'évolution, en utilisant des algorithmes génétiques.

Bien que l'adaptation soit un thème particulièrement important dans l'étude des systèmes multi-agents (Lestel et al. 1994), nous ne le développerons que très partiellement, en ne l'abordant qu'au travers des comportements collectifs, l'adaptation résultant alors de mécanismes de régulation issus des actions conjointes d'un ensemble d'agents.

Chapitre 3

Organisation multi-agent et réorganisation

Les concepts de sociétés méthodologiques et d'organisation ont beaucoup intéressé les chercheurs en informatique, notamment la réalisation des systèmes méthodologiques par l'application des systèmes multi-agents. L'organisation est, avec l'interaction, l'un des concepts de base des systèmes multi-agents. Lorsqu'on parle d'organisation, on suppose qu'il existe un ensemble d'entités formant une certaine unité, et dont les différents éléments sont subordonnés entre eux dans un ensemble solidaire et dans une activité convergente. L'organisation nécessite donc un certain ordre entre les entités éventuellement hétérogènes, lequel concourt à la cohérence du tout. [7]

3.1 Définition d'organisation multi-agent

L'organisation en Sociologie

Dans le domaine de la Sociologie, selon [14], les organisations doivent être vues comme des coalitions dynamiques d'intérêts divergents, et souvent conflictuels que comme des systèmes de collaboration parfaite.

Selon Morin [15], l'organisation peut être définie comme un agencement de relations entre composants ou individus qui produisent une unité, ou système, dotée de qualités inconnues au niveau des composants ou individus. L'organisation lie de façon interrelationnelle des éléments, événements ou individus divers, qui dès lors deviennent les composants d'un tout. Elle assure solidarité et solidité relative, donc assure au système une certaine possibilité de durée en dépit de perturbations aléatoires.

Définition en Sciences de Gestion

En Sciences Cognitives, on distingue l'organisation du travail et l'organisation sociale. L'organisation du travail s'appuie sur l'allocation d'une tâche particulière. Cette allocation se développe pour chaque situation, et guidée par les compétences des acteurs (ou agent) et la "technologie du domaine de travail". Elle détermine ainsi que le contenu de la communication inter-agents.

En revanche, l'organisation sociale s'appuie sur l'interaction sociale entre les agents. Elle dépend de la forme de la communication, qui à son tour dépend de la stratégie de la coordination adoptée [17] [18].

Organisation dans les SMA

Dans le domaine des systèmes multi-agents, plusieurs définitions ont également été données. Nous donnons ici, à titre d'exemple, les définitions suivantes :

L'organisation définit comme étant une structure décrivant les interactions et autres relations, qui existent (dans le but d'assouvir un objectif commun) entre les membres de la dite organisation [16] [17].

Dans un système multi-agent [7], il existe de nombreuses interrelations entre les agents, par le biais de délégation de tâches, de transfert d'informations, d'engagements, de synchronisations d'actions, etc. Ces interrelations ne sont possibles qu'à l'intérieur d'une organisation, mais réciproquement, les organisations exigent l'existence de ces interrelations. Les organisations constituent donc à la fois le support et la manière dont se passent ces interrelations, c'est-à-dire la façon dont sont réparties les tâches, les informations, les ressources, et la coordination d'actions. De plus, et c'est ce qui rend ce terme si difficile à cerner, l'organisation désigne à la fois le processus d'élaboration d'une structure et le résultat même de ce processus. Cette dualité du terme montre qu'il existe toujours un aspect dynamique dans l'organisation: il n'existe, à proprement parler, pas d'organisations statiques. L'organisation est nécessairement dynamique et toujours en réorganisation de l'ensemble des entités et des liens qui unissent ces entités.

Dans le sens commun du terme, l'organisation est à la fois la façon, la manière dont un ensemble est constitué en vue de son fonctionnement et le fait, l'action d'organiser. Dans le premier sens on se réfère à un ordre, un régime, une structure ; dans le second sens, l'idée sous-jacente est celle d'un processus, d'une dynamique. Il y a deux aspects dans ce seul terme : l'action et le résultat de l'action [17].

D'une manière générale, l'organisation est un modèle permettant aux agents de coordonner leurs actions au cours de la résolution d'une ou de plusieurs tâches. Elle définit d'une part, une structure (ex., une hiérarchie) comprenant un ensemble de rôles qui doivent être attribués aux agents, et un ensemble de chemins de communication entre ces rôles.

Elle définit d'autre part un régime de contrôle (ex., une relation maître/esclave) qui dicte le comportement social des agents. Enfin, elle définit des processus de coordination qui déterminent la décomposition des tâches en sous-tâches, l'allocation des sous-tâches aux agents, et la réalisation des tâches dépendantes de façon cohérente [3].

3.2 Structures organisationnelles et organisations concrètes

La structure organisationnelle est ce qui caractérise, sur un plan abstrait, une classe d'organisation. A l'inverse, l'organisation concrète (ou simplement organisation) est une instanciation (au sens des langages objets) possible d'une structure organisationnelle, une réalisation comprenant l'ensemble des entités qui participent effectivement à l'organisation, ainsi que l'ensemble des liens qui associent ces agents à un moment donné. A toute organisation concrète, il est possible d'associer une structure organisationnelle, mais une même structure organisationnelle peut servir de base à la définition d'une multitude d'organisations concrètes. En appliquant aux organisations la définition de P. Delattre (Delattre 1968) pour les structures, on peut dire qu'une structure organisationnelle est définie par la double donnée:

1. d'un ensemble de classes d'agents caractérisées par les rôles affectés aux agents,
2. de l'ensemble de relations abstraites existant entre ces rôles.

Enfin, on classera les structures organisationnelles en fonction de leur capacité à s'adapter et à évoluer en fonction des besoins. Pour obtenir une organisation concrète, il est nécessaire d'y ajouter l'ensemble des agents appartenant aux différentes classes, et de décrire les relations concrètes qui existent entre les agents, c'est-à-dire "d'instancier" l'organisation abstraite. La figure 6 illustre cette différence. La structure organisationnelle O est décrite par l'ensemble des rôles qui peuvent y être tenus (A, B, C), alors que les structures concrètes comprennent un grand nombre d'entités concrètes (A_1, A_2, B_1, \dots), les A_i étant associées au rôle A . Les liens entre les entités concrètes ($g_1, g_2, h_1, h_2, \dots$) correspondent aux liens abstraits qui existent entre les entités abstraites (g, h, \dots). Par exemple, le lien abstrait g qui lie l'entité abstraite A à B s'exprime concrètement dans l'organisation concrète O_1 sous la forme de deux liens g_1 et g_2 qui relient respectivement A_2 à B_1 et A_1 à B_1 , alors que dans O_2 des liens g_1 et g_2 semblables relient respectivement A_1 à B_1 et A_1 à B_2 .

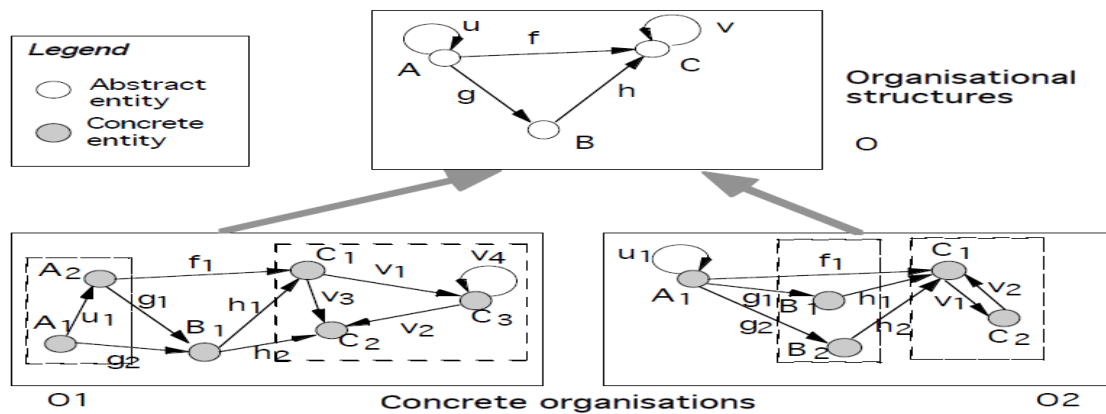


Fig.6 : Organisations et structures organisationnelles

3.3 Définition d'auto organisation

L'auto-organisation est un processus endogène ascendant, concernant les systèmes dans lesquels seules des informations et représentations locales sont manipulées par les agents, qui sont inconscients de l'état de l'organisation dans sa globalité. Afin d'adapter le système à la pression environnementale en modifiant indirectement l'organisation, donc en changeant directement la configuration du système (topologie, voisinages, influences, différenciation), ou l'environnement du système, par des interactions et propagations locales, en évitant le biais de modèles prédéfinis. [4]

3.4 Définition de la réorganisation

La réorganisation est un processus endogène ou exogène, concernant les systèmes dans lesquels l'organisation est explicitement manipulée au travers de spécifications, des contraintes ou autres moyens. Afin d'assurer un comportement global adéquat, lorsque l'organisation n'est pas adaptée. Les agents étant conscients de l'organisation, ils sont capables de manipuler des primitives afin de modifier leur environnement social. Ce processus peut-être à la fois initié par une entité externe au système ou par les agents eux-mêmes, en raisonnant directement sur l'organisation (rôles, spécification organisationnelle), et sur les schémas de coopération (dépendances, engagements, pouvoirs). [4]

[19] « Un système auto-organisateur est un système qui change sa structure de base comme fonction de son expérience et de son environnement. »

[20] La réorganisation est une description d'un comportement, elle a une valeur heuristique et elle permet d'indiquer un phénomène. Elle est

condamnée à rester une simple description, tant qu'on ne se préoccupe pas de rechercher le mécanisme qui est à son origine.

[21]La réorganisation est un processus où l'organisation (contrainte, redondante) d'un système croît de manière spontanée, i.e. sans que cet accroissement soit contrôlé par l'environnement ou ce qui l'entoure ou encore un système externe.

[22]La réorganisation définit la propriété d'un système qui s'organise ou se réorganise dans le temps, pour former des structures sémantiquement remarquables.

[23]L'auto-organisation caractérise tout processus au cours duquel des structures émergent au niveau collectif (ou plus généralement apparition d'une structure à l'échelle N+1 à partir d'une dynamique définie à l'échelle N), à partir de la multitude des interactions entre individus, sans être codées explicitement au niveau individuel.

[17] l'auto-organisation correspond à une réorganisation décidée de manière autonome par les agents au sein du système. L'auto-organisation d'un système s'est produite si le système a changé de structure.

[24]L'auto-organisation d'un système, consiste en la transformation de la topologie (c'est-à-dire des connections du réseau) de ses parties en tant que résultat du fonctionnement de ce même réseau, dans le cadre du couplage structurel avec l'environnement. Dans un système auto-organisateur, les règles d'organisation sont intérieures au système, qui apparaît informationnellement clos. Les systèmes auto-organiseurs appartiennent à la classe des systèmes autonomes (systèmes spécifiés par des mécanismes internes d'auto-organisation) et non pas hétéronomes (définis par des mécanismes extérieurs de contrôle).

La réorganisation, est un des moyens pour qu'un SMA (Système Multi-Agent) parvienne à surmonter les perturbations éventuelles de l'environnement et pour réaliser des systèmes adaptatifs. Elle consiste [69] à passer d'une organisation à une autre. Dans le cas où l'organisation s'avère n'est pas la plus appropriée, le système procède à une réorganisation de répartition des rôles, et passe à une nouvelle organisation plus adéquate qui améliorera la situation. Ce passage est réalisé par le SMA d'une manière autonome sans intervention externe, donc un système qui s'auto-organise par ces parties (auto-organisation).

3.5 processus de réorganisation

De manière générale, afin de modifier une organisation, nous pouvons identifier un processus générique qui sera implémenté différemment en fonction de l'approche suivie. Ce processus est normalement composé de deux phases – la surveillance (ou monitoring), et la réparation.

Cette dernière phase suivant le type de SMA, peut être constituée des phases de conception (ou design), de sélection et d'exécution (ou d'implémentation) [70] ou uniquement des phases de sélection et d'exécution. Il peut s'inscrire dans un bouclage environnement-système : perception, processus de changement, action, et ainsi de suite.

1. La phase de surveillance ou monitoring est destinée à détecter les problèmes d'inadéquation entre le système, l'organisation ou les agents, et l'environnement. Quelle que soit l'entité responsable de cette surveillance (agent, organisation, entité externe), et donc quel que soit le niveau d'abstraction de cette détection (niveau macro ou niveau micro), il convient de définir des situations de non adaptation.
2. Une fois qu'un besoin de modification est détecté lors du monitoring, la phase de réparation met en œuvre un processus pour que l'organisation retrouve un fonctionnement le plus optimal possible et ce, en cours de fonctionnement du système.
3. Pour cela, la phase design vise à définir et à développer un ensemble d'alternatives possibles pour l'organisation courante, de manière ascendante ou descendante.
4. La phase de sélection détermine une des alternatives pour modifier l'organisation. Le principal problème est donc la définition de critères d'évaluation des propositions les plus prometteuses.
5. La phase d'exécution ou d'implémentation correspond à la mise en œuvre de l'alternative choisie précédemment.

3.6 Quelques travaux de réorganisation

D'après la classification découlant de la fig.5 et qui repose sur deux principales visions qui sont : centré organisation et centré agent, nous allons positionner quelques travaux :

a) SMA a organisation émergente

Y'a pas une spécification explicite de l'organisation donc les agents n'ont pas une représentation locale de celle-ci et ne peuvent raisonner sur puisqu'elle n'est pas modélisée. Ce sont généralement les travaux des communautés SASO : AMAS, ADELFE. [71]

AMAS (Adaptative Multi Agent Systems)

La théorie des AMAS vise à réaliser un système complexe adaptatif pour répondre au changement d'un environnement en constante évolution, elle aborde trois aspects :

1. L'apprentissage.
2. L'auto-organisation.
3. L'émergence.

L'approche adoptée dans cette théorie est l'adaptation par auto-organisation prenant comme moyen technique les SMA. La fonction globale n'étant pas spécifiée elle émerge durant la progression du système, et l'apprentissage se fait par auto-organisation des agents coopératifs. Donc l'organisation qui émerge est observable elle n'est pas définie par le concepteur.

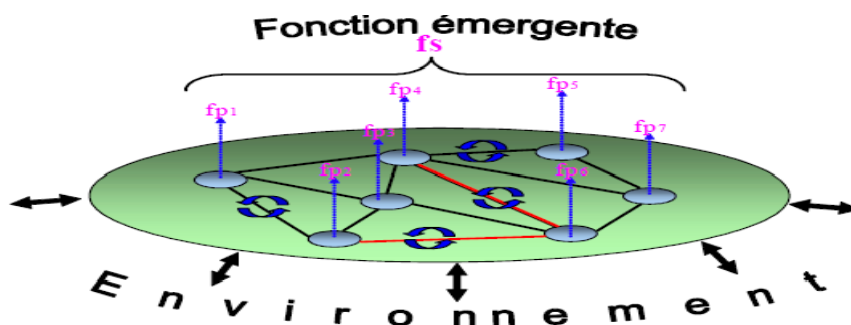


Fig.7 adaptation par auto-organisation d'un système en interaction dynamique

La théorie des AMAS, expose le théorème d'adéquation fonctionnelle de la fonction globale émergente qui permet au système de porter un jugement sur son propre comportement dans l'environnement, et le système doit changer d'organisation dès qu'il n'est plus en adéquation fonctionnelle.

La démonstration de ce théorème se déduit par l'application d'un axiome et de quatre lemmes. [5]

Aussi la technologie des AMAS se base sur la définition de non coopération idéale comme condition conduisant au déclenchement du processus de réorganisation :

$$\text{Non coopération} = \neg C1 \vee \neg C2 \vee \neg C3$$

Ou :

$\neg C1$: un signal perçus et incompris ou ambigu.

$\neg C2$: l'information reçu est inconnue ou n'a aucune conséquence logique.

$\neg C3$: compte tenu de ces croyances locales, l'agent considère la transformation comme pas bénéfique pour autrui.

b) SMA basés sur la coalition

Y'a toujours pas de spécification explicite de l'organisation, les agents ne peuvent pas raisonner sur l'organisation globale mais néanmoins ils ont conscience de l'organisation. Ils possèdent leurs propres modèles locaux d'interaction et des relations de dépendances avec leurs voisinage. Cela en respectant des règles sociales et des schémas prédéfinis.

Exemple d'une telle approche SMA basé sur le contrat net Protocol [25], où les agents au cours des interactions et des expériences, se construisent des représentations des autres agents et l'organisation globale sera la somme des représentations locales. De manière plus élaborée, les coalitions multi-agent [13], raisonne directement sur des schémas de coopération comme exemple Cassiopée, tropes sont des rare méthodes ascendantes.

Cassiopée

C'est une méthode ascendante (bottomup), dans laquelle la structure globale et l'organisation des rôles sont définis à partir des actions nécessaires à l'obtention d'une tâche globale. Cette méthode a été développée par Collinot et Drogoul [26] [27].

Les concepts clés

Rôle, agent, dépendances et groupes, un agent est vu comme un ensemble de rôle organisés en trois niveaux :

- Rôle individuel : qui sont les comportements que les agents sont individuellement capables de mettre en œuvre, indépendamment de la façon dont ils le font.
- Rôle relationnel : c.-à-d. comment ils choisissent d'interagir avec les autres (en activant/- désactivant les rôles individuels), en se basant sur les dépendances mutuelles qu'entretiennent les différents rôles individuels.
- Rôle organisationnel : ou comment les agents peuvent gérer leurs interactions pour devenir ou rester organisés (en activant/désactivant leurs rôles relationnels).

Le processus

Se décompose en cinq phases qui alternent entre le point de vue local (agent) et le point de vue global (organisation).

- *Calque des rôles individuels*, qui définit les différents rôles d'agents, ou tâches, nécessaires, et permettent de définir les différents types d'agents.
- *Calque des dépendances*, qui définit les dépendances entre rôles, grâce à des graphes de dépendances.
- *Calque des rôles relationnels*, qui définit la façon dont les agents gèrent ces dépendances, en jouant les rôles relationnels adéquats.
- *Calque des groupes*, qui définit les groupes pouvant apparaître en cours de résolution.
- *Calque des rôles organisationnels*, qui décrit la dynamique de ces groupes, c.-à- d. les rôles organisationnels que les agents peuvent jouer pour les faire apparaître, évoluer ou disparaître.

Ce processus se limite donc aux phases d'analyse, et au début de la conception, si on considère la définition des états internes comme appartenant à la conception (dans les processus classiques orientés objet, c'est le cas).

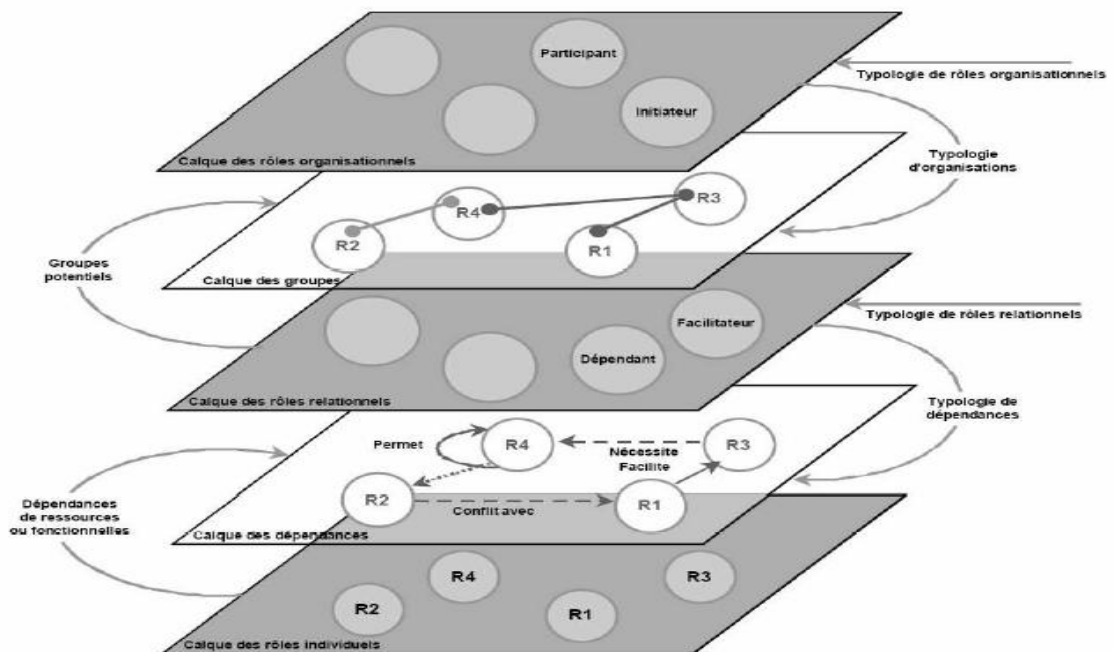


Fig.8 Phases du processus de la méthode Cassiopeia.

c) Ingenierie orienté agent

L'organisation, est spécifiée durant la phase de conception bien avant l'implémentation. Les travaux de la communauté OASE propose plusieurs méthodes : MaSE, INGENIAS, ASPECT, GAIA, MASA method.

GAIA

Gaia est une extension des approches d'ingénierie logicielle classique. C'est une méthode plus complète et bénéficie, de plus, d'une large reconnaissance dans le domaine multi-agent. Il existe deux versions de GAIA méthodologie qui sont différente dans l'organisation d'abstraction, concepts et modèles. La deuxième version augmente l'ancienne par un certain nombre d'organisation d'abstraction nouvelle. [28] [29]

Gaia se veut être générale et applicable à n'importe quel domaine, et compréhensible par la distinction entre macro-niveau et micro-niveau. Les agents modélisés, pouvant être hétérogènes, sont des systèmes computationnels à gros grain, qui vont essayer de maximiser une mesure de qualité globale. Toutefois, Gaia ne prend pas en compte les systèmes admettant de réels conflits. L'organisation (d'une centaine d'agents environ) est clairement statique dans le temps, de même que les services offerts par les agents. [30]

Gaia manipule six modèles d'analyse et de conception différents :

- *modèle de rôle*
- *modèle d'interaction*
- *modèle d'agent*
- *modèle de service*
- *modèle d'organisation* ou d'acointances
- *modèle environnemental*

Gaia rend facile la manipulation d'un grand nombre de concepts multi-agents, grâce à ces modèles, ce qui a certainement fait sa popularité. La propriété d'autonomie de contrôle ou de rôle des agents est exprimée par le fait qu'un rôle encapsule sa fonctionnalité, qui est interne et non affectée par l'environnement. La réactivité et la pro-activité sont, elles, plus ou moins exprimées grâce aux propriétés de vivacité des responsabilités. On peut regretter que l'environnement ne soit défini que comme une liste de variables caractérisées en lecture ou écriture. Enfin, les notions sociales sont abordées dans le modèle organisationnel d'acointances.

Le processus

La figure 9 montre le processus de développement proposé dans Gaia. Trois phases sont principalement abordées : l'analyse, la conception architecturale et la conception détaillée. Lors de l'analyse, le système se voit divisé en sous-organisations, dont vont découler les modèles d'environnement, de rôle et d'interaction (préliminaires). A partir de ces modèles, l'analyse doit spécifier les règles organisationnelles (permissions,

vivacité et sûreté) de dépendances entre rôles. La conception architecturale doit aboutir au raffinement des modèles de rôle et d'interaction par l'analyse des structures organisationnelles. Enfin, lors de la conception détaillée, les modèles d'agent (détermination des types et instances d'agents) et de services sont implantés. Ce processus est donc très limité, et se focalise uniquement sur les premières phases de conception. Il n'y a donc pas de phase de vérification/validation.

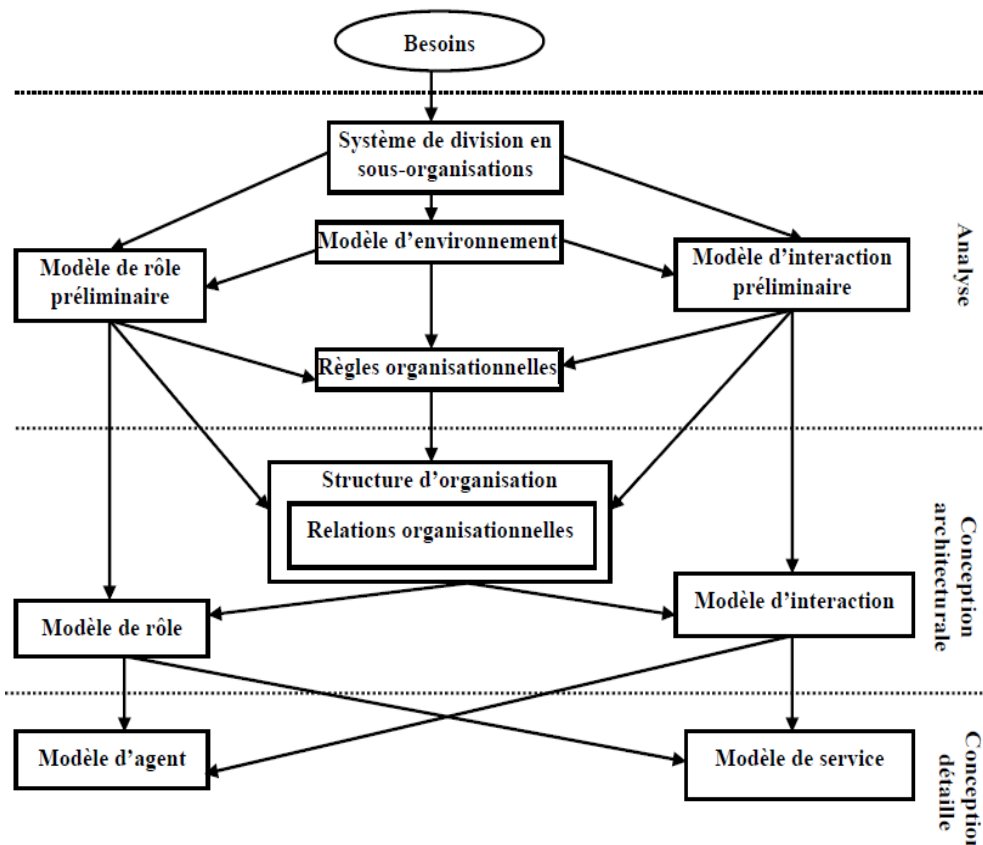


Fig.9 Le processus de développement de Gaia et les modèles manipulés.

d) SMA orientés organisation

En même temps l'organisation existe et elle est bien spécifiée par le concepteur, les agents aussi ont leur propre représentation de l'organisation et peuvent la modifier. Nous citons les exemples suivants : AGR [35], TAEMS [32], STEAM [31], MOISE+ [33], ISLANDER [34].

MOISE+

Inspirée de deux types d'approches globales des SMA.

- 1) l'approche basée sur des plans globaux, s'intéressant à l'allocation des tâches aux agents, a la coordination pour exécuter un plan.
- 2) Les approches basées sur la notion de rôle et de structure organisationnelle.

La structure organisationnelle est décrite en trois volets :

Spécification structurelle, spécification fonctionnelle et la spécification déontique.

Spécification structurelle

La spécification structurelle est constituée d'un ensemble de rôles, de liens entre ces rôles et de groupes.

- **Les rôles** : ne sont à ce stade que des étiquettes. Ils peuvent être structurés par une relation d'héritage.
- **Les liens** : peuvent être de trois types : lien d'acointance (autorisant la représentation d'un agent par un autre), lien de communication (autorisant bien sûr la communication) et lien d'autorité (autorisant le contrôle d'un agent par un autre).
- **Les groupes** : sont constitués de rôles et de sous-groupes. A cela s'ajoutent des informations sur la cardinalité des rôles et des sous-groupes, une relation de compatibilité entre les rôles (indiquant si deux rôles peuvent être joués par le même agent), et une donnée sur la portée des liens (inter- ou intra-groupe).

Spécification fonctionnelle

La spécification fonctionnelle est constituée de Schémas Sociaux. Un tel schéma est un ensemble de plans globaux, organisés en missions.

- **Un plan global** : est une arborescence de buts collectifs, décomposés en un arbre admettant trois types de décomposition : séquentielle, sélective ou parallèle.
- **Une mission** : est un ensemble de buts contenus dans un même plan global. Chaque mission possède une cardinalité décrivant combien d'agents peuvent l'accomplir simultanément.

Spécification déontique

Elle établit le lien entre les deux premiers types de spécification. Elle est formée de permissions et d'obligations, qui spécifient les missions sur lesquelles un agent prenant un rôle donné peut ou doit s'engager, respectivement.

Apports et limites du modèle

MOISE+ propose un formalisme plus riche que les autres approches présentées ci-dessus. Il prend en compte des aspects importants, notamment : l'héritage entre rôles, les différents types de liens, et la représentation explicite d'une hiérarchie de buts collectifs. Ce modèle nous semble néanmoins souffrir de quelques imperfections [36] :

- L'absence de modèles d'interactions représente une lacune importante pour la spécification d'un SMA. Ce point devrait cependant être vite corrigé, puisque [37] signale qu'un tel modèle est en cours de développement (ainsi d'ailleurs qu'un modèle environnemental).
- L'approche n'est pas du tout orientée vers la réutilisabilité ; en particulier, l'existence de liens intergroupes exclut l'utilisation du groupe comme entité réutilisable dans un autre contexte.
- L'approche se limite à la spécification et n'aborde pas du tout le passage à l'implémentation. De fait, les agents sont les grands absents de MOISE+. Une réflexion semble donc s'imposer pour réduire le fameux « implémentation gap ».

Chapitre 4

Approches pour la conception des systèmes multi-agent adaptatifs

Résumé

Le développement des systèmes complexes évoluant dans un environnement ouvert et dynamique est une tâche difficile, la programmation classique est inappropriée pour la résolution des systèmes à haut degré d'interaction. Ce qui a amené à l'émergence de l'ingénierie orienté agent. Effectivement, les systèmes multi-agent adaptatifs s'appliquent parfaitement aux problèmes complexes dans un environnement en changement constant. Dans ce chapitre nous allons faire une revue des travaux les plus pertinents, essayés de faire une classification de ces travaux selon des critères particuliers, formaliser les notions, identifier les applications utilisées, voir l'utilisation des rôles dans le cadre de l'adaptation, faire les comparaisons en particulier avec les notions d'organisation et de rôle dans mon mémoire et en fin tirer les conclusions.

4.1 Introduction

Un système multi-agent, peut être autrement vu qu'un système en interaction, mais aussi comme une organisation où chaque agent est un composant appartenant à cette organisation.

L'organisation a fait l'objet de plusieurs recherches scientifiques en économie, sociologie, science politique...etc. Une organisation fournit une structure de contraintes influant sur le comportement des agents (ensembles de rôles), qui peut focaliser leurs décisions et l'action de certains agents. [1]

La notion d'organisation a été adoptée dans les SMA comme un avantage, inspirée de l'organisation sociale ou humaine. L'être humain à créer l'organisation pour être plus efficace et plus puissant que d'autres individus en interaction sans aucune organisation. [2]

Une organisation anticipe la coordination, cette anticipation mène à une organisation performante et efficace. Elle implique aussi un coût et une perte

de flexibilité, qui peut être profitable à l'organisation si l'anticipation est bien exploitée dans son environnement. [38]

Il est donc clair qu'une organisation est moins flexible qu'un ensemble en interaction (négociation, coordination, coopération). En effet du fait qu'il y'a certains nombre de contraintes, découlant de la spécification de l'organisation, régissant l'interaction entre les composants d'une organisation.

D'une manière générale, l'organisation est un modèle permettant aux agents de coordonner leurs actions au cours de la résolution d'une ou de plusieurs tâches. Elle définit d'une part, une structure (ex., une hiérarchie) comprenant un ensemble de rôles qui doivent être attribués aux agents, et un ensemble de chemins de communication entre ces rôles.

Elle définit d'autre part un régime de contrôle (ex., une relation maître/esclave) qui dicte le comportement social des agents. Enfin, elle définit des processus de coordination qui déterminent la décomposition des tâches en sous-tâches, l'allocation des sous-tâches aux agents, et la réalisation des tâches dépendantes de façon cohérente. [3]

L'environnement d'un SMA est imprévisible et en changement dynamique, et d'autre part un SMA est un système ouvert de nouveaux agents peuvent rejoindre le système et d'autre peuvent le quitter également. Les avantages d'une organisation peuvent s'avérés comme inconvenient, dans le cas où l'organisation n'arrive plus à s'adapter si l'environnement est trop instable. [39]

L'organisation donc est une abstraction importante qui a capté l'intérêt de plusieurs chercheurs en ingénierie orienté agent. Elle définit la topologie de la forme d'interaction et contrôle le régime de l'activité de l'organisation. [40, 41]

Voila pourquoi la structure des SMA doit être vue comme une première classe d'abstraction et doit agir comme un méta-concept indispensable pour mettre au point une méthodologie AOSE. [42]

4.2 Classification

Plusieurs travaux ont été menés portant sur le développement d'un système multi-agent robuste qui répond au changement de son environnement d'évolution, tout en maintenant un comportement cohérent et correct. On peut faire une classification de ces travaux, selon les moyens d'adaptations

du système qui vont nous servir de critères de classification. Nous allons prendre comme critères les techniques d'adaptations suivantes :

1. Modification de la structure relationnelle (changement des liens et d'interactions avec les autres membres de l'organisation permet le changement de la fonction globale émergente du système, on se basant également sur des algorithmes d'apprentissage, utiliser l'historique d'interactions des agents pour guider leur adaptation quand et comment...etc.).
2. allocation et réallocation des rôles (ex : en maximisant la fonction d'utilité...etc.).
3. la séparation entre le code opérationnel et fonctionnel de l'agent.
4. Interaction humaine pour le choix de la décision d'adaptation.

Selon le type de la technique d'adaptation utilisée on, peut classer les travaux dans l'une des quatre classes citée (classification au tableau2). Diverses approches portant sur ce domaine ont vu le jour certaines d'elles ce base sur la notion du rôle, son allocation et réallocation. On constate que les rôles peuvent être définis explicitement comme ils peuvent aussi ne pas être clairement définis par le concepteur du système multi-agent, dans le deuxième cas le rôle est une entité discrète, il émerge implicitement a travers les actions des agents de l'organisation.

Dans le premier cas se sont généralement des méthodes clairement spécifié ou la notion de rôle est explicitement définie. On constate également que la plupart des définitions, rôle explicitement définie, s'entendent sur le fait qu'un rôle a des responsabilités, permissions et des capacités. Effectivement un rôle une fois affecté à un agent il va spécifier explicitement les objectifs et les responsabilités de cet agent, le type de ressources auxquels il aura accès et le types de tâches que l'agent serai capable de mener a bien.

Dans d'autres travaux la définition peut même s'étendre au volet social et organisationnel ou le rôle peut spécifier à l'agent le protocole d'interactions et de communication avec ces semblables du système multi-agent. Les travaux du deuxième cas ne sont pas l'objet de notre chapitre, mais dans ce qui suit nous nous intéressons plutôt aux différentes définitions claires et explicites du rôle.

4.3 Approches multi-agent pour systèmes adaptatifs

4.3.1 Wooldridge et al. [43] Définissent un rôle comme l'abstraction des responsabilités de ce que performe l'agent ainsi que ces fonctions prévues.

- Permissions (droits associés au rôle) : En fonction du type et de la quantité de ressource pouvant être exploités lors de l'accomplissement d'un rôle, ce rôle aura des permissions qui lui seront associés. La permission associée à un rôle peut avoir deux aspects :
 - 1- Elle identifie la ressource qui peut être utilisée pour accomplir le rôle.
 - 2- Elle précise l'état de la limite de la ressource pour laquelle le rôle ne peut opérer.
- Responsabilité de survie d'un rôle : Un rôle est crée pour faire quelque chose, donc il a certaines fonctionnalités qui seront représentés par des attributs appelés responsabilités. Cette responsabilité peut être divisée en deux catégories :
 - 1- La longévité de la responsabilité, pour dire que «'quelque chose va se produire' et que l'agent assumant le rôle est toujours en vie.
 - 2- La sécurité exigée, est spécifiée par des prédicats qui sont typiquement exprimés par une liste de variable dans les attributs de permission du rôle.

Pour illustrer les différents concepts associé au rôle, on va choisir un exemple simple celui du 'Coffe Filler ' le rôle du 'CoffeeFiller'. L'objectif de ce rôle est de s'assurer que le gobelet du café d'un groupe de travailleurs reste toujours plein :

- Responsabilité de survie du rôle 'CoffeeFiller' :
 - 1- Remplir le gobelet à chaque fois qu'il devient vide.
 - 2- S'assurer que les travailleurs sont informés à chaque fois qu'un café frais est préparé.

$x.y$	x followed by y	$x y$	x or y occurs
x^*	x occurs 0 or more times	x^+	x occurs 1 or more times
x^ω	x occurs infinitely often	$[x]$	x is optional
$x y$	x and y interleaved		

Tableau 1 : opérateurs d'expression de survie.

Dans ce modèle les propriétés de survie de l'agent sont spécifiées par des expressions de survie qui définissent le cycle de vie du rôle, les expressions de survie définissent le potentiel d'exécution aux file des activités et interactions associés au rôle. La forme générale d'une expression de survie est :

$$\text{Rolename} = \text{expression}$$

Ou 'Rolename' c'est le nom du rôle, 'expression' c'est les l'expression de survie définissant les propriétés de survie du rôle. Le composant atomique d'une expression de survie est le protocole. Comme on a déjà mentionné les responsabilités du rôle 'CoffeeFiller' on note alors :

$$\text{CoffeeFiller} = (\text{Fill.InformWorkers.ChechStock.AwaitEmpty})^\omega$$

Cette expression indique qu'un 'CoffeeFiller' consiste a exécuté le protocole Fill, suivi par le protocole InformWorkers, suivi par le protocole CheckStock puis AwaitEmpty. Ces quatre protocoles sont ensuite répétés a l'infinie (w pour une répétition a l'infinie). On peut également noté :

$$\text{CoffeeFiller} = (\text{All})^\omega \text{ ou } \text{All} = \text{Fill.InformWorkers.ChechStock.AwaitEmpty.}$$

Dans plusieurs cas, il n'est pas suffisant de spécifier les responsabilités de survie du rôle. Parce que l'agent exécutant un rôle doit maintenir certains invariants, un agent de commerce par exemple doit s'assurer qu'il ne dépense pas plus d'argent qu'on lui a alloué. Ces invariants sont appelés les conditions de sécurité, elles assurent de ne pas se trouver dans des conditions indésirables. Dans cette méthodologie, les conditions de sécurités sont exprimées par une liste de prédicats via une liste de variables dans les attributs de permission du rôle. Dans le rôle 'CoffeeFiller' un agent assumant ce rôle doit s'assurer que le stock ne soit jamais vide : $\text{CoffeeStock} \geq 0$.

ROLE SCHEMA:	<i>name of role</i>
Description	<i>short English description of the role</i>
Protocols	<i>protocols in which the role plays a part</i>
Permissions	<i>"rights" associated with the role</i>
Responsibilities	
Liveness	<i>liveness responsibilities</i>
Safety	<i>safety responsibilities</i>

Fig.10 schéma modèle d'un rôle.

ROLE SCHEMA:	CoffeeFiller		
DESCRIPTION:	This role involves ensuring that coffee is kept filled, and informing the workers when fresh coffee has been brewed.		
PROTOCOLS:	Fill, InformWorkers, CheckStock, AwaitEmpty		
PERMISSIONS:	reads	supplied coffeeMaker coffeeStatus	// name of coffee maker // full or empty
	changes	coffeeStock	// stock level of coffee
RESPONSIBILITIES			
LIVENESS:	CoffeeFiller = (Fill.InformWorkers.CheckStock.AwaitEmpty)^ω.		
SAFETY:	<ul style="list-style-type: none"> • <i>coffeeStock</i> ≥ 0 		

Fig.11 schéma du rôle 'CoffeeFiller'.

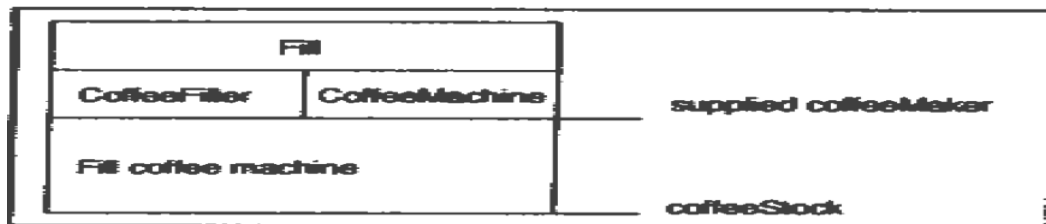


Fig.12 définition du protocole Fill.

4.3.2 La méthodologie Gaia [44] est un modèle conceptuel pour l'analyse et la conception des systèmes multi-agent. Les phases précédant l'implémentation i.e. l'analyse et la conception sont bien séparées, et les concepts clés sont bien définis dans chaque phase. Le but de la phase d'analyse c'est l'élaboration des caractéristiques fonctionnelles du système multi-agent et son organisation, qui inclue : identification des rôles joués par les agents de l'organisation, l'interaction entre ces rôles contrôlés par des protocoles, avec un maintien constant d'une cohérence des contraintes des rôles et de ces protocoles. Ces trois domaines (i.e. rôles, protocoles, et contraintes) constituent le modèle de rôle, le modèle d'interaction et les règles d'organisation qui sont introduites dans la phase de conception. Le domaine du modèle des rôles dans Gaia est un ensemble de rôles clés dans le système. Un rôle est défini par quatre attributs :

- 1- Protocoles : qui déterminent la façon d'interaction avec les autres rôles.
- 2- activités : tâches associées au rôle qu'un agent peut accomplir sans interaction avec d'autres agents.
- 3- Permissions : qui déterminent l'accès à l'information (ressources).

- 4- Responsabilités : qui déterminent le comportement prévu et le fonctionnement d'un rôle.

4.3.3 La structure ALAADIN [45, 46] est une organisation centrée méta modèle générique des systèmes multi-agent. Elle définit une structure conceptuelle générale qui sera utilisée dans le développement du système multi-agent. Elle décrit le système multi-agent d'une perspective organisationnelle, jusqu'à l'utilisation du terme d'état mental d'agents (centré agent) ces modèles aussi appelés AGR sont concentré sur trois concepts de base : Agent, Groupe, et Rôle. Un rôle est une représentation abstraite de la position fonctionnelle d'un agent dans un groupe et en même temps, c'est une vue de l'agent assumant ce rôle.

4.3.4 BRAIN (Behavioral Roles for Agent INteractions) [47] est une approche multi couches basée rôle, qui support le processus de développement des systèmes multi-agent. Elle fournit les composants : modèle d'interaction, notation basée XML et une infrastructure d'interaction. BRAIN représente un rôle par un ensemble de capacités et de comportement prévu. L'ensemble de capacités d'un rôle, est un ensemble d'action valable que l'agent assumant ce rôle peut performés pour accomplir une tâche. Le comportement prévu est un ensemble d'évènements que l'agent assumant le rôle devra manier. L'interaction entre deux agents est sous la forme d'un couplet (action, event), ou action est sous la forme d'un ensemble de capacités de l'agent initial, et event est géré par le comportement prévu de l'agent recommandé.

4.3.5 Un agent [48] joue des rôles au sein des groupes. Un agent peut avoir plusieurs rôles et appartenir à plusieurs groupes. Le rôle est une notion fonctionnelle abstraite, spécifique à un groupe et pouvant être joué par plusieurs agents. Un rôle est caractérisé par son unicité (vraie ou fausse), ses compétences (ou services), et ses capacités afin de remplir son rôle. Un groupe est un ensemble d'agents ayant des points communs et qui peuvent communiquer. La communication est interdite, pour des raisons de sécurité entre agents de groupes différents.

4.3.6 RoleEP [49] (Role based Evolutionary Programming,) est système supportant la construction d'applications coopératives d'agent mobile. Ce modèle est décrie via quatre notions de base : objectifs, rôles, agents et environnements. Un rôle consiste à des méthodes, attributs et interfaces de liens. Ils manipulent le voyage et la collaboration des fonctions des agents.

4.3.7 Eric Matson et al [50] proposent une solution en utilisant plusieurs agents organisés en équipe, pour donner aux robots la capacité de s'adapter pour combler les senseurs ou/et effecteurs qui viennent à tomber en panne. Lorsqu'on perd un effecteur ou/et senseur, l'équipe peut se réorganiser pour fournir le plus haut niveau d'utilité opérationnelle, avec les capacités courantes qui excitent. Les robots peuvent s'adapter en substituant le manque possible d'effecteurs ou/et de senseurs, par d'autres ensemble d'effecteurs ou/et de senseurs, qui doivent fournir plus de capacités pour combler le manque qui peut survenir. Les robots peuvent être amenés à opérer avec un bas niveau d'efficacités, mais seront capable d'atteindre l'objectif de la mission assigné.

Dans leurs recherches ils ont développés un modèle d'organisation basé système multi-agent, qui peut être appliqué au problème de perte de senseur/effecteur des robots. Dès qu'un senseur/effecteur tombe en panne, un autre est alloué a l'organisation, et y'aura une réassignation de tâches a l'ensemble entier de senseur/effecteur pour assurer un système tolèrent au faute et l'accomplissement de l'objectif du système. Pour implémenter une équipe d'agent autonome et hétérogène, le modèle d'organisation proposé est composé d'un modèle de structure, d'état et de fonction de transition.

La figure 13 montre la combinaison de la structure et de l'état du modèle en utilisant la notation standard d'UML. Le mode standard inclus l'ensemble d'objectifs (G) à atteindre par l'équipe, l'ensemble de rôles (R) qui doivent être joués pour parvenir à l'objectif voulu, ensemble de capacités (C) nécessaire pour jouer ces rôles et l'ensemble de lois (L) qui contraignent l'organisation. Le modèle contient également les relations statiques entre les rôles et les objectifs à achever, les rôles et les capacités requises et entre les rôles individuels reliés. Formellement on note une organisation structurée comme suit :

$$O_{\text{structure}} = \langle G, R, L, C, \text{achieves}, \text{related}, \text{requires} \rangle$$

Achieves : $R, G \rightarrow [0..1]$. Related : $R, R \rightarrow \text{Boolean}$. Requires : $R, C \rightarrow \text{Boolean}$.

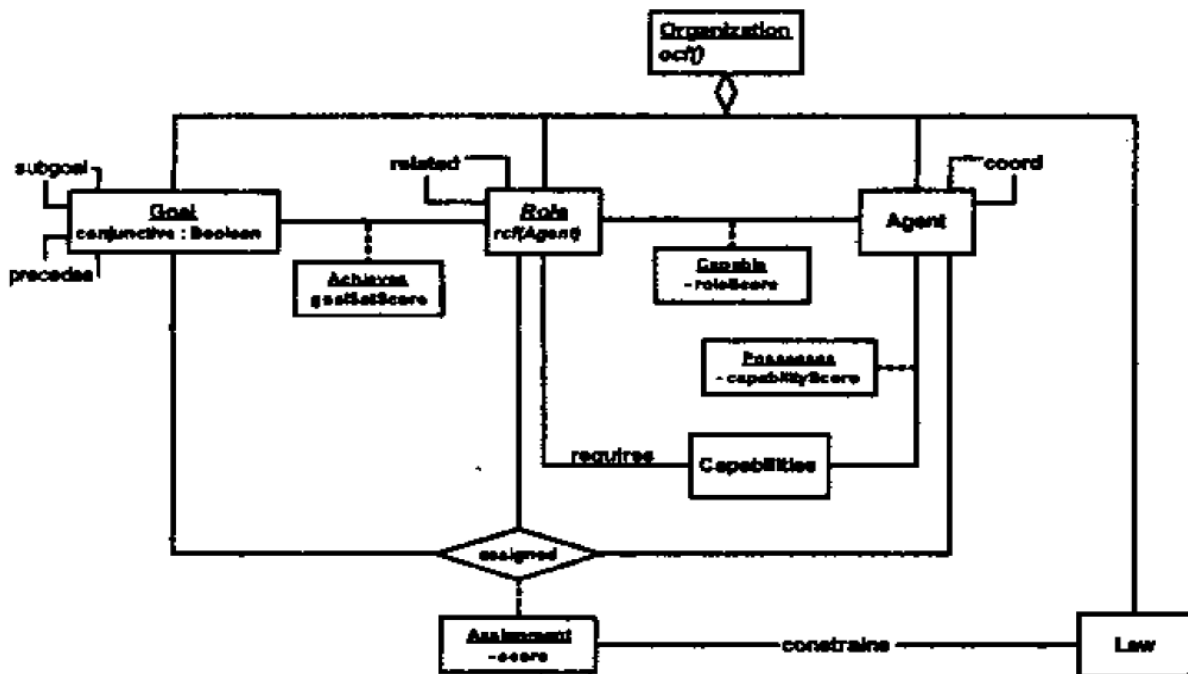


Fig.13 modèle d'organisation.

Le modèle d'état d'organisation définit une instance d'une organisation d'une équipe, et inclut un ensemble d'agent (A) et les relations actuelles entre les agents et les différents composants du modèle structurel.

$O_{state} = \langle A, possesses, capable, assigned, coord \rangle$
 Possesses: $A, C \rightarrow [0..1]$. Capable: $A, R \rightarrow [0..1]$.
 Assigned: $A, R, G \rightarrow [0..1]$. Coord: $A, A \rightarrow Boolean$

Un agent qui possède 'possesses' les capacités requises pour un rôle est dit capable de jouer ce rôle, donc tous les agents ne sont pas créés d'une façon égale. 'Possesses' prennent des valeurs réelles : 0 représente pas de capacités de jouer un rôle, 1 une excellente capacité. Mais ces capacités peuvent se dégrader avec le temps. Durant le processus d'organisation un agent spécifique est sélectionné pour jouer un rôle particulier, pour satisfaire un objectif spécifique. Cette relation est exprimée par une fonction 'assigned' qui inclut un score d'une valeur réelle, qui exprime comment un agent peut jouer un rôle pour satisfaire un objectif donné. Dans leurs recherches ils ont considéré uniquement la réorganisation qui engage l'état de l'organisation. Toutefois ils définissent deux types de réorganisations distinctes : une réorganisation d'état (qui permet uniquement la modification des états), et une réorganisation structurelle (qui permet la modification de la structure organisationnelle). Ils assument qu'un agent peut quitter ou rejoindre l'équipe ce qui entraîne une réorganisation, pour que l'organisation décide qu'une

réorganisation est nécessaire elle se base sur l'information 'possesses'. Calculée par l'agent sur sa capacité d'exécution des rôles. Pour démontrer les capacités d'adaptation, un scénario simple d'équipe de multi-agent est présenté. L'équipe d'organisation à un objectif global qui est $sense_{0-270}$ qui est décomposé en trois sous objectifs : $sense_{0-90}$, $sense_{90-180}$ et $sense_{180-270}$. Il y'a quatre agents (sonar1, sonar2, sonar3 et bump1), trois rôles (sensorA, sensorB et sensorC) et quatre capacités ($detect_{0-90}$, $detect_{90-180}$, $detect_{180-270}$ et $detect_{270-360}$).

$$G = \{sense_{0-90}, sense_{90-180}, sense_{180-270}\}.$$

$$A = \{\text{sonar1}, \text{sonar2}, \text{sonar3}, \text{bump1}\}.$$

$$R = \{\text{sensorA}, \text{sensorB}, \text{sensorC}\}.$$

$$C = \{detect_{0-90}, detect_{90-180}, detect_{180-270}, detect_{270-360}\}.$$

Les capacités requises pour chaque rôle sont :

$$C_r (\text{sensorA}) = \{detect_{0-90}\}$$

$$C_r (\text{sensorB}) = \{detect_{90-180}\}$$

$$C_r (\text{sensorC}) = \{detect_{180-270}\}$$

Initialement les capacités des quatre agents sont :

$$C_a (\text{sensor1}) = \{detect_{0-90}\}$$

$$C_a (\text{sensor2}) = \{detect_{90-180}\}$$

$$C_a (\text{sensor3}) = \{detect_{180-270}\}$$

$$C_a (\text{bump}) = \{detect_{90-180}, detect_{180-270}\}$$

4.3.8 Quelques approches, considèrent que le système multi-agent est une structure organisationnelle, dans laquelle les acteurs sont les membres du groupe afin de performer des tâches. Dans les travaux de Lucia R.D. Bastos et al. [51] l'organisation comprend : des groupes, des membres, des rôles et des interactions. Un membre a qui on assigne un rôle ne travail pas seul d'une façon isolé il coopère avec les autres rôles.

Cette approche, prend des sous groupes comme moyen pour décrire comment les acteurs dans une organisation peuvent être divisés en composants architecturaux, sur la base de leurs formes de relations avec les autres. Les rôles sont utilisés comme un concept intuitif pour l'analyse des besoins du système multi-agent, ou comme structure comportementale pour l'assignement des responsabilités aux composants d'une architecture logiciel. L'organisation est un groupe social dirigé par l'objectif est décrite en terme

d'acteurs et les rôles que jouent ces acteurs. Un rôle est une représentation abstraite du comportement des acteurs qui performes des fonctions similaires dans un groupe. Un rôle dénote une collection de responsabilités (éducateur, enseignant, etc.). Accomplir ces responsabilités implique la réalisation d'un ensemble de responsabilités de rôles. Comme exemple le rôle d'enseignant implique les tâches : 'd'enseigner' et 'de superviser'.

- Position : c'est le statu social ou l'identité sociale d'un membre dans un groupe (professeur associé).
- Responsabilité : identifie l'ensemble des tâches nécessaires pour l'achèvement de l'objectif (but) social d'un acteur jouant un rôle donné.
- Objectif : condition ou état de l'affaire du monde qu'on veut achever.
- Tâche : spécifie la manière particulière de faire quelque chose. Les tâches peuvent aussi être vues comme la solution dans un système, qui va satisfaire l'objectif. Cette solution fournit les opérations, les processus, la représentation de donnée, la structure et les contraintes pour avoir l'état voulu une fois l'objectif est atteint.

Un point important, est de noté la distinction entre un acteur i.e. l'entité organisationnelle physique, et la notion de rôle qui exprime la responsabilité de satisfaire certains objectifs organisationnels, en performant des tâches dans un groupe. Le rôle est séparé de l'acteur qui l'exécute, exemple un professeur peut jouer plusieurs rôles 'enseignant', 'chef département', etc.

4.3.9 Mirko Viroli et al. [52] Présentent une infrastructure pour les systèmes multi-agent ouvert sur un modèle de contrôle d'accès basé rôle (RBAC), qui fournit des agents qui donnent la permission et contrôlent les actions en fonction des autres entités du système multi-agent (agents et ressources). En générale un contexte de coordination de l'agent est créé et assigné à un agent dès qu'il intègre le système multi-agent, qui agit comme un sorte d'interface privé face à l'environnement : n'importe quelle action est permise et contrôlée par le contexte de coordination de l'agent, qui prévient tout comportement incorrect tout en respectant le rôle joué par l'agent.

Chaque rôle a une politique, qui spécifie les actions possibles et permises ainsi que la perception au fil du temps. Dans le RBAC un rôle est une sémantique construite autour de laquelle la politique de contrôle d'accès est formulée,

qui conduit vers une collection d'utilisateurs ainsi que les permissions des transitions. Un rôle peut représenter des compétences pour faire des tâches spécifiques mais aussi peut incarner l'autorité et la responsabilité : tout ces cas sont exprimés en terme de permettre ou non des opérations dans le cadre de l'objectif de l'organisation.

4.3.10 L'approche de Zhiyong Zhang et al. [53] propose un modèle de management référence pour système multi-agent coopératif : CACM 'Collaboration Access Control Model', l'essentiel de ce modèle est le rôle joué par un agent dans une société. Un agent qui acquière un rôle ou plus peut prendre tout les capacités correspondantes, et achever donc des activités définies des tâches de la société d'agent. Un CACM pour un système multi-agent, est composé de quelques composants de bases comme : agent, rôle, capacité, tâche, activité, société et contrainte. Zhiyong Zhang et al. Définissent un rôle comme suit :

Un rôle est une entité abstraite qui a des capacités et des responsabilités définis dans une organisation ou une société, un agent qui exécute un rôle devient une instance de ce rôle. La relation entre un rôle et un agent est une relation (n, n).

$$\forall r (r \in R) r = \{ ag_1, ag_2, \dots, ag_n \mid ag_i \in Ag \}.$$

$$\forall r (r \in R) r = \{ C_1, C_2, \dots, C_m \mid C_i \in C \}.$$

4.3.11 Alors que l'approche proposée par Xiaoqin Zhang et al. [54] dans leurs travaux sur les systèmes multi-agent qui, vise a facilité et formalisé le développement des systèmes multi-agent aussi augmenté leurs fiabilité et réutilisabilité. Ils visent à couvrir la phase d'analyse et de modélisation, la conception et l'implémentation. Leurs premier objectif est la séparation entre les différents concernés du système multi-agent, via trois couches de processus de développement. Ils visent aussi à séparer le domaine de connaissance par apport à la capacité d'intelligence de la résolution du problème. Ils ont adapté un modèle basé rôle, les rôles sont définis selon le domaine de connaissance relié avec : objectifs, permission, organisation des relations, les protocoles d'interaction ... etc. ou les agents est une entité concrète qui a des motivations, des ressources et des capacités pour la résolution du problèmes.

Leurs deuxième objectif est d'automatisé le processus de génération d'agent, en utilisant les outils et les mécanismes existants. Dans leur exemple chaque rôle est défini par son objectif, un plan arborisant, ensemble de motivations de quantité de production (MQPS), certificat et autres attributs.

Un objectif est une tâche qu'un rôle doit accomplir, est le plan arborisant spécifie le domaine de connaissance nécessaire à la manière d'accomplissement de cet objectif, et sa décomposition en sous objectifs.

ROLE : médecin. GOAL : fournir un remède.

MQPS : (MQ Professionnel P, p1), (MQ moral P, p2), (MQ expérience P, p3).

CERTIFICATE : MD (Docteur en Médecine).

Alors le rôle du médecin est de fournir un remède, le plan arborisant de la fig.14 nous montre le domaine de connaissance de comment atteindre notre objectif. Les MQPS spécifies le type et le nombre d'unités de motivation. Le certificat définie dans le rôle décrit la qualification requise pour se rôle. Ce rôle ne peut être accompli que par un agent détenteur de cette quantification ou certificat 'le rôle du médecin est défini par le certificat MD'.

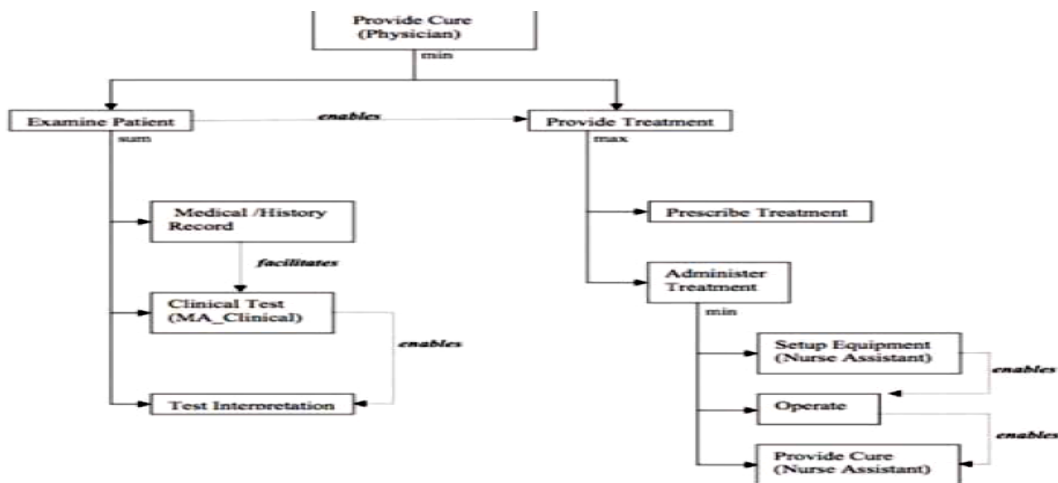


Fig.14 plan arborisant définissant le rôle de médecin.

4.3.12 Scott A. DeLoach et al [55] dans leurs recherches ont développé un méta modèle pour les organisations artificielles appelées OMACS (Organization Model for Adaptive Computational Systems) ce modèle permet aux développeurs des systèmes multi-agent de définir une structure pour supporter des applications spécifiques.

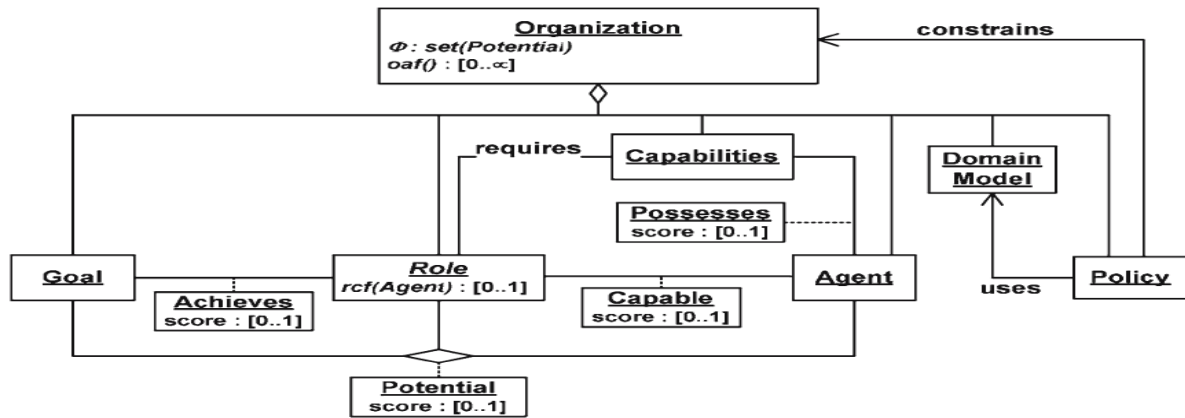


Fig.15 modèle d'organisation.

OMACS définit une organisation générale comme suit :
 $O = \langle G, R, A, C, \Phi, P, \Sigma, \text{oaf}, \text{achieves}, \text{capable}, \text{requires}, \text{possesses}, \text{potential} \rangle$.

- G objectif de l'organisation
- R ensemble de rôles
- A ensemble d'agents
- C ensemble de capacités
- Φ relation $G \times R \times A$ qui définit l'ensemble courant d'agent/rôle/objectif assigné
- P ensemble de contraintes sur Φ
- Σ modèle domaine utilisé pour spécifier l'objectif de l'environnement, ces inter relations et les opérations qui peuvent être performé
- Oaf fonction $P(G \times R \times A) \rightarrow [0..\infty]$ qui définit la qualité de l'assignement proposé
- Achieves fonction $G \times R \rightarrow [0..1]$ qui définit comment un rôle doit bien accomplir son objectif
- Capable fonction $A \times R \rightarrow [0..1]$ qui définit comment un agent doit bien jouer son rôle
- Requires fonction $R \rightarrow P(C)$ qui définit l'ensemble des capacités requise pour jouer un rôle
- Possesses fonction $A \times C \rightarrow [0..1]$ qui définit la qualité des capacités d'un agent
- Potential fonction $A \times R \times G \rightarrow [0..1]$ qui définit comment un agent doit jouer son rôles pour bien accomplir son objectif

Chaque organisation contient un ensemble de rôles (R) qui sont utilisés pour atteindre les objectifs de l'organisation. Un rôle défini une position dans une organisation son comportement prévu est d'accomplir un ou plusieurs

objectifs. Chaque rôle défini un ensemble de responsabilités, des droits et des relations définis pour aider la corporation a performer des fonctions diverses pour atteindre les objectifs fixés. Des agents sont assignés pour accomplir ces rôles et assumer les responsabilités des rôles, en utilisant les droits et les relations définis par le rôle. Un rôle OMACS est composé du nom du rôle et de la fonction de capacité du rôle rcf, chaque role $r \in R$ est un couple $\langle \text{name}, \text{rcf} \rangle$

Name : chaine de caractère. Rcf : fonction $A \rightarrow [0..1]$ qui définit comment un agent peut jouer un rôle sur la base des capacités que possède l'agent.

L'application d'approches spécifiques pour la réorganisation, permet au concepteur de définir des heuristiques pour guider la réorganisation du système. Au lieu d'utiliser des algorithmes génériques, les concepteurs peuvent spécifier un ordre dans lequel les agents peuvent exécuter les rôles. OMACS modélise cette heuristique comme un ensemble spécial de politique d'organisationnelle, appelé politique de réorganisation P_{reorg} . La réorganisation du système commence d'abord par tenter la politique de réorganisation, s'il y'a échec d'autres approches générales seront utilisées. Un exemple d'application d'une règle de politique de réorganisation :

$$\begin{aligned} \langle a1, r1, g1 \rangle \in \phi \wedge \neg \text{capable}(a1, r1) \wedge \text{capable}(a2, r1) \\ \Rightarrow \langle a2, r1, g1 \rangle \in \phi' \wedge \langle a1, r1, g1 \rangle \notin \phi' \end{aligned}$$

Ici ϕ' se réfère a l'assignement après la réorganisation. Dans se cas la règle spécifique appliquée, est que l'agent a1 jouant r1 pour atteindre l'objectif g1 puis a1 devient incapable de jouer le rôle r1. Alors si a2 a les capacités de jouer le rôle r1 ce rôle donc devrait être assigné à l'agent a2 et a1 va être déchargé du rôle r1. Voici donc l'algorithme général de réorganisation.

```

function reorganize(oaf,  $G_w$ ,  $A_w$ )
1. for each  $g \in G_w$ 
2.   for each role  $r \in R$ 
3.     if achieves( $r, g$ ) > 0 then  $m \leftarrow m \cup \langle r, g \rangle$ 
4.    $ps \leftarrow P_\phi(\text{powerset}(m))$ 
5.   for each agent  $a \in A_w$ 
6.     for each set  $s \in ps$ 
7.       if capable( $a, s, r$ ) then  $pa \leftarrow pa \cup \langle a, s \rangle$ 
8.    $pas \leftarrow \text{powerset}(P_\phi(pa))$ 
9.   for each assignment set  $i$  from  $c$ 
10.    for each assignment  $x$  from  $pa$ 
11.       $\Phi \leftarrow \Phi \cup \langle x.a, x.s_i \rangle$ 
12.    if  $P_\phi(\Phi)$  is valid
13.      if oaf( $\Phi$ ) > best.score then best  $\leftarrow \langle \text{oaf}(\Phi), \Phi \rangle$ 
14. return best. $\Phi$ 

```

Afin de démontrer l'efficacité du modèle OMACS, la simulation de Battlefield Information system (BIS) a été implémentée. L'objectif de ce système est de fournir une information efficace. Différent type de senseurs aux différents locations sont utilisés pour détecter l'ennemi. Ces senseurs sont sujets d'erreurs, de pannes et ont un délai de transmission d'information.

4.3.13 Le travail de Kota et al [56] démontre une approche décentralisée robuste, pour l'adaptation structurée des problèmes explicitement modélisés orientés organisation d'agent. Cette méthode permet aux agents d'utiliser l'historique de leurs interactions pour guider leur adaptation quand et comment. Basée sur le principe d'auto organisation cette méthode, permet aux agents de modifier leurs relations structurelles, pour accomplir une meilleure allocation des tâches.

Ils démontrent que la performance de leur approche est empiriquement proche d'une méthode d'allocation centralisée, dans les deux cas d'une organisation ouverte ou fermée, est meilleure qu'une adaptation aléatoire. L'adaptation structurelle décentralisée dans l'organisation des agents, est la méthode la plus appropriée pour la réalisation de l'auto organisation. Ici l'organisation est exprimée en matière de relation entre les agents, ce qui implique qu'une adaptation de la structure revient à changer les relations entre les agents en redirigeant leurs interactions.

L'organisation consiste en un groupe d'agents A qui fournissent des services, et chaque agent est capable d'accomplir un ensemble fixe de services, et possède des capacités computationnelles fixes. Dans leurs modèles les agents de l'organisation est un groupe résolvant des problèmes, les agents coopératifs sont placés dans un environnement de tâches (Task Solving Environment). Les agents résolvants les problèmes reçoivent en entrée un certain nombre de tâches à accomplir, et retournent après exécution des sorties. L'environnement de tâche présente un flot continu de tâche à exécuter, il est associé un cout à la communication entre les agents (message échangé entre agent) et au changement des relations également (réorganisation).

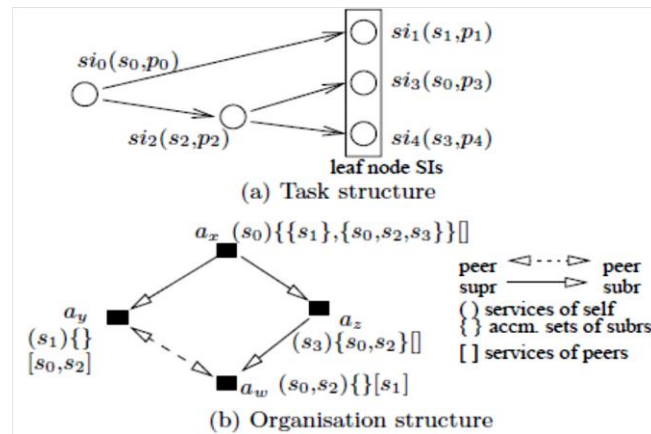


Fig.16 Représentation d'un ensemble de tâche et d'organisation.

Dans la Figure 16 on voit un exemple d'une tâche composée de cinq ensemble de services SI_s ($si_0 \dots si_4$), chaque un nécessite un service particulier ($s_0 \dots s_3$) et a besoin d'un nombre spécifié de computation ($p_0 \dots p_1$). Dans une organisation donnée, l'ensemble des agents peut rester statique et on dit qu'on a une organisation fermée. Sinon les agents peuvent rejoindre ou quitter l'organisation, on dit alors que notre organisation est dynamique. La performance d'un système computationnel dénote comment les tâches sont-elles exécutées? En termes d'organisation d'agent, la performance dans notre modèle est la somme des couts des messages transmis (communication), et des couts de changement de relations (réorganisation) entre agents.

$$cost_{ORG} = C \cdot \sum_{ax \in A} Cx + D \cdot d$$

Ou C est le coefficient du cout de la communication représentant le cout d'un message, D est le coefficient de cout du changement d'une relation simple, c_x est le nombre de messages transmis par l'agent x et d est le nombre de relation changé dans l'organisation. L'auto organisation basée sur l'adaptation structurelle, est une méthode qui peut être employée de façon continue par tous les agents résolvants des problèmes organisationnels.

Puisque on pose que les agents ne possèdent aucune information concernant les tâches qui peuvent arrivées au futur, on se base donc sur l'historique des interactions entre les agents.

Ils présentent le mécanisme l'adaptation, sous forme d'algorithme en pseudo code.

Algorithme 1

```

1  $Chosen \leftarrow$  selected from the acquaintances set of  $a_x$ ;
2 foreach  $a_y \in Chosen$  do
3    $Actions \leftarrow$  possible_actions( $x, y$ );
4    $U_{x,y} \leftarrow \emptyset$ ;
5   foreach  $e \in Actions$  do
6      $U_e \leftarrow$  calculate_utility $_{x,y}(e)$ ;
7      $U_{x,y} \leftarrow U_{x,y} \cup U_e$ ;
8   end
9    $e_{best} \leftarrow$  argMax( $U_{x,y}$ );
10  take action  $e_{best}$  with  $a_y$ ;
11 end

```

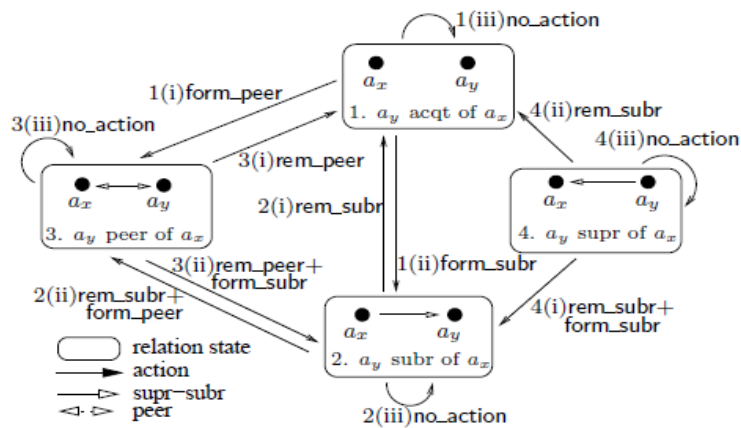


Fig.17 diagramme de transition.

Afin de démontrer l'efficacité de cette méthode, elle est comparée à deux autres méthodes centrale et aléatoire. Les résultats dans les deux cas d'une organisation statique ou dynamique montre que cette méthode, appelée k-adapt, donne des performances meilleurs (de 80% que la méthode centralisée).

4.3.14 Roman Neruda et al. [57] décrivent et compose un système multi-agent ad hoc, qui représente un modèle d'hybridation computationnelle entre datamining et l'analyses objectives des données. Ils se concentrent sur une ontologie de description convenable, qui permet l'automatisation de la création et la configuration géré donnée d'un système multi-agent. Le rôle ici inclus des groupes d'objectifs avec un comportement similaire. La décision de traiter un rôle comme un concept séparé de classes et d'objet, est motivé par le fait qu'un objet peut exhiber plus d'une seule forme de comportement, et peut accepter un nouveau rôle durant son exécution dans le système. Un rôle est défini par un ensemble de capacités i.e. actions (interactions) qu'un

agent assumant un rôle peut utiliser, et un ensemble de responsabilités ou évènements qu'un agent doit manier. La définition découle du concept d'agent et contient leurs responsabilités, les agents computationnels (CompAgent) ont comme responsabilité de répondre au contrôle de connexions :

CompAgent \subseteq Agent $\cap \exists$ hasResponder.ControlMsgResp
 The data source (DataSource) handles requests for data:
 DataSource \subseteq Agent $\cap \exists$ hasResponder.DataMsgResp
 The task manager (TaskManager) role only sends messages
 in a group:
 TaskManager \subseteq Agent

4.3.15 Dans les systèmes multi-agent ouvert OMAS (Open Multi Agent System), la décision du choix du candidat avec qui il faut interagir est une tâche difficile pour l'agent. On présente un mécanisme qui peut être utilisé par les agents dans les OMAS, pour avoir plus d'information pour prendre la bonne décision de sélection afin d'accroître leurs utilités. Le mécanisme qui supervise les interactions dans un OMAS, induit une taxonomie de rôle. Il assigne les agents aux rôles, on se basant sur l'observation de leurs performances dans différents types d'interactions.

Centeno et al. [58] propose un mécanisme adaptatif, qui développe une taxonomie de rôle en utilisant des algorithmes de classification multidimensionnel, pour la capture des formes de comportement des agents. Le mécanisme d'organisation structurelle d'agent pour la prise de décision, est décrit dans un environnement basé sur un système multi-agent orienté tâche T-MAS (Task Multi-Agent System) qui est défini comme suit :
 TM = < Ag, H, T, U > ou :

- A_g : est l'ensemble d'agent participant dans le système multi-agent, on assume que chaque agent ($a \in A_g$) a une fonction d'utilité $U_a : A_g \times T \rightarrow R$ ou A_g sont les agents délégués pour performer les tâche T.
- H : est l'ensemble des états de l'espace.
- T : ensemble des tâches qui peuvent être performé par les agents.
- $U : H \rightarrow R$: est le système de fonction d'utilité.

Le fonctionnement de T-MAS assigne une tâche à chaque agent si un agent $a_1 \in A_g$ ne peut plus performer une tâche, il va déléguer cette tâche à un autre agent $a_2 \in A_g$. Donc a_2 va performer cette tâche et a_1 obtiendra l'utilité de performance. Le mécanisme présenté par Centeno et al. Est basé sur l'utilisation du concept de rôle et la taxonomie de spécialisation de rôle. On conçoit un rôle d'un point de vu observateur i.e. comme un ensemble d'expectation, selon le comportement des agents performant certaines actions.

Ca veut dire qu'un rôle génère de lui-même quelques expectations publiques, selon le comportement d'agent assumant certaines actions ou ayant déjà accomplis. Un rôle dans T-MAS est une paire $\langle r, E \rangle$ donc les agents assumant un rôle r , sont qualifiés a performés les tâches de l'ensemble E , dans le sens qu'ils sont habilités pour ces tâches. La structure de la spécialisation de taxonomie de rôle, établie une relation spéciale \triangleright_r basée sur les compétences de l'agent assumant ces rôles ; étant donné deux rôles différents $r_1, r_2 \in R$. On dit qu'on a un sous ensemble de tâches de r_2 dans lequel des agents peuvent assumer mieux un rôle r_1 si et seulement si $r_1 \triangleright_r r_2$.

4.3.16 Danny weyns et al [59] présentent un modèle d'organisation de contexte de conduite, pour une organisation d'agent dynamique. Le modèle est une partie de l'approche MACODO (Middleware Architecture for Context-driven Dynamic agent Organizations). Dans ce modèle le management du cycle de vie de l'organisation dynamique, est séparé des agents : l'organisation représente les citoyens de première classe, tandis que leurs coté dynamique est gouverné par des lois. Les lois spécifient comment le changement dans le système, et le changement du contexte mène vers une organisation dynamique.

Le modèle est appliqué pour spécifier l'organisation dynamique pour la surveillance du trafic routier. Les systèmes multi-agents appartiennent à la classe des systèmes décentralisés, qui sont connues pour leurs qualités d'adaptabilité, et de robustesse avec beaucoup de dynamisme et un haut degré de distribution de ressources. Dans une organisation les agents travaillent ensemble basés sur des rôles bien définis, chaque rôle est responsable d'une fonction particulière de l'organisation. Un rôle décrit un ensemble cohérent de capacités requises pour réaliser une fonction particulière utile a la collaboration.

Le système de surveillance du trafic routier, consiste en un ensemble de caméras intelligentes distribuées tout au long de la route. Chaque caméra à

une couverture limitée, elles sont donc placées pour avoir le maximum de couverture possible et le minimum de zone d'ombre.

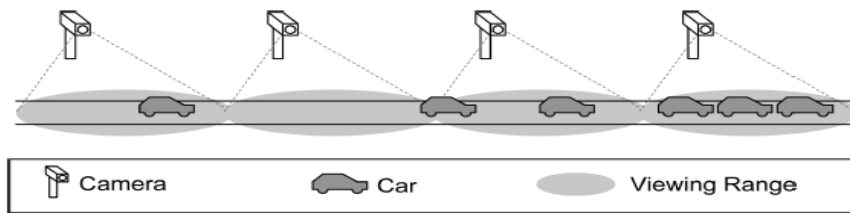


Fig.18 exemple de trafic avec cameras.

Les caméras sont équipées d'un processeur de données, capable de traiter les données de surveillance et d'une unité de communication pour que les caméras puissent communiquer entre elles. La tâche des caméras est de détecter et surveiller le trafic d'une façon décentralisée, pour éviter la congestion et les bouchons. La figure 19 est une abstraction basic du modèle MACODO :

Les agents sont les entités actives du système, un agent logiciel est déployé au niveau de chaque caméra. Au scénario agent1 est déployé à la caméra C1, agent2 à la caméra C2 ...etc. chaque caméra peut jouée trois rôles différent : data observer, data pusher et data aggregator. Un agent peut jouer plusieurs rôles simultanément. L'organisation org3 consiste au agent3 et agent4, qui collaborent ensemble pour tenir informer les clients concernés de la congestion de trafic dans la couverture des caméras C3 et C4. Donc on peut définir le système MACODO, comme un ensemble d'agents et d'organisations qui se conforme au modèle MACODO. Le système MACODO pour la surveillance du trafic routier, est une application qui consiste en quatre caméras et nombre d'organisation de trafic.

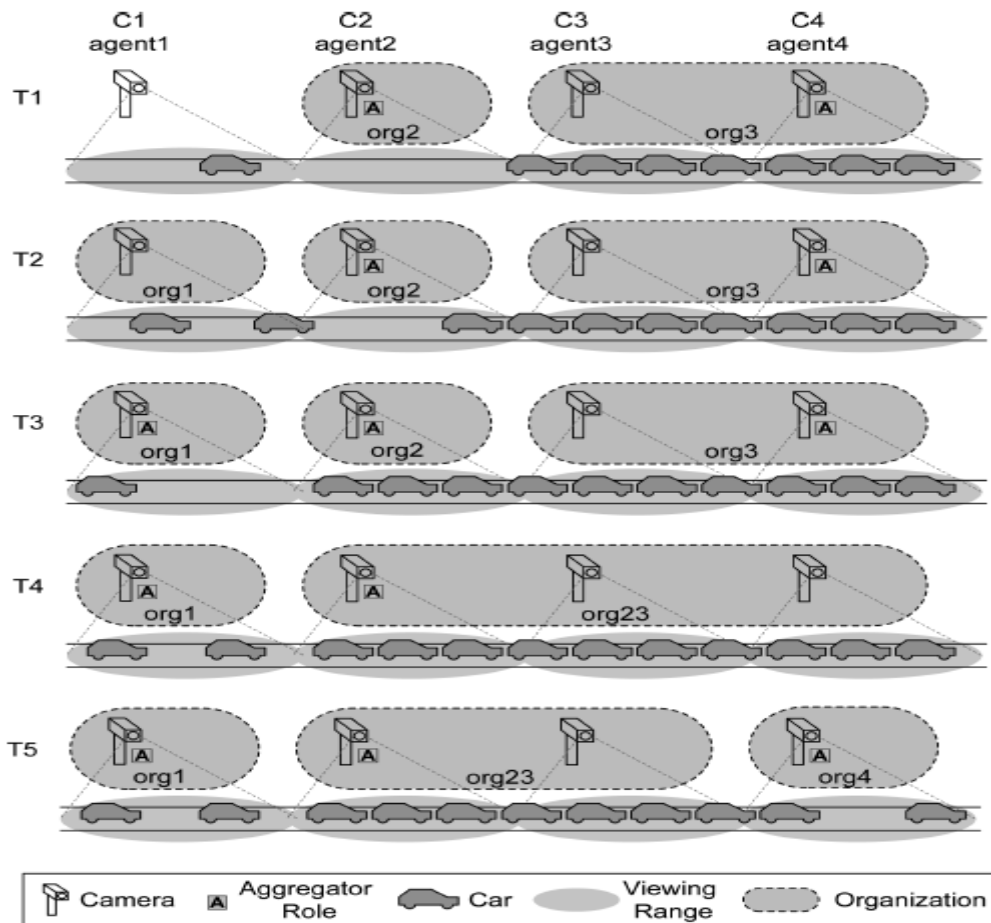


Fig.19 scénario illustratif d'une abstraction basic du modèle MACODO.

Formellement on utilise Z comme un langage spécifique, Z est un langage formel hautement expressif, qui est régulièrement utilisé pour la description et la modélisation des systèmes computationnels, y compris les systèmes multi-agent. La description formelle du modèle organisationnel MACODO, consiste en deux parties. La première partie introduit un ensemble de schémas, pour décrire le modèle de donnée qui représente l'état du système MACODO.

La deuxième partie introduit un nombre de fonctions et de schéma opération, pour décrire les lois qui représentent le comportement du système MACODO. Le modèle de donnée est composé de trois parties : la première partie incluse un ensemble basic de noms, contexte, capacités et rôles. La deuxième partie définit la position du rôle, contrat du rôle et les agents. La troisième partie définit l'organisation, contexte, les organisations et le système MACODO. Une organisation est définie comme suit :

<p><i>ORG</i> [<i>ORGSTATE</i>]</p> <p><i>name_o</i> : <i>ORGNAME</i></p> <p><i>rolepositions_o</i> : \mathbb{P} <i>ROLEPOS</i></p> <p><i>rolecontracts_o</i> : \mathbb{P} <i>ROLECONT</i></p> <p><i>orgcontext_o</i> : <i>ORGCONTEXT</i>[<i>ORGSTATE</i>]</p> <hr/> <p>$\forall rp : rolepositions_o \bullet rp.orgname = name_o$</p> <p>$\forall rc : rolecontracts_o \bullet rc.roleposition.orgname = name_o$</p>

4.3.17 Le travail de Haibin Zhu [60] sur assignement des rôles dans la collaboration basée rôle, il formule et identifie le problème du processus d'assignement des rôles, avec des contraintes d'agents conflictuelles. Initialement il propose une solution basée sur les algorithmes d'assignement des rôles précédents, pour expliquer que le problème d'assignement des rôles devient très complexe une fois des contraintes conflictuelles des agents sont considérées.

Il révisé dans son travail le modèle E-CARGO ou un système Σ peut être décrite comme suit $\Sigma = \langle C, O, A, M, R, E, G, S_0, H \rangle$ ou C est l'ensemble des classes, O est l'ensemble des objectifs, A est l'ensemble des agents, M est l'ensemble des messages, R est l'ensemble des rôles, E est l'ensemble d'environnement, G est des groupes, S_0 est l'état initial du système et H est l'ensemble des utilisateurs. A et H, E et G sont des ensembles fortement couplés. Les utilisateurs de A et leurs agents assument ensemble leurs rôles. Chaque groupe doit travailler dans un environnement. Un environnement régule un groupe. La définition du rôle dans le travail de Haibin Zhu est comme suit :

Un rôle est définie comme un triplet $r = \langle id, \textcircled{R}, \textcircled{W} \rangle$ ou :

- Id est l'identificateur du rôle.
- \textcircled{R} sont les requêtes d'un agent pour assumer un rôle r.
- \textcircled{W} sont les droits et les devoirs d'un agent pour assumer un rôle r.

4.3.18 M. Becht et al. [61] introduisent un environnement de programmation et une architecture de développement application coopérative, basé agent en utilisant une approche basée rôle. Ils se concentrent sur l'aspect coopératif en introduisant les processus coopératifs (CP) comme un concept à part entière, les CP décrivent tout et seulement la partie coordination et la coopération de l'application. La documentation explicite du mécanisme de coordination et de coopération utilisé dans un système multi-agent, permet

leurs évaluations et réutilisations. Il est possible de changer des processus déjà existant ou d'introduire de nouveau processus en online, sans modifier les agents existants. Dans la spécification du processus de coopération, on introduit un rôle comme une abstraction d'un agent, l'agent achève l'action initiée par le rôle. Le rôle fournit une bonne interface (Role-Agent Interface) entre l'agent et le processus de coopération, qui permet à l'agent de lire et suivre les règles normales données par le processus de coopération.

On définit un rôle comme une entité constituée d'un ensemble de permissions requises R , d'un ensemble de permissions accordées G , un graphe direct de services à invoqués A et un état S visible seulement pour l'environnement courant mais pas aux autres agents. A décrit le comportement des agents et peut contenir un nombre arbitraire d'alternatives, l'état du rôle sera utilisé par l'environnement courant pour déterminer si le processus de coopération a atteint le stage prochain, les rôles peuvent engendrer des sous coopérations et alors un des sous ensembles de rôles associés, les sous rôles hérite des permissions de leurs super rôles.

4.3.19 Les approches de réorganisations actuelles sont principalement focalisées à fournir une solution de réorganisation, qui maximise le bénéfice aux organisations futur. Elles négligent l'impact du coût qui pourra valoir le processus de réorganisation. Juan M. Alberola et al [62] fournissent un facilitateur de réorganisation de service, qui implémente justement un mécanisme de réorganisation, qui prend en compte le coût basé sur la transition d'organisation. Le mécanisme fournit le coût associé à chaque transition vers une nouvelle organisation, et les conséquences des étapes nécessaires pour cette transition.

Une organisation à un moment spécifique t est définie comme suit $O_t = \langle OS_t, OE_t, \Phi_t \rangle$. OS : détaille les éléments de l'organisation par une paire $OS = \langle SD, FD \rangle$, SD est la structure dimensionnelle et FD est la fonction dimensionnelle. SD décrit l'ensemble des rôles R contenus dans l'organisation à un moment spécifique t . $FD = \langle S, provider \rangle$ décrit l'ensemble des services S qu'offre l'organisation à un moment spécifique t . chaque service est offert par un ensemble de rôle cela implique une relation provider : $S \rightarrow 2^R$. L'entité organisationnelle OE décrit la population d'agent A à un moment spécifique t . en fin l'organisation dynamique $\Phi = \langle plays, providers \rangle$ représente les relations qui existe entre les éléments de OS et OE :

- Plays : $A \rightarrow 2^R$, relie un agent assumant un ensemble de rôle à un moment spécifique.
- Providers : $A \rightarrow 2^R$, relie un agent fournissant un ensemble de services à un moment spécifique.

Pour qu'un agent 'a' soit capable de jouer un rôle 'r', 'a' doit fournir tout les services qu'offre 'r' :

$$\forall \text{ plays}(a, r) \in \Phi^t \mid \text{provider}(s, r) \in OS^t \rightarrow \text{provides}(a, s) \in \Phi^t$$

Soit une organisation donnée, à un moment spécifique t composée des éléments qui regroupent les objets (rôles, agents, et agents), et le relations (plays, provides, et provider). Ces éléments peuvent changer pendant la durée de vie de l'organisation.

La transition d'organisation nous permette de relier deux instances différentes de la même organisation à deux moments différents, initiale (ini) et finale (fin). La fonction de transition définie comment passer d'une organisation O^{ini} à une autres O^{fin} en exécutant le mécanisme de transition, aussi on passe de Φ^{ini} , OS^{ini} et OE^{ini} a de nouveau OS^{fin} , OE^{fin} et Φ^{fin} .

Un évènement ε défini chaque changement qui peut s'opérer à un objet ou une relation durant la transition d'organisation, en termes d'ajout ou de suppression d'objet ou de relation. Exemple : $\varepsilon = \text{add}(\text{role})$) pour dire qu'on ajoute un rôle r. T défini l'ensemble d'évènements qui permettent la transition ver O^{fin} on les appliquant a O^{ini} : $O^{\text{ini}} \times T \rightarrow O^{\text{fin}}$

$T = \{ \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n \}$. À chaque évènement est associé un cout, l'application d'un ensemble T d'évènements nécessaire pour une transition implique donc un cout $C_{\text{transition}}$.

$$C_{\text{transition}} = C(T) = \sum_{\varepsilon \in T} C(\varepsilon)$$

L'organisation dynamique Φ^{fin} représente la relation entre OS^{fin} et OE^{fin} . Cette relation définie quel service fourni chaque agent, et quels rôles jouent chaque agent à un moment spécifique t. Le problème de réallocation de rôle, est de rendre le cout d'un ensemble d'évènements T_ϕ computationnel, qui permet la transition d'une organisation initiale O^{ini} à une autres finale O^{fin} . Cet ensemble d'évènement contient plays, des relations provides. Ce cout doit définir combien coutera à un agent de faire : acquérir un service pour jouer un rôle spécifique, commencer à jouer ce rôle, arrêter de jouer le rôle courent et d'arrêter de fournir le service de l'ancien Rôle. Chaque possible

réallocations définies une Φ^{fin} différente qui implique OS^{fin} et OE^{fin} , et qui a un cout de transition associé C_{ϕ} . le challenge est donc de trouver l'ensemble d'évènements T_{ϕ} qui minimise le cout de transition i.e. $T_{\phi^{\text{min}}}$.

On note Θ l'ensemble de tout les T_{ϕ} possible : $T_{\phi^{\text{min}}} = \{\sum \epsilon \in T_{\phi} \mid C(\epsilon) \mid T_{\phi} \in \Theta\}$

4.3.20 Dans leurs travaux [63] ils présentent une approche d'adaptation de système multi-agent basée sur un mécanisme de délibération de transition multidimensionnel (MTDM). Elle considère la transition dans multiples dimensions et vise à une adaptation qui maximise le potentiel, pour une meilleure utilité mais aussi basée sur le cout d'adaptation.

Par contre ils présentent une nouvelle définition de l'organisation : Une organisation à un moment spécifique t est définie comme suit : $O^t = \langle O^t_O, O^t_R \rangle$. O^t_O : ensemble d'objet de l'organisation $O^t_O = \{R^t, S^t, A^t\}$, R^t est l'ensemble de rôle contenus dans l'organisation à un moment spécifique t , S^t est l'ensemble de service coffre l'organisation à un moment spécifique t , A^t est la population d'agent de l'organisation à un moment spécifique t .

O^t_R : ensemble de relation de l'organisation, ce sont les liens entre les objets, $O^t_R = \{\text{offers}^t, \text{provides}^t, \text{plays}^t, \text{acquaintance}^t\}$ avec les conditions suivantes :

- 1- $\text{offers}^t = \{(r, s) \in R^t \times S^t\}$ représente la relation entre des rôles et des services
 (r, s) : un rôle r offre un service s à un moment spécifique t .
- 2- $\text{provides}^t = \{(a, s) \in A^t \times S^t\}$ représente la relation entre des agents et des services
 (a, s) : un agent a fourni un service s à un moment spécifique t .
- 3- $\text{plays}^t = \{(a, r) \in A^t \times R^t\}$ représente la relation entre des agents et des rôles
 (a, r) : un agent a joue un rôle r à un moment spécifique t .
- 4- $\text{acquaintance}^t = \{(a, a') \in A^t \times A^t\}$ représente une relation entre une paire d'agent
 (a, a') : un agent a et un agent a' son connectés par une accointance à un moment spécifique t .

Juan M. Alberola et al on utilisés l'application de service des touristes comme exemple. Cette application est composée d'agents groupés en trois organisations différentes : users agents, broker agents et provider agents. User agents nécessite des services des touristes et demande des informations

concernant la réservation d'hôtel, de vols, trains ... etc. Ces agents interagissent avec broker agent pour obtenir l'information demandée. Broker agent maintiennent l'information des touristes, et agissent comme un intermédiaire entre user agent et provider tourist. Provider agents sont des agents qui font partie d'hôtels spécifiques, compagnie aérienne ...etc. Comme exemple d'organisation c'est l'organisation de broker agents qui sera choisie comme exemple. Ils utilisent dans leurs exemple le modèle de transition, une organisation $O^{ini} = \langle OS^{ini}, OE^{ini}, \Phi^{ini} \rangle$ est définie comme l'instance courante de l'organisation des broker agents. La spécification organisationnelle $OS^{ini} = \langle SD^{ini}, FD^{ini} \rangle$ définit la dimension structurelle qui spécifie l'ensemble des rôles $R^{ini} = \{r_1, r_2\}$ de l'organisation au moment ini :

- le rôle r_1 représente le rôle book_service_provider. Les agents jouant ce rôle ne peuvent pas sauvegarder des informations concernant les services des touristes, mais ils peuvent plutôt interagir avec les agents qui stockent ces informations à savoir les agents jouant le rôle r_1 .
- le rôle r_2 représente le rôle search_service_provider. les agents qui jouent ce rôle stock des informations de services de touristes, et sont uniquement capable d'interagir avec les agents assumant le rôle r_1 .

La fonction dimensionnelle FD^{ini} spécifie l'ensemble de services S^{ini} offert par l'organisation au moment ini : $S^{ini} = \{s_1, s_2, s_3, s_4\}$ ou :

1. s_1 représente le service search_hotel qui fourni l'information concernant les hôtels comme la disponibilité, prix, suites, location ...etc.
2. s_2 représente le service book_hotel qui fourni le service de réservation d'un hôtel spécifique en fonction de paramètres comme la date, nombre de nuit, petit déjeuner ...etc.
3. s_3 représente le service search_fly qui fourni l'information concernant les vols comme la disponibilité, prix, départ, arrivée ...etc.
4. s_4 représente le service book_fly qui fourni le service de réservation d'un vol spécifique, en fonction de certain paramètres comme lieu de départ, date, nombre de place disponible ...etc.

La FD^{ini} spécifie également le rôle qu'offre chaque service comme les relations de provider suivantes : provider (s_2, r_1), provider (s_4, r_1), provider (s_1, r_2), provider (s_3, r_2).

L'entité organisationnelle OE^{ini} définit la population courante : $A^{ini} = \{a_1, a_2, a_3, a_4, a_5\}$.

Et finalement l'organisation dynamique Φ^{ini} spécifie le service fourni par chaque agent comme provides relation :

Provides (a_1, s_1) , Provides (a_1, s_3) , Provides (a_2, s_1) , Provides (a_2, s_3) ,
Provides (a_3, s_1) ,

Provides (a_3, s_3) , Provides (a_4, s_1) , Provides (a_4, s_3) , Provides (a_5, s_2) ,
Provides (a_5, s_4) ,

Φ^{ini} spécifie également le rôle joué par chaque agent comme une relation plays :

Plays (a_5, r_1) , Plays (a_1, r_2) , Plays (a_2, r_2) , Plays (a_3, r_2) , Plays (a_4, r_2) .

4.4 Comparaison

Pour pouvoir comparer les différentes approches, nous allons devoir nous baser sur quelques critères de comparaison. Nous allons reprendre plus au moins les mêmes critères de comparaison que Juan M. Alberola et al [64] :

L'une des difficultés rencontrées lors de la conception d'un système multi-agent adaptatif, est quand faut-il déclencher le mécanisme de réorganisation (d'adaptation), ou qu'est ce qui peut justement déclencher le processus. Il se peut que le système soit **réactif** guidé par les événements, il fait donc face au changement quelconque s'opérant sur le système et son environnement. Tel que l'arrivée d'un nouvelle agent, ou au contraire un agent qui quitte le système, aussi l'ajout ou la suppression d'un rôle ou plus ...etc. Comme il est possible de concevoir un système **proactif** capable de raisonner sur l'état actuel du système, et décider si besoin d'adapter le système.

Le processus d'adaptation suit une logique grasse à laquelle il décide, si un système nécessite une adaptation. En respectant toujours cette même logique il procède effectivement à l'adaptation du système, cette logique peut être **prédéfinie** au moment de la conception du mécanisme d'adaptation, ou bien une logique **adaptable** qui peut changer : on peut citer quelque logique tel que maximiser une fonction d'utilité [50], maximiser la fonction d'utilité toute en minimisant le cout du processus de

réorganisation [62] ...etc. On définit deux types d'implémentation d'adaptation, l'implémentation est dite **centralisée** si un agent est capable à lui tout seul de décider si une adaptation est nécessaire, par contre elle est dite **distribuée** si c'est l'ensemble des agents qui doivent décider si une adaptation est oui ou non nécessaire. Il y a des caractéristiques qui ne sont pas liées au mécanisme d'adaptation, mais liées plutôt au système qui peut être un système **ouvert** : ou un agent peut quitter le système ou un nouvel agent peut rejoindre également le système. Le système est dit **émergent** si le comportement social de la population change, comme la suppression ou l'addition de rôle ou d'objectif. Dans le tableau suivant une comparaison des différentes approches selon les critères déjà exposés :

		OMACS [55]	T-MAS [58]	[50]	[62]	TSE [56]	MACODO [59]	Notre approche
adaptation	stratégie	réactive	Proactive/ réactive	réactive	réactive	Proactive/ réactive	réactive	Proactive/ réactive
	logique	prédéfinie	prédéfinie	prédéfinie	prédéfinie	prédéfinie	prédéfinie	prédéfinie
	implémentation	centralisée	centralisée	distribuée	centralisée	distribuée	Centralisée/ distribuée	centralisée
système	ouvert		✓	✓	✓		✓	✓
	émergent		✓		✓			✓
classification		2	2	2/4	2	1	3	2

Tableau 2 : Tableau comparatif.

4.5 Conclusion

Dans ce chapitre, nous avons fait un survol des contributions les plus pertinentes dans le domaine des systèmes multi-agent adaptatifs. Nous nous sommes intéressés plus particulièrement aux différents travaux, où la notion du rôle est définie d'une façon claire et explicite. Dans ce chapitre, nous avons choisis le moyen d'adaptation utilisé dans chaque approche comme critère pour proposer une classification en quatre classes possibles : 1, 2, 3 et 4. Finalement la spécification de quelques critères dans la section 4.4 qui vont nous servir de base pour pouvoir faire une comparaison entre quelques travaux.

Chapitre 5

Problème posé et solution proposée

5.1 Problématique

Notre objectif est de réaliser une approche pour l'adaptation des systèmes multi-agents basés organisation des rôles, une approche capable de répondre aux contraintes d'un environnement ouvert, dynamique et en constante évolution.

Pour cela nous prenons pour base le concept d'adaptation du système multi-agent, par une réorganisation d'assignation des rôles aux agents, et comme moyen technique les systèmes multi-agents basés organisation (auto-organisation) des rôles.

5.2 Approches existantes

Plusieurs travaux ont été menés portant sur le développement d'un système multi-agent robuste, qui répond au changement de son environnement d'évolution, tout en maintenant un comportement cohérent et correct. Il existe plusieurs moyens d'adaptation utilisant diverses techniques d'adaptation on peut citer :

1. Modification de la structure relationnelle (changement des liens et d'interactions avec les autres membres de l'organisation, permet le changement de la fonction globale émergente du système. On se basant également sur des algorithmes d'apprentissage, utiliser l'historique d'interactions des agents pour guider leur adaptation quand et comment...etc.). On peut citer le travail TSE de Kota et al [56].
2. allocation et réallocation des rôles (ex : en maximisant la fonction d'utilité...etc.). On trouve le travail T-MAS de Centeno et al [58] et, le travail OMACS de Scott A. DeLoach et al [55] et ceux de Juan M. Alberola et al [62] [63].
3. la séparation entre le code opérationnel et fonctionnel de l'agent. Danny weyns et al ont proposés l'approche MACODO [59] qui adopte ce type d'approche.
4. Interaction humaine pour le choix de la décision d'adaptation. On peut citer le travail d'Eric Matson et al [50].

5.3 Contribution

La réalisation d'une approche adaptative, capable d'adapter le système aux nouveaux changements de l'environnement. Une nouvelle réorganisation de répartition des rôles aux agents du système, grâce à une réassignation des rôles qui sera différente de la répartition initiale des rôles. Cela en réponse aux nouveaux événements qui peuvent surgir aux cours de l'évolution du système et son interaction avec son environnement.

L'organisation des agents n'est pas fixe, elle se transforme suivant les besoins du système et sa confrontation aux événements, l'organisation sera donc dynamique. L'intelligence de notre approche réside donc dans sa faculté d'auto-adaptabilité (elle est dite robuste) aux changements de contexte.

Ce qui rend notre approche plus efficace c'est sont trait multicritères, puisqu'elle s'inspire de [67] qui porte sur le développement d'organisation des systèmes multi-agents basés rôles. Il propose une approche d'optimisation d'organisation OP (Organization Partitioning) en adaptant l'algorithme PA (Particles Approach [68]) au partitionnement des rôles entre les agents du système.

Nous allons prendre à notre tour l'approche d'optimisation d'organisation, pour l'appliquer aux systèmes multi-agents adaptatifs. L'adaptation donc dans notre travail se fait on adoptant une méthode multicritères on introduisant des contraintes explicites, variés et qui peuvent être conflictuelles. Combinées ensemble pour ne constituer qu'un seul critère qui est le résultat de la force, ce qui nous permet de prendre en considération, d'une façon beaucoup plus large et plus complète, plusieurs paramètres dans la mise en œuvre de l'adaptation du système.

Contrairement à d'autres approches de réorganisation qui sont généralement focalisées sur un seul paramètre, par exemple à maximiser une fonction d'utilité ou encore à minimiser une fonction du coût de réorganisation. Ces critères donc seront calculés et mis-à-jour à chaque cycle d'exécution, pour servir à la résolution de conflit pour une meilleure coopération des agents de l'organisation.

Les interactions entre les agents sont fixes et s'effectuent par l'intermédiaire d'un rapport de dominance, défini lors de la conception. Pour le choix du rôle candidat à la migration (rôle dominant) d'un agent source à un autre agent destination, l'établissement de techniques de résolution de conflit est nécessaire. En appliquant justement un rapport de dominance en fonction des multicritères déjà spécifiés. C'est également un rapport de dominance, qui sera utilisé pour résoudre le conflit du choix de l'agent destination ou direction (agent dominant) parmi les voisins de l'agent source.

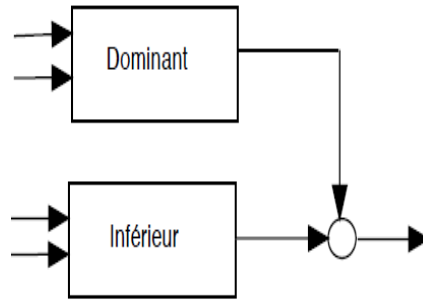


Fig.20 Interaction à l'aide d'un rapport de dominance.

Définition du rapport de dominance

La partition X domine la partition Y si les deux conditions suivantes sont vérifiées :

(A) $\forall k \in \{1, 2, \dots, m\} \quad f^{X_{gk}}(t_i) \leq f^{Y_{gk}}(t_i)$ 'X n'est pas mauvais que Y pour tous les critères.'

(B) $\exists k \in \{1, 2, \dots, m\} \quad f^{X_{gk}}(t_i) < f^{Y_{gk}}(t_i)$ 'Il existe au moins un critère pour lequel X est strictement meilleur que Y.'

A noté que la fonction d'évaluation reste ouverte pour permettre d'exprimer les différentes contraintes possibles qu'on voudra prendre en considération.

$$f_{(i,k)} = \langle \langle f_{1(i,k)}, f_{2(i,k)}, \dots, f_{n(i,k)} \rangle \rangle$$

Notre approche est une approche cohérente à tout moment de l'exécution et de l'évolution du système. Pour préserver cette cohérence de l'organisation il faut d'abord vérifier, avant d'autoriser la migration d'un rôle, quelques contraintes préventives de cohérence. D'autres contraintes correctives de rétablissement de cohérence, doivent également être vérifiées après la migration du rôle candidat.

Techniquement on suppose qu'au début du processus les agents n'assument qu'un seul rôle au plus. Pour chaque rôle r_i on associe un compteur z_i qui sera initialisé à zéro lors de la génération du rôle. À chaque migration z_i sera incrémenté. On définit une constante appelée max_{migs} , qui définit le nombre maximal de migration permise. Une migration ne peut s'effectuer que si $x_i = 1$ avec :

$$x_i := \begin{cases} 1, & \text{if } r \geq \frac{z_i}{max_{migs}} \\ 0, & \text{otherwise} \end{cases}$$

r est une variable aléatoire comprise dans l'intervalle $[0, 1)$.

Chaque agent de la société exécute d'une façon continue l'algorithme d'OP [67] :

```

cycle
  wait for load change
  if load increase
    then
       $M \leftarrow \{i \in T \mid x_i = 1\}$ 
      load_balanced  $\leftarrow$  false
      while  $M \neq \emptyset$  and not load_balanced do
        get information from neighbors

        for all  $i \in M$  do
           $f(i, s) = \langle 0, 0 \dots 0 \rangle$ 
          for all  $k \in \text{neighbors}$  do
             $f(i, k) = \langle f_1(i, k), f_2(i, k) \dots f_n(i, k) \rangle$ 
          end for
        end for

        repeat
          cand :=  $p$  with  $\underbrace{i \in M \wedge i \neq p \wedge k \in \text{neighbors}}_{\text{not dom } (f(i,k), f(p,k))}$ 
          direct :=  $q$  with  $\underbrace{k \in \text{neighbors} \wedge k \neq q}_{\text{not dom } (f(\text{cand},k), f(\text{cand},q))}$ 
          chosen := verify-constraints (cand, direct)
          remove  $f(\text{cand}, \text{direct})$ 
        until chosen

        if direct  $\neq s$  then
          migrate(cand, direct)
          re-establish-coherence (cand, direct)
        else
          load_balanced  $\leftarrow$  true
        end else
      end while
    end then
  else
    send new load data to neighbors
  end else
end cycle

```

5.4 Scénario de validation

Pour implémenter notre approche, nous allons utiliser un exemple dans le domaine du commerce électronique, basé sur une application de services pour touristes. Cette application est composée d'agents groupés en trois différentes organisations : user agents, broker agents et provider agents. User agents à besoin de services pour des touristes et demande des informations concernant la réservation d'hôtels, de vols, de trains et de bateaux etc.

Les agents de cette organisation interagissent avec broker agents afin d'obtenir les informations requises. Broker agents agissent comme un intermédiaire entre user agents et provider agents puisqu'ils sont capable de garder les informations des touristes en mémoire. Provider agents représente les agents appartenant à un hôtel spécifique, compagnie de vol, ferroviaire ou maritime. Pour notre application nous allons nous concentré sur l'organisation broker agents comme exemple d'organisation.

5.5 Conception

Dans cette section, nous allons décrire la méthodologie de développement de notre exemple SMA et pour cela nous avons choisi la méthodologie MASA-Method [65] comme processus méthodologique. MASA-Method [65] est une méthodologie complète. Elle intègre les quatre phases principales du développement organisationnel de SMA.

La section 5.5.1 explique le développement des tâches de systèmes. Elle permet de considérer une caractérisation procédurale des objectifs comme tâches de systèmes. La section 5.5.2 présente les notions de développement des modèles d'organisations élémentaires. Elle décrit la dérivation de ces organisations à partir des expressions des objectifs. La section 5.5.3 décrit le développement des modèles de systèmes multi-agent. Elle se concentre sur les aspects permettant de dériver ces modèles et sur l'utilisation des agents d'interface pour l'intégration des agents hétérogènes dans un même SMA.

Chacune des étapes (sections) comporte deux parties :

1. Moyens de description,
2. Processus méthodologique de développement.

Les moyens de description sont décrits sous forme de tables. Chaque table comporte les éléments à inclure dans une description donnée. Elle est composée d'au moins six parties : Numéro, objet, identification, corps, légende et commentaires.

Le numéro permet une identification pratique des tables. L'objet permet d'indiquer la nature de la table. L'identification comporte un identificateur. L'identificateur identifie l'objet en question. Elle peut comporter aussi des relations avec d'autres descriptions (tables). Le corps contient la description de l'objet. Il peut comporter lui aussi plusieurs parties selon la description de l'objet. La légende comporte une description des symboles utilisés dans les différentes descriptions de la table. Les commentaires sont destinés à commenter les différents éléments de la description.

Un processus méthodologique comporte quatre parties : Entrées, Sorties, Remarques et Processus de dérivation des sorties à partir des entrées.

5.5.1 Développement des tâches de systèmes

L'objectif d'un système, est le but qu'on lui fixe de telle sorte qu'un tel but sera atteint à l'issue du fonctionnement de ce système. Ce but sera une configuration que l'environnement aura à la fin du fonctionnement du système. Il s'agit de développer un système dont le fonctionnement permet d'atteindre un objectif donné.

Au lieu de la description de l'objectif lui-même, il est nécessaire, dans la caractérisation procédurale, de décrire comment atteindre l'objectif en employant les éléments du modèle de l'environnement. L'expression « comment atteindre l'objectif » peut être qualifiée de tâche. L'expression procédurale d'un objectif est alors définie comme étant un couple (M, T) où :

1. M est un modèle de l'environnement,
2. T est une tâche permettant d'atteindre l'objectif.

Le système met en œuvre les tâches. A la fin de cette mise en œuvre, l'objectif du système sera atteint. Le système n'a pas besoin qu'on lui indique l'objectif lui-même.

L'expression procédurale de l'objectif dans le cas de notre application est le couple (M, T) définie comme suit :

1. M : Modèle de l'environnement
2. T : Assurer aux clients un service d'information, fiable et conforme à la réalité, sur des prestataires de services d'hôtels et de compagnies de transport aériens, ferroviaires et maritimes etc. pour une éventuelle réservation.

5.5.1.1 Moyens de description et processus de développement

MASA-Method [65], adopte la modélisation orientée objet de l'environnement. Pour la caractérisation des objectifs, nous avons adopté la caractérisation procédurale en utilisant les réseaux de Pétri colorés (CPN) pour la modélisation des tâches.

Dans la modélisation des tâches des systèmes, un objet du modèle orienté objet sera une abstraction d'un objet de l'environnement. Les méthodes d'un objet donné modélisent comment peut on agir sur cet objet pour effectuer une certaine opération considérée, dans ce contexte, comme élémentaire. Nous considérons alors les tâches élémentaires comme des invocations de méthodes.

Dans MASA-Method [65], la description d'une tâche se fait comme suit :

1. Transitions : Une transition T représente une invocation d'une tâche élémentaire. Elle est décrite par $O.M$. O est un objet. M est l'une des méthodes de O . Le garde de la transition est une fonction des valeurs des arcs entrants,
2. Arcs d'entrée : Les étiquettes des arcs entrants de la transition T doivent être des paramètres effectifs d'entrée $iPark$ de la méthode M ,
3. Arcs de sortie : Un arc sortant de T est marqué par un paramètre de sortie $oPark$ de la méthode M ,
4. Places d'entrée : Une place P d'entrée d'une transition T contient des paramètres effectifs d'entrée nécessaires à l'invocation de la méthode M dans la transition T ,
5. Places de sortie : Une place de sortie d'une transition T contient des paramètres de sorties effectifs résultat de l'invocation de la méthode M dans la transition T .

Les résultats des évaluations des fonctions des arcs d'entrée sont utilisés en tant que paramètres effectifs d'entrée de la méthode $O.M$. Les évaluations des fonctions d'arcs de sortie $oPar$ utilisent les résultats d'invocation de la méthode $O.M$ comme paramètres variables de ces fonctions. Les résultats des évaluations sont des jetons à ajouter aux places de sortie.

Quand il n'y a pas lieu de confusion, on considère les deux notations (1) $O.M$ où O est un objet et M est une méthode, et (2) P où P est une procédure d'un savoir-faire, sont équivalentes. On suppose alors que P représente une méthode M d'un objet O . Une telle simplification est faite dans le but de simplifier la présentation des explications.

La description des tâches de systèmes comporte une seule table. Cette table comporte :

1. Une description informelle (en langage naturel) de la tâche du système. Le plus souvent, la description informelle n'est pas précise. Elle permet de donner une première idée sur la description formelle,
2. Une description formelle de la tâche du système sous forme d'un CPN sous forme d'une synchronisation de tâches (plus) élémentaire,
3. Une description des éléments du CPN. Pour les places marquées initialement, nous avons inclus ce marquage dans leur description,
4. Une description des tâches élémentaires.

5.5.2 Développement des modèles d'organisations

5.5.2.1 Organisation élémentaire

En utilisant la méthodologie MASA-Method [65] on définit une organisation sous forme d'un couple (SR, CR) comme une instance de l'organisation courante de broker agents où :

1. SR (Set of Roles) : $SR = \{r / r \text{ est un rôle}\}$ ou r est un triplet (SKP, SCP, T)

SKP (Set of Knowledge Procedures): $SKP = \{kp / kp \text{ est une procédure du savoir-faire nécessaire. } kp \text{ est une combinaison d'un ensemble d'actions}\}$.

L'ensemble des actions SAct est défini comme suit :

SAct = {Action / Action est une description d'une actions élémentaire qui sera accomplie sur l'environnement par l'agent qui assumera le rôle r }.

T (Task) : Tâche est la partie de tâche de l'organisation que représente le rôle r dans l'organisation.

SCP (Set of Communication Protocol) : $SCP = \{cp / cp \text{ est un protocole de communication}\}$

$SR = \{r_1, r_2, r_3\}$. $SCP = \{\text{dans notre exemple le protocole de communication entre tous les rôles du SMA est le protocole demande/réponse c'est le protocole utilisé dans l'architecture client/serveur}\}$.

$r_1: (SKP = \{kp_1, kp_2, kp_3, kp_4\}, SCP_{r_1} = (\text{demande/réponse}), T_1 = \{\text{update svc}\})$.

$r_2: (SKP = \{kp_5, kp_6, kp_7, kp_8\}, SCP_{r_2} = (\text{demande/réponse}), T_2 = \{\text{check svc}\})$.

$r_3: (SKP = \{kp_9, kp_{10}, kp_{11}, kp_{12}\}, SCP_{r_3} = (\text{demande/réponse}), T_3 = \{\text{book svc}\})$.

r_1 : représente le rôle `update_provider_svc`.

kp_1 : représente la procédure `update_hotel_svc` qui fournie au broker agent des informations fraîches concernant les hôtels comme la disponibilité, prix, suites, localisation etc.

kp_2 : représente la procédure `update_fly_svc` qui fournie au broker agent des informations fraîches concernant les vols comme l'aérodrome de départ et de destination, la date et l'heure de réservation, le nombre de place disponible etc.

kp3 : représente la procédure `update_train_svc` qui fournit au broker agent des informations fraîches concernant les trains comme la disponibilité, le prix, le départ, l'arrivée etc.

kp4 : représente la procédure `update_boat_svc` qui fournit au broker agent des informations fraîches concernant les bateaux comme la disponibilité, le prix, le départ, l'arrivée etc.

r2 : représente le rôle `check_provider_svc`.

kp5 : représente la procédure `check_hotel_svc` qui permet au broker agent de chercher des hôtels selon des informations spécifiques concernant ces hôtels comme la disponibilité, prix, suites, localisation etc.

kp6 : représente la procédure `check_fly_svc` qui permet au broker agent de chercher des vols selon des informations spécifiques concernant ces vols comme l'aérodrome de départ et de destination, la date et l'heure de réservation, le nombre de place disponible etc.

kp7 : représente la procédure `check_train_svc` qui permet au broker agent de chercher des trains selon des informations spécifiques concernant ces trains comme la gare de départ et de destination, la date et l'heure de réservation, le nombre de place disponible etc.

kp8 : représente la procédure `check_boat_svc` qui permet au broker agent de chercher des bateaux selon des informations spécifiques concernant ces bateaux comme le quai de départ et de destination, la date et l'heure de réservation, le nombre de place disponible etc.

r3 : représente le rôle `book_provider_svc`.

Kp9 : représente la procédure `book_hotel_svc` qui fournit le service de réservation d'un hôtel spécifique selon des paramètres spécifiques comme la disponibilité, prix, suites, localisation etc.

Kp10 : représente la procédure `book_fly_svc` qui fournit le service de réservation d'un vol spécifique selon des paramètres spécifiques comme l'aérodrome de départ et de destination, la date et l'heure de réservation, le nombre de place disponible etc.

Kp11 : représente la procédure `book_train_svc` qui fournit le service de réservation d'un train spécifique selon des paramètres spécifiques comme la gare de départ et de destination, la date et l'heure de réservation, le nombre de place disponible etc.

Kp12 : représente la procédure `book_boat_svc` qui fournit le service de réservation d'un bateau spécifique selon des paramètres spécifiques comme le quai de départ et de destination, la date et l'heure de réservation, le nombre de place disponible etc.

2. CR: $CR = \{(r1, r2) / (r1, r2) \in SR \times SR \text{ et } r1 \text{ est en relation de communication avec le rôle } r2\}$, dans notre cas nous avons l'ensemble de relation de communication $CR : CR = \{(r1, r1), (r1, r2), (r2, r3), (r3, r1)\}$.

Dans MASA-Method [65] les éléments de l'organisation dérivent de l'objectif de l'organisation :

- 1- à ajouter aux tâches élémentaires les rôles qui seront responsables de leur réalisation.
- 2- Définir les protocoles de communication des rôles ajoutés aux tâches élémentaires.

Le résultat est une tâche de l'organisation. Celle-ci est un couple (SpR, Tâche) où :

- 1- SpR : est un ensemble de rôles partiellement définis. Chaque rôle a un identificateur et un ensemble de protocoles de communication. SpR est défini alors comme suit :

$\{ \langle idR, SCP \rangle / idR : \text{Identificateur de rôle, } SCP : \text{Ensemble de protocoles de communication} \}$,

$SpR = \{ \langle r1, SCP_{r1} = (\text{demande}/\text{réponse}) \rangle, \langle r2, SCP_{r2} = (\text{demande}/\text{réponse}) \rangle, \langle r3, SCP_{r3} = (\text{demande}/\text{réponse}) \rangle \}$.

- 2- Tâche est une description de la tâche du système avec des rôles des agents qui accompliront les tâches élémentaires. Elle représentera alors la tâche de l'organisation.

T1: Update svc, T2 : Check svc, T3 : Book svc.
 $T = \{ r1 \text{ play } (T1), r2 \text{ play } (T2), r3 \text{ play } (T3) \}$.

5.5.2.2 Organisations composées

Certaines organisations sont larges et leur développement est difficile. Il est alors nécessaire de décomposer de telles organisations en sous organisations pour pouvoir les maîtriser. L'organisation résultat est dite organisation composée. Une organisation composée est appelée mère relativement à ses sous organisations. Par conséquent, une sous-organisation est appelée fille

d'une organisation composée. Deux sous organisations d'une même organisation composée sont appelées sœurs.

Une sous organisation est, elle aussi, une organisation (élémentaire ou composée). Elle a la particularité d'être un élément d'une organisation composée. En tant qu'organisation, une sous organisation possède une structure et une tâche. En plus, une sous organisation possède un ensemble de rôles dits représentants. De tels rôles représentent la sous organisation dans ses communications avec ses sœurs.

Dans notre cas d'étude l'organisation qui nous intéresse est BAO 'broker agent organization' qui est une sous organisation de l'organisation mère TSO 'tourist service organization' qui englobe elle les trois sous organisations sœur suivantes : UAO 'user agent organization', BAO 'broker agent organization' et PAO 'provider agent organization'.

r1, r2 et r3 sont les rôles de notre organisation BAO sujette d'étude, BC est le rôle représentant de l'organisation.

5.5.2.3 Moyens de description

Le développement des modèles d'organisations comporte les tables suivantes :

1. Description d'un modèle de tâches d'une organisation élémentaire.
2. Description d'un modèle de tâches d'une organisation composée.
3. Description d'un modèle d'organisations élémentaires.
4. Description d'un modèle de rôles.
5. Description d'un modèle d'une organisation composée.

La table de description de la tâche d'une organisation élémentaire comporte :

1. Un identificateur du modèle de la tâche de l'organisation élémentaire,
2. Une référence au modèle de tâches du système que ce modèle de tâches d'organisation implémente,
3. Un modèle de l'environnement qui est le même que celui de la tâche du système,
4. Une référence vers la description (formelle) de la tâche de cette organisation,

5. Une description informelle des rôles impliqués dans la tâche de l'organisation.

La table de description de la tâche d'une organisation composée comporte :

1. Un identificateur du modèle de la tâche de l'organisation élémentaire,
2. Une référence vers le modèle de tâches du système que ce modèle de tâches d'organisation implémente,
3. Un modèle de l'environnement qui est le même que celui de la tâche du système,
4. Une référence vers la description (formelle) de la tâche de cette organisation,
5. Une description informelle des différentes sous-organisations impliquées dans l'organisation composée.

La table de description d'un modèle d'organisations élémentaires comporte :

1. Un identificateur du modèle d'organisations,
2. Une organisation mère si cette organisation élémentaire est une sous-organisation,
3. Une référence vers le modèle de la description (formelle) de la tâche de cette organisation,
4. Une référence vers le modèle de l'organisation substituée dans le cas où cette organisation est une substitution d'une organisation composée,
5. Une description des rôles sous forme de couples (Identificateur de rôle, Référence vers le modèle de ce rôle),
6. Une description de la relation de communication. La relation de communication est décrite de deux façons différentes. Représentation graphique qui est une représentation visuelle. L'expression analytique est décrite sous forme de couples (Rôle émetteur, Rôle récepteur). Elle permet des traitements et des descriptions des implémentations.

La table de description d'une organisation composée comporte :

1. Un identificateur de l'organisation composée,
2. Une référence vers l'organisation mère, si cette organisation est une sous-organisation,
3. Une référence vers la tâche de cette organisation,
4. Une référence vers l'organisation substituée, si cette organisation est une substitution d'une organisation composée,
5. Des représentants : Ils ne seront présents que si cette organisation composée est une sous organisations d'une autre organisation composée,
6. Une description des sous organisations sous forme de triplets (Identificateur de la sous organisation, Les représentants de la sous organisation, Référence vers le modèle de cette sous organisation),
7. Une description de la relation de communication. Cette relation est décrite de deux façons différentes. La première est une représentation graphique visuelle. La seconde est une expression analytique qui décrite sous forme de couples (Représentant de la sous-organisation émettrice, Représentant de la sous organisation réceptrice). La représentation analytique permet des traitements et des descriptions des implémentations.

La table de description d'un modèle de rôles comporte :

1. Un identificateur du modèle de rôles,
2. Une organisation de ce rôle,
3. Un modèle de l'environnement du rôle. Ce modèle sera dérivé de celui de la tâche de l'organisation,
4. Une description (formelle) de la tâche du rôle. Cette tâche sera dérivée de la tâche de l'organisation,
5. Un savoir-faire du rôle : Les identificateurs des procédures du savoir-faire ainsi que leurs paramètres seront dérivés de la tâche du rôle. Cependant, leurs descriptions seront faites par le développeur.

- | | |
|-----|---|
| (1) | R.C : recueil des requêtes auprès des clients et la production d'une demande de réservation. |
| (2) | B.S : assurer aux clients le service de réservation et informer les fournisseurs des réservations effectuées. |
| (3) | S.C : recueil et mise à jour des services fournis par les fournisseurs de services. |

Fig. 21 description informelle de la tâche du système d'application de services pour touristes.

5		Modèle de tâches d'un système	
Identificateur	TSA		
Description informelle	Figure 21		
Description formelle	Modèle de l'environnement	{R, B, S}	
	Modèle de la tâche	Figure 22	
Tâches élémentaires	Identificateur	Description	
	R	Recueil des requêtes auprès des clients et la production d'une demande de réservation.	
	B	Assurer aux clients le service de réservation et informer les fournisseurs des réservations effectuées.	
Légende	Acronyme	Description	
	R	Requests	
	B	Book	
	S	Services	
Commentaires	TSA	Tourists Service Application	

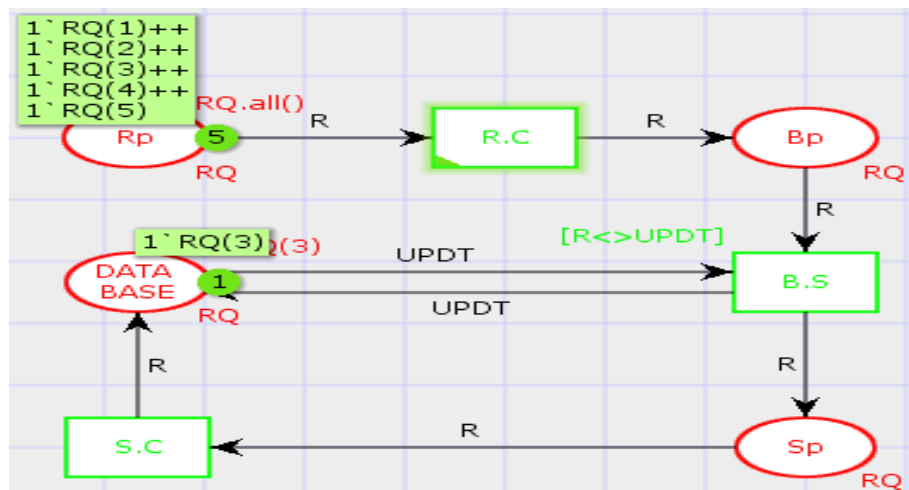


Fig.22 Description formelle de la tâche d'une application de services pour touristes. La description des éléments du CPN est donnée dans le tableau 3.

	Symbole	Description
Places	Rp	Request place
	Bp	Book place
	Sp	Service place
	DB	DATA BASE
Types	RQ	Request
Transitions	R.C	Request collection
	B.S	Book service
	S.C	Services collection
Fonctions d'arcs	R	Request
	UPDT	Update
Grades	R<>UPDT	Request not find in DATA BASE

Tableau 3 Description des éléments du CPN de la figure 22.

(1)	UAO.R.C : recueil des requêtes auprès des clients par la sous organisation UAO et la production d'une demande de réservation.
(2)	BAO.B.S : assurer aux clients le service de réservation par la sous organisation BAO et informer les fournisseurs des réservations effectuées.
(3)	PAO.S.C : recueil et mise à jour par la sous organisation PAO des services fournis par les fournisseurs de services.

Fig.23 Description informelle de la tâche de l'organisation d'application de services pour touristes.

6		Modèle de tâches d'une organisation composée		
Identificateur de la tâche	TSOA			
Tâche du système	TSA (table 5)			
Organisation	TSO (table 7)			
Tâche substituée	Aucune			
Description de la tâche	Modèle de l'environnement	{R, B, S}		
	Modèle de la tâche	Description informelle	Figure 23	
		Description formelle	Figure 24	
	Sous organisation	Identificateur	Description	
		UAO	Sous organisation de recueil des requêtes	
		BAO	Sous organisation de réservation	
PAO	Sous organisation de recueil et mise à jour des services			
Légende	Acronyme	Description		
	UAO	User agent organisation		
	BAO	Broker agent organisation		
	PAO	Provider agent organisation		
Commentaires	///			

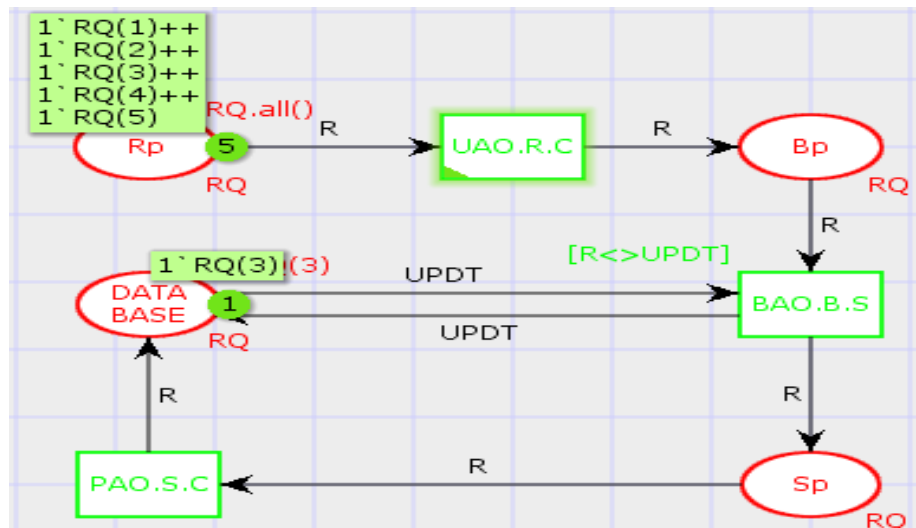


Fig.24 Description formelle de la tâche de l'organisation d'application de services pour touristes. La description des éléments de ce CPN est similaire à celle de la figure 23 donnée dans le tableau 3.

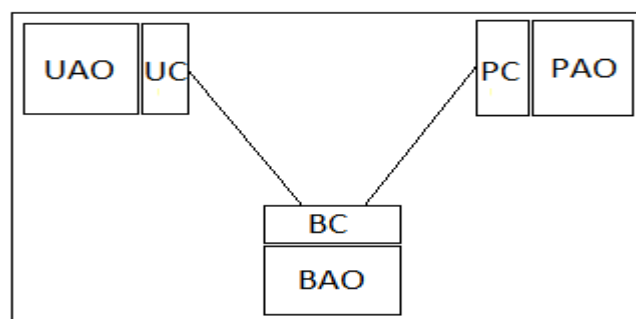


Fig.25 Représentation graphique de la relation de communication entre les représentants des sous organisations de l'organisation d'application de services pour touristes TSOA de la table 14.

- (1) Up.Svc : assurer aux brokers agent une connaissance correcte et fraîche.
- (2) Ch.Svc : vérifier la disponibilité, selon des paramètres spécifiques, d'un service spécifique fourni.
- (3) Bk.Svc : la réservation d'un service spécifique selon des paramètres spécifiques.
- (4) Up.Svc : mettre à jour les informations des fournisseurs.

Fig.26 Description informelle de la tâche d'un système de réservation.

7		Modèle d'une organisation composée		
Identificateur		TSO		
Organisation mère		Aucune		
Tâche		TSOA (Figure 24)		
Organisation substituée		Aucune		
Sous organisation	Identificateur	Représentant	Modèle	
	UAO	UC	Not given	
	BAO	BC	Table 11	
	PAO	PC	Not given	
Relation de communication	Graphique		Analytique	
	Figure 25		Emetteur	Récepteur
			UC	BC
			BC	UC
			PC	BC
BC	PC			
Légende	Symbole	Description		
	UAO	User agent organisation		
	BAO	Broker agent organisation		
	PAO	Provider agent organisation		
	TSO	Tourist service organisation		
	UC	User chief		
	BC	Broker chief		
	PC	Provider chief		
Comment	Cette organisation décrit une organisation d'un processus d'application de service pour touristes			

8		Modèle de tâche d'un système	
Identificateur	TSA		
Description informelle	Figure 26		
Description formelle	Modèle de l'environnement	{Svc}	
	Modèle de la tâche	Figure 27	
Tâches élémentaires	Identificateur	Description	
	Up.Svc	assurer aux brokers agent une connaissance correcte, elle met a jour également les informations des fournisseurs.	
	Ch.Svc	vérifier la disponibilité, selon des paramètres spécifiques, d'un service spécifique fourni.	
	Bk.Svc	la réservation d'un service spécifique selon des paramètres spécifiques.	
Légende	Acronyme	Description	
	Svc	Services	
	Up	Update	
	Ch	Check	
	Bk	Book	
Commentaires	///		

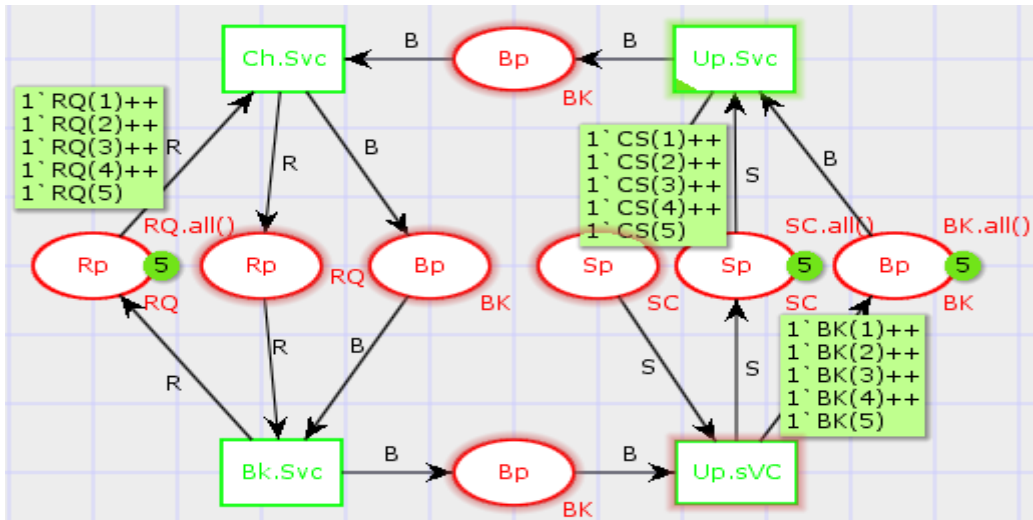


Fig.27 Description formelle de la tâche du système de réservation. La description des éléments du CPN est donnée dans le tableau 4.

- (1) BC.Up.Svc : assurer aux brokers agent une connaissance correcte et fraîche.
- (2) BC.Ch.Svc : vérifier la disponibilité, selon des paramètres spécifiques, d'un service spécifique fourni.
- (3) Bi.Bk.Svc : la réservation d'un service spécifique selon des paramètres spécifiques.
- (4) BC.Up.Svc : mettre à jour les informations des fournisseurs.

Fig.28 Description informelle de la tâche de l'organisation de réservation.

	Symbole	Description
Places	Rp	Request place
	Bp	Book place
	Sp	Service place
Types	///	///
Transitions	Up.Svc	Update services
	Ch.Svc	Check services
	Bk.Svc	Book services
Fonctions d'arcs	R	Request
	B	Book
	S	Service
Grades	///	///

Tableau 4 Description des éléments du CPN de la figure 27.

9		Modèle de tâche d'une organisation élémentaire		
Identificateur	BAOT			
Tâche du système	TSA (table 5)			
Organisation	BAO (table 10)			
Tâches substituées	Aucune			
Description de la tâche	Modèle de l'environnement	{Svc}		
	Modèle de tâches	Informelle	Figure 28	
		Formelle	Figure 29	
	Rôles	Identificateur	Description	
		BC	Broker chief (Head of broker agent team)	
		B ₁	Broker 1	
		B ₂	Broker 2	
...		...		
B _n	Broker n			
Légende	Acronyme	Description		
	///	///		
Commentaires	///			

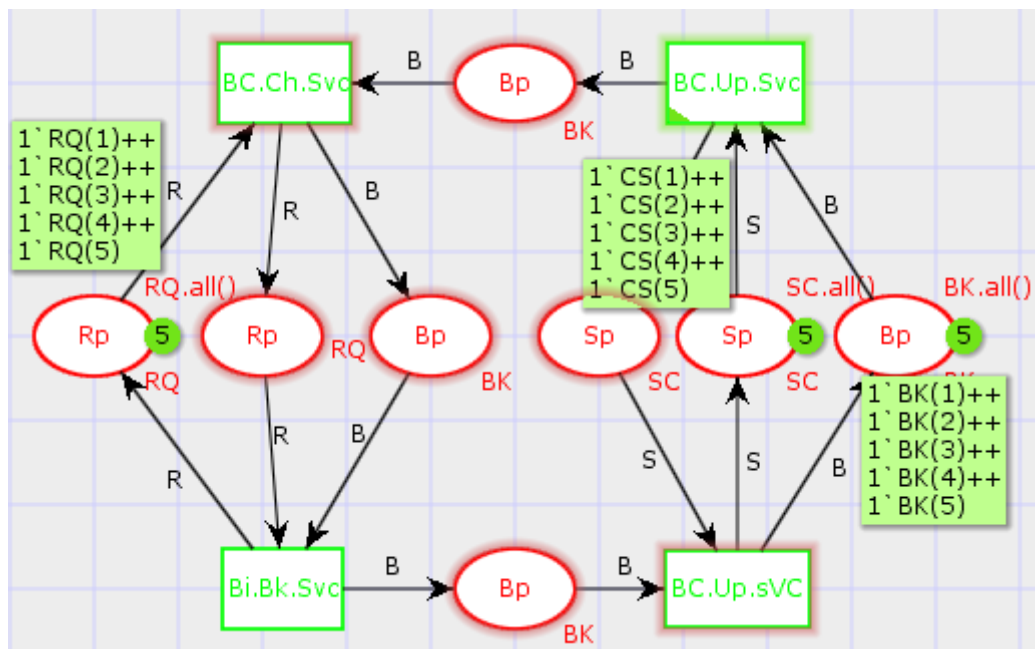


Fig.29 Description formelle de la tâche de l'organisation de réservation. La description des éléments de ce CPN est similaire à celle du CPN de la figure 27 donnée dans le tableau 4.

10		Modèle d'une organisation élémentaire		
Identificateur	BAO			
Organisation mère	TSO			
Tâche	BAOT (table 9)			
Organisation substituée	aucune			
Rôles	Identificateur		Modèle	
	BC		BCM	
	B ₁		B1M (table 11)	
	B ₂		B2M (table 12)	
	
	B ₃		B3M (table 12)	
Relation de communication	Graphique		Analytique	
	Figure 30		Emetteur	Récepteur
			BC	B ₁
			BC	B ₂
		
			BC	B _n
			B ₁	BC
			B ₂	BC
		
B _n	BC			
Légende	Identificateur		Description	
	BAO		Broker agent organization	
	TSO		Tourist services organization	
	BAO-GT		Broker agent organization global task	
	BC		Broker chief	
	B _i / $i \in [1, n]$		book_provider_svc role from 1 to n	
	BCM		Broker chief model	
	BM		Broker model	
Commentaires	Cette sous organisation est également une organisation qui sera utilisée en tant qu'exemple d'une organisation de laquelle nous dérivons les modèles multi-agent et son implémentation dans les phases suivantes de la méthodologie MASA-Method.			

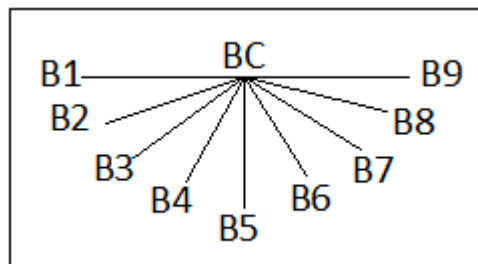


Fig.30 Représentation graphique de la relation de communication entre les rôles de l'organisation des brokers agent.

5.5.3 Développement des modèles de systèmes multi-agent

5.5.3.1 Moyens de description

La table de description d'un modèle de SMA comporte, en plus de la légende et des commentaires :

1. Un identificateur du modèle de SMA,
2. Une référence vers le modèle de l'organisation que ce modèle de SMA va implémenter.
3. Une attribution des rôles : C'est un ensemble d'association de couple (Agent A, ensemble de rôles SR). SR est l'ensemble de rôles que l'agent A va assumer,
4. Des liens de communication : Précisent les liens de communication entre les agents à établir dans une implémentation du modèle de SMA. Ces liens sont présentés de deux façons différentes : La forme graphique et la forme analytique.

La table de description d'un modèle d'agents comporte :

1. Un identificateur du modèle d'agents,
2. Le modèle multi-agent auquel ce modèle d'agents appartient,
3. Le contrôle de l'agent qui peut être intelligent, déclaratif ou procédural,
4. Les rôles que cet agent va assumer. L'ensemble des tâches de ces rôles constitue la tâche du modèle d'agents. Les rôles doivent être énumérés selon leur ordre de priorité. Cet ordre sera utilisé par l'agent pour interpréter les tâches des rôles. L'agent ne passera de l'interprétation de la tâche d'un rôle r1 à celle d'un autre rôle r2 moins prioritaire que s'il n'a rien à faire dans la tâche de r1,
5. Les aspects de communication incluant un ensemble de senseurs, un ensemble d'effecteurs et un ensemble de protocoles de communication,
6. La relation avec l'environnement incluant le modèle de l'environnement, le savoir-faire de l'agent, un ensemble de senseurs et un ensemble d'effecteurs. Le modèle de l'environnement est constitué d'un ensemble de modèles d'objets. Le savoir-faire est un ensemble de

procédure utilisant les senseurs et les effecteurs de l'agent relatifs à l'environnement pour accomplir des objectifs particuliers.

5.5.4 Modélisation multi-agent d'un environnement de services pour touristes (commerce électronique)

5.5.4.1 Organisation d'application de services pour touristes

L'application de services pour touristes dans le commerce électronique peut être vu comme un système dont la description de la tâche est donnée dans la table 5 Page 77. Les tâches élémentaires de la tâche d'application de services pour touristes seront des tâches de trois sous organisations. Le modèle de tâches pour cette organisation est décrit dans la Table 6 Page 78. La description d'un modèle d'organisations d'application de services pour touristes est donnée dans la table 7 Page 80. Il est nécessaire de détailler, ensuite, chaque sous organisation. La sous organisation Broker agent est choisi pour être détaillées. La sous organisation broker agent est une organisation élémentaire. Cette organisation sera utilisée, par la suite, pour illustrer les prochaines étapes de développement d'un modèle de SMA.

Ce modèle est celui d'implémentation de modèles conceptuels. Le modèle de tâches du système est fourni dans la Table 8 Page 80, le modèle de tâches de l'organisation broker agent est donné dans la table 9 Page 82 et, enfin, le modèle d'organisations broker agent est donné dans la Table 10 Page 83. Les modèles de rôles sont décrits dans les Tables 11 Page 86 et 12 Page 87. Notons que les formalismes utilisés pour exprimer les différentes tables de MASA-Method sont autosuffisantes. La majorité des explications sont contenues dans ces tables qui n'exigent pas d'explications complémentaires. Pour cette raison, le reste de cette section ne comporte pas du texte. Il est constitué principalement des tables décrivant le développement d'un modèle de SMA.

5.5.4.2 Développement d'un modèle multi-agent d'application de services pour touristes

Cette étape commence par la dérivation d'un modèle multi-agent. Ceci concerne la description du modèle multi-agent décrit dans la Table 13 Page 88. Les descriptions des modèles d'agents seront présentées dans les tables 14 Page 89 et table 15 Page 90.

11 Modèle de rôles		
Identificateur	BM	
Organisation	BAO	
Modèle de l'environnement	{CONF, D, DISP, H, L, LA, LD, NH, NN, P}	
Tâche	BMT (figure 31)	
Savoir faire	Identificateur	Bk.Svc
	Paramètres	Input : L, NH, D, NN, DISP, P Or Input : D, LD, LA, H, DISP, P Output : CONF
Savoir faire	Description	la réservation d'un service spécifique selon des paramètres spécifiques (D, DISP, H, L, LA, LD, NH, NN, P).
Légende	Symbole	Description
	CONF	Confirmation
	D	Date
	DISP	Disponibilité
	H	Heure
	L	Localisation
	LA	Lieux arrivé
	LD	Lieux départ
	NH	Nom hôtel
NN	Nombre de nuitée	
P	Prix	
Commentaire	///	

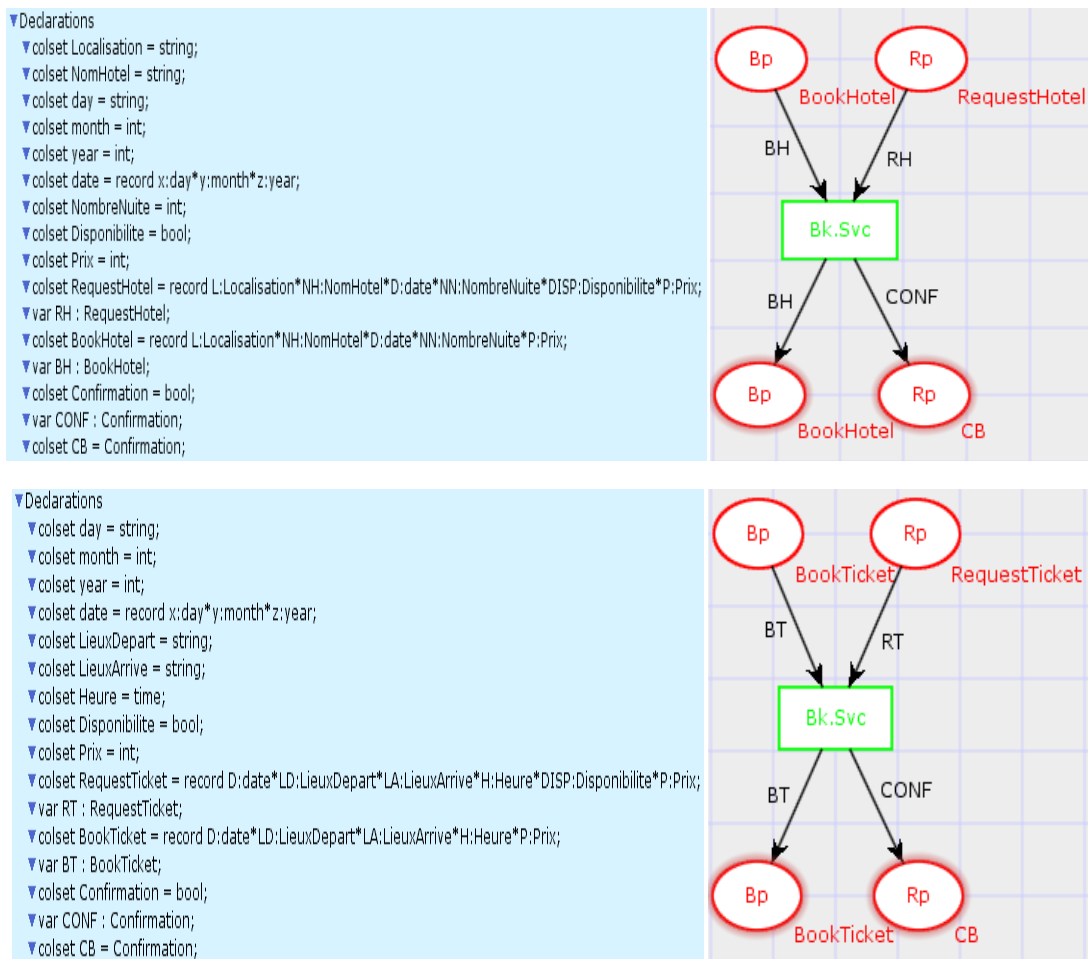
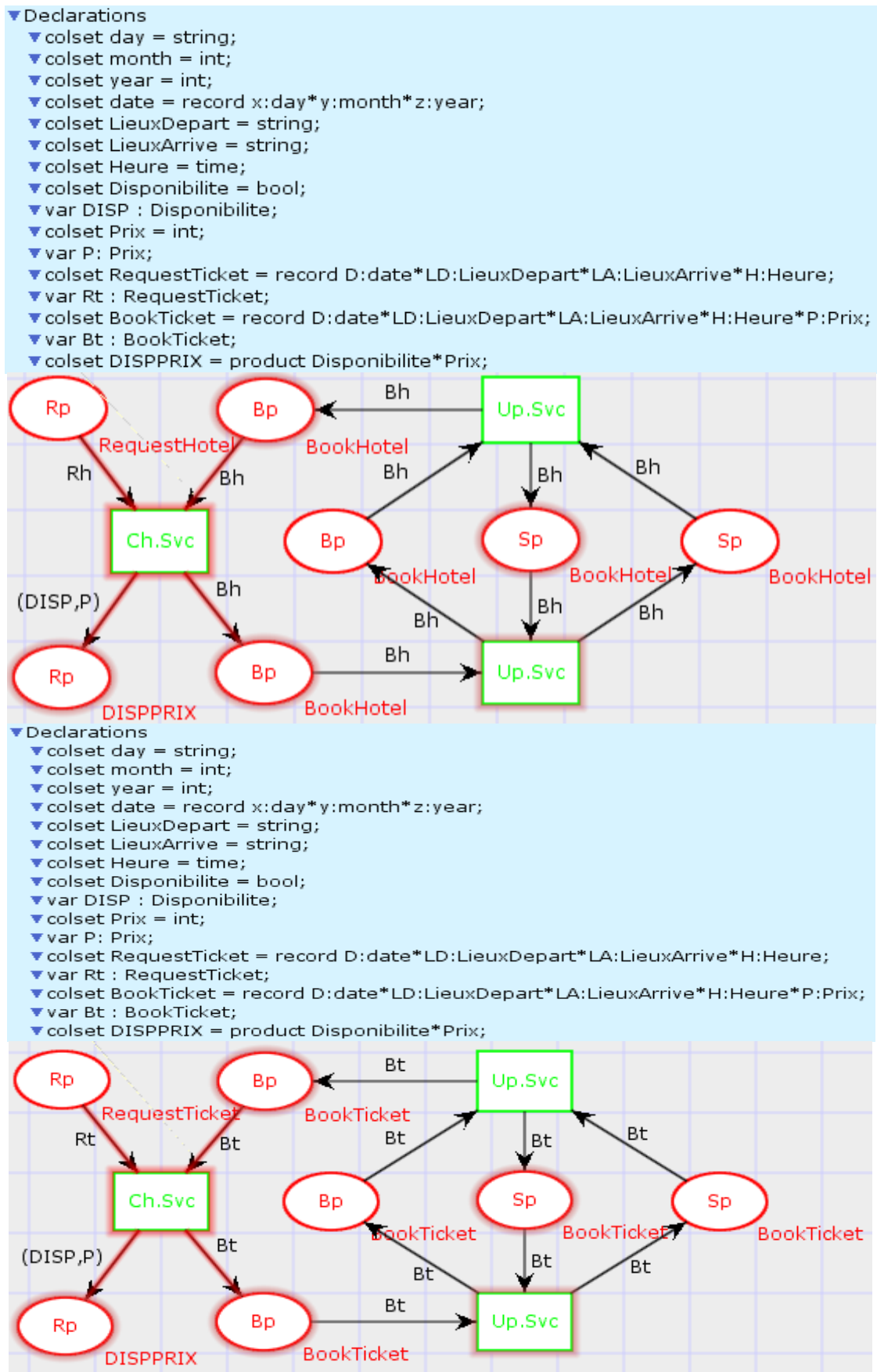


Fig.31 Description formelle de la tâche du modèle de rôles BM.



12	Modèle de rôles		
Identificateur	BCM		
Organisation	BAO		
Modèle de l'environnement	{D, DISP, H, L, LA, LD, NH, NN, P}		
Tâche	BCMT (figure 32)		
	Identificateur	Paramètres	Description
Savoir faire	Up.Svc	Input / output : L, NH, D, NN, P Or Input / output : D, LD, LA, H, P	assurer aux brokers agent une connaissance correcte de tous les paramètres (D, H, L, LA, LD, NH, NN, P).
	Ch.Svc	Input : L, NH, D, NN. Or Input : D, LD, LA, H. Output : DISP, P.	vérifier la disponibilité (DISP), selon des paramètres spécifiques (D, H, L, LA, LD, NH, NN, P), d'un service spécifique fourni.
	Up.Svc	Input / output : L, NH, D, NN, P Or Input / output : D, LD, LA, H, P	mettre à jour également tous les informations des fournisseurs (D, H, L, LA, LD, NH, NN, P) après réservations.
Légende	Symbole	Description	
	D	Date	
	DISP	Disponibilité	
	H	Heure	
	L	Localisation	
	LA	Lieux arrivé	
	LD	Lieux départ	
	NH	Nom hôtel	
	NN	Nombre de nuitée	
P	Prix		
Commentaire	///		

13	Modèle de systèmes multi-agent		
Identificateur	BA-MAS		
Organisation	BAO (table 10)		
Attribution de rôles	Identificateur de l'agent	Rôles	
	Mng	BC	
	Op1	B1, ..., Bq	
	Op2	Bq, ..., Br	
	Op3	Br, ..., Bs	
Lien de communication	Graphique	Analytique	
	Figure 33	Emetteur	Récepteur
		Mng	Op1
		Mng	Op2
Légende	Acronyme	Description	
	Mng	Manager	
	Op1	Operator1	
	Op2	Operator2	
Légende	Op3	Operator3	
	Op3	Operator3	
Commentaire	///		

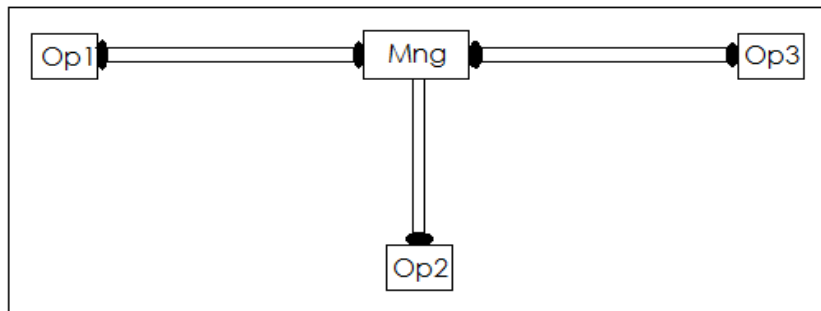


Fig.33 : Représentation graphique des liens de communication entre les agents du système multi-agent de l'équipe de broker agent.

14		Modèle d'agent	
Identificateur	Mng		
Modèle de système multi-agent	BA-MAS		
Contrôle	Intelligent		
Rôle	BC		
Communication	Senseurs	MngOp1S : ..., MngOp3S : ..., MngOp2S : ...,	
	Effecteurs	MngOp1E : ..., MngOp3E : ..., MngOp2E : ...,	
	Protocoles	Demande / réponse	
Relation avec l'environnement	Objets	D : ..., DISP : ..., H : ..., L : ..., LA : ..., LD : ..., NH : ..., NN : ..., P : ...,	
	Savoir-faire	Up.Svc : ..., Ch.Svc : ...,	
	Senseurs	Clavier : ..., Disque : ..., Souris : ...,	
	Effecteurs	Ecran : ..., Disque : ...,	
Légende	Acronyme		Description
	MngOp <i>i</i> S		Senseur de communication de l'agent Mng avec l'opérateur Op <i>i</i>
	MngOp <i>i</i> E		Effecteur de communication de l'agent Mng avec l'opérateur Op <i>i</i>
Commentaires	///		

15	Modèle d'agent	
Identificateur	Bi	
Modèle de système multi-agent	BA-MAS	
Contrôle	procédural	
Rôle	Bi, ..., Bj	
Communication	Senseurs	OpiMngS
	Effecteurs	OpiMngE
	Protocoles	Demande / réponse
Relation avec l'environnement	Objets	CONF :..., D :..., DISP :..., H :..., L :..., LA :..., LD :..., NH :..., NN :..., P :...,
	Savoir-faire	Bk.Svc : ...,
	Senseurs	Disque : ...,
	Effecteurs	Disque : ...,
Légende	Acronyme	Description
	OpiMngS	Senseur de communication avec l'agent Mng
	OpiMngE	Effecteur de communication avec l'agent Mng
Commentaires	///	

Chapitre 6

Mise en œuvre

6.1 Implémentation

6.1.1 Environnement d'implémentation NetLogo

Nous avons choisi le NetLogo comme logiciel d'implémentation puisque c'est un environnement de modélisation programmable, permettant de simuler des phénomènes naturels et sociaux. C'est un logiciel qui s'adapte particulièrement bien à la modélisation de systèmes complexes évoluant au cours du temps.

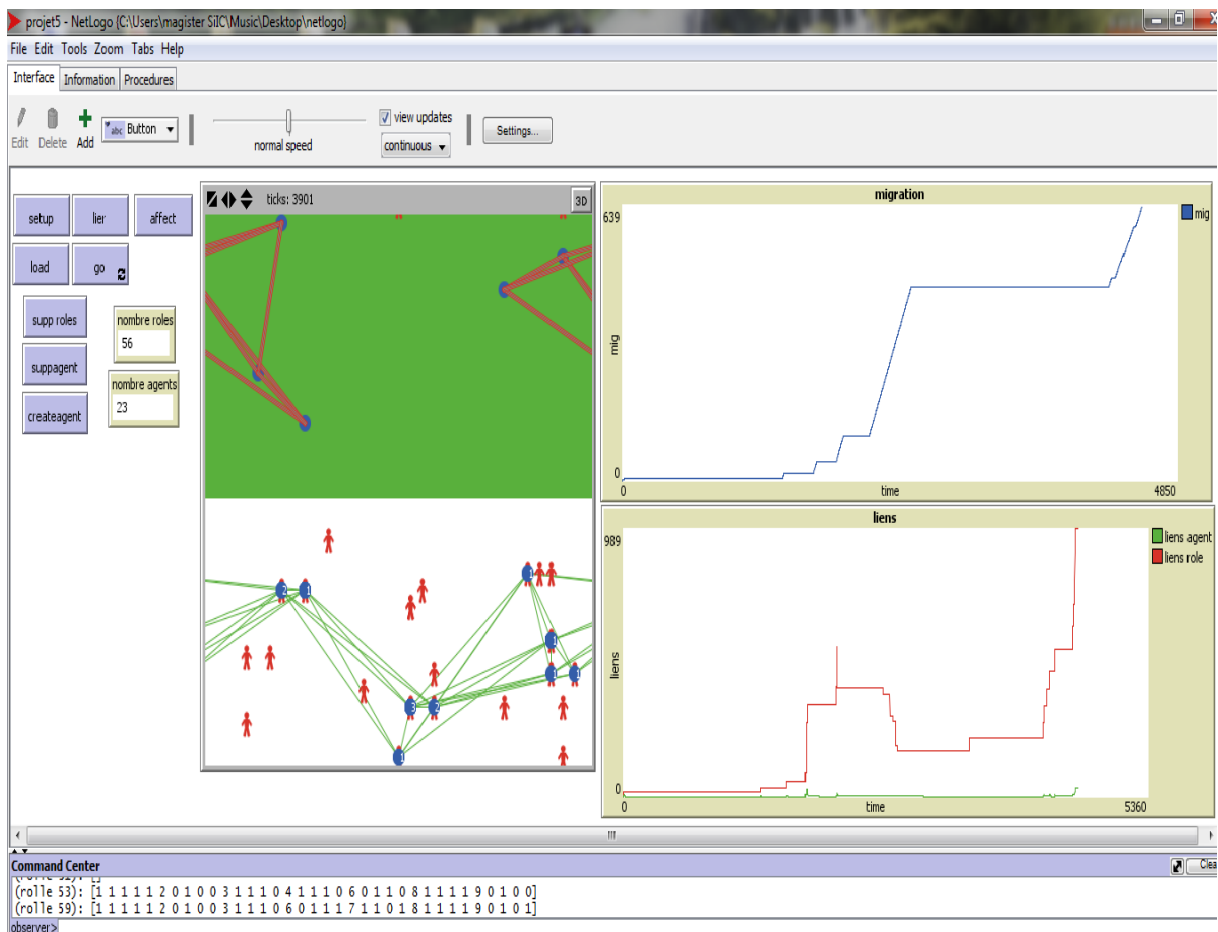


Fig.34 Environnement de notre application de service Pour touristes.

6.1.2 Choix et prise de décision

L'adaptation dans notre approche est réalisée en adoptant une méthode multicritères. En introduisant des contraintes explicites qui peuvent être variés, conflictuelles et de différentes formes, elles seront combinées pour ne constituer, comme résultat de toutes les contraintes, qu'un seul critère.

Pour choisir le rôle candidat à la migration d'un agent source 's' vers un agent destination voisin, nous appliquons la relation de dominance aux multicritères spécifiés en dessous. Cette même relation de dominance sera utilisée pour déterminer l'agent de destination, entre les voisins, vers qui le rôle sélectionné va migrer. Dans notre exemple d'application de broker agent on considère les critères suivant :

- (f1) : déterminer pour chaque voisin de 's' s'il fourni ou non un nombre minimal exigé de services de touristes considéré comme un seuil minimal exigé.
- (f2) : calculer pour chaque voisin de 's' le nombre de services de touristes fourni qui diffèrent de ceux fourni par l'agent source 's'.
- (f3) : déterminé pour chaque voisin de 's' s'ils ont encore de l'espace mémoire libre ou non.
- (f4) : calculer pour tout les voisins de 's' le nombre de rôles assumés pour les comparer après.

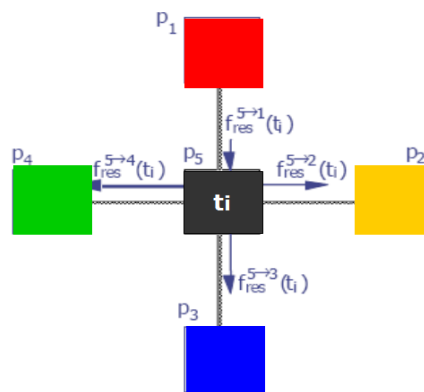


Fig.35 Choix et prise de décision.

6.1.3 Etude du comportement

A l'Initialisation on assume qu'un agent n'assume qu'un seul rôle.

Une évolution qui préserve la cohérence de l'organisation requière une vérification de cohérence de certain contraintes (avant la migration des

rôles) et une cohérence de rétablissement pour d'autres contraintes (après migration des rôles).

Comme vérification préventive de cohérence

- Le rôle r_i / $i \in \{1, 2, 3\}$ sera délégué d'un agent initial source 's' = B_x à un autre agent B_y voisin de B_x si l'agent B_x fourni un nombre minimal de services de touristes. Cette restriction impose la condition qu'un agent doit fournir un nombre minimal de services de touristes (considéré comme un seuil minimal) pour pouvoir assumer un rôle r_i .
- Le rôle r_i / $i \in \{1, 2, 3\}$ sera délégué d'un agent initial source 's' = B_x à un autre agent B_y voisin de B_x si le nombre de services de touristes fourni par l'agent B_y qui sont différent de celui fourni par B_x et qu'il soit supérieur d'un seuil qui sera fixé avant de commencer le mécanisme d'adaptation.
- Chaque agent a un nombre maximum de rôles qui peut assumer, il ne peut pas recevoir plus de rôles que le maximum toléré. Cette restriction vise à maintenir un équilibre dans l'allocation des rôles.

Comme contrainte corrective

- Après migration de chaque rôle on doit mettre à jour les relations entre les agents pour maintenir le SMA correct a tout moment.

Exemple : Plays (B_5, r_1): B_5 plays role r_1 , Plays (B_3, r_2): B_3 plays role r_2 .

And (r_2, r_1): r_2 played by B_3 is in relation with r_1 played by B_5 .

Si r_1 migre de B_5 vers B_6 on a : on supprime d'abord la relation (r_2, r_1) de B_3 vers B_5 et on ajoute la relation (r_2, r_1) de B_3 vers B_6 .

6.1.4 L'adaptation dans notre environnement

Le commerce électronique est un environnement très dynamique qui change de façon continu, en général il obéit a la loi de 'l'offre et la demande' c'est cette loi qui contrôle le marché de commerce, des paramètres aussi variables nécessite un système capable d'adapter ces fonctionnalités au changement de l'environnement.

Dans notre exemple la demande des user agent change à chaque arrivée d'un nouveau client, cela implique nouvelle requête de réservation pour broker agent, d'autres client par contre peuvent changer ou annuler leurs réservation initial. Aussi les fournisseurs changent leurs services continuellement cela implique nouvelle information à mettre à jours pour les brokers agent : à n'importe quel moment les hôtels, les vols, les trains ou les bateaux complets peuvent devenir disponibles ou l'inverse. Ils peuvent

également être annulés cause météo, maintenance ou autres. L'heure de départ peut changée de nouveau services peuvent être programmés. En plus de nouveau broker agent peuvent rejoindre le SMA initial et autres peuvent le quitter. Tous ces changements doivent être pris en considération pour l'adaptation de notre SMA et fournir un système robuste qui ne va pas tombé en panne même dans un environnement dynamique.

6.1.5 Expérimentations et résultats

Pour valider notre approche multi-agent basée-rôles pour systèmes adaptatifs, une application de services pour touristes en commerce électronique, basée sur notre approche multicritères, a été développée et simulée dans un environnement Netlogo permettant de voir l'évolution du système au cours du temps.

6.1.5.1 Cas d'étude

L'environnement de simulation est un monde composé d'agent broker ayant pour mission de prendre en charge les demandes de services des clients en ligne auprès des fournisseurs de ces services (établissement hôtelier, compagnies aérienne, ferroviaire et maritime...etc.) dans un environnement dynamique et très instable. Ces agents broker ont une capacité de mémoire pour sauvegarder et mettre à jour les informations sur la disponibilité des services auprès des fournisseurs.

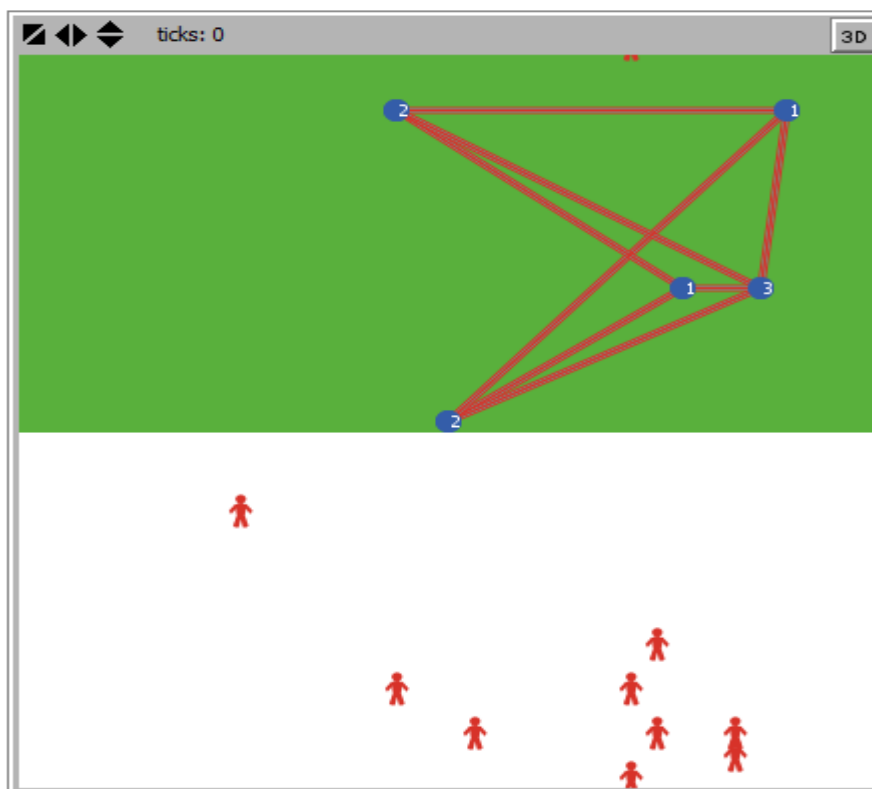


Fig.36 établissement des liens en rouge entre les rôles en bleus.

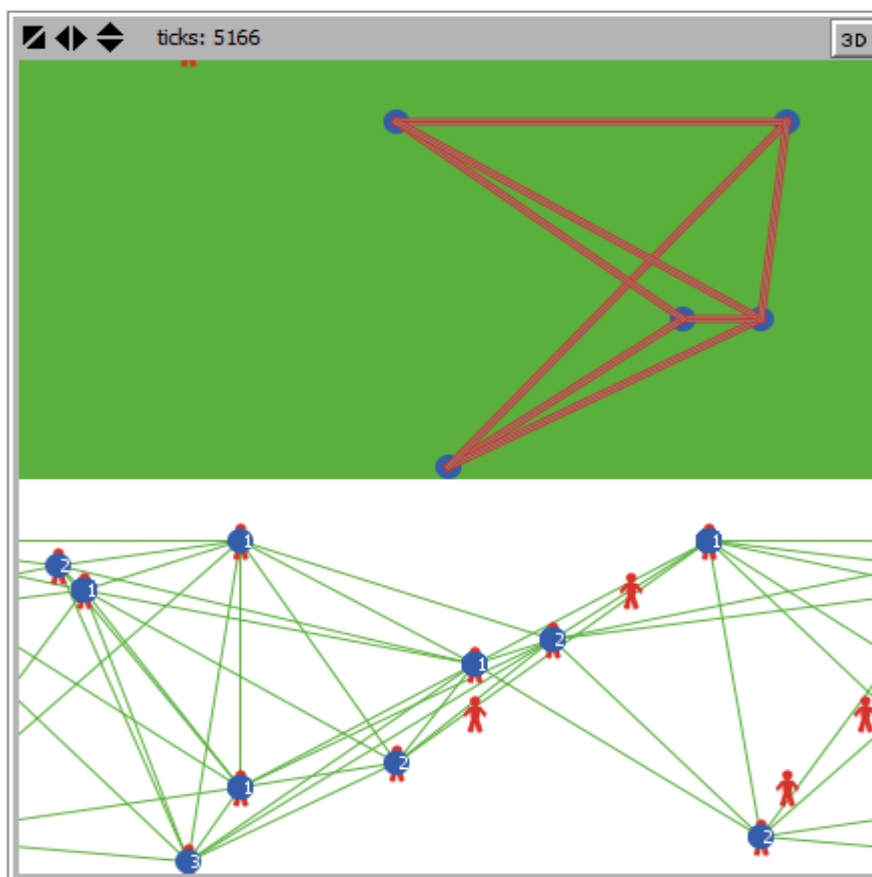


Fig.37 affectations des rôles (1, 2, 3) en bleus aux agents broker en rouge
En établissant les liens correspondant entre les agents.

Le graphe ci-dessous représente la comparaison des liens entre les rôles, qui sont fixes, et ceux entre les agents qui évoluent au cours du temps.

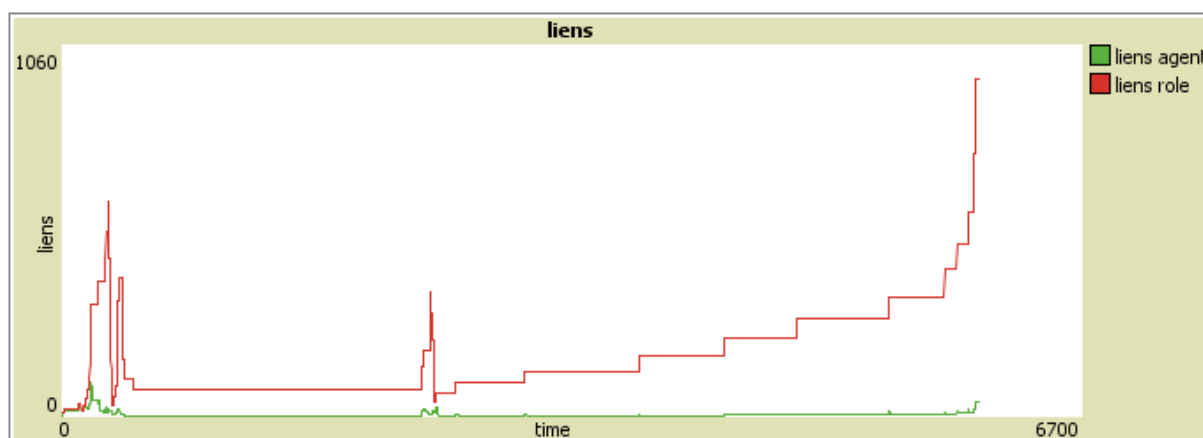


Fig.38 comparaison de la courbe des liens entre les agents et la courbe des liens entre les rôles.

Notre approche multicritères permet de prendre en considération plusieurs paramètres à la fois, permettant ainsi de prendre en compte différents aspects et contraintes dans le comportement des agents broker. D'où le graphe ci-dessous représentant le taux de migrations des rôles entre les agents.

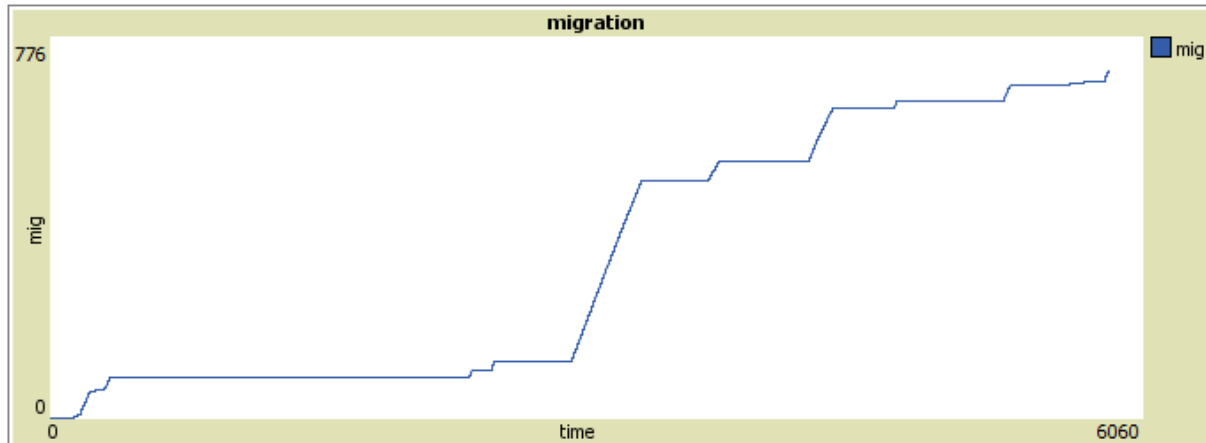


Fig.39 courbe du taux de migration des rôles au cours d'évolution.

6.1.5.2 Emergence d'affectation des rôles

Après une batterie de test nous constatons que le comportement de migration des rôles a généralement tendance à préférer de migrer vers les agents qui ont le plus de liens '**liensag**' avec les autres agents du monde. Tandis que la quantité d'information des services garantis par les fournisseurs et sauvegardée par l'agent, l'espace de stockage encore disponible et le nombre d'information différentes dont dispose l'agent n'influent pas beaucoup sur le l'émergence de la réorganisation des rôles. Le système reste stable (pas d'émergence) lorsque les agents ont un nombre de '**liensag**' plus ou moins identique.

Les tests nous ramènent à dire que l'agent dominant est généralement celui qui possède le plus de liens avec les autres agents du système. La relation de dominance conduit donc les agents broker à un comportement qui privilégie l'affectation des rôles dominant à l'agent qui a le plus de contacts avec ces semblables. Nous pouvons qualifier ce comportement d'émergent parce que les agents n'ont pas une information ou une conscience sur le nombre de contacts que son voisin '**neighbors**' peut avoir avec les autres agents du système et que ce paramètre ne figure pas comme critère explicite pour le choix de l'agent destination qui va recevoir le rôle candidat à la migration.

Et justement ce comportement émergent permet une meilleure performance pour notre application puisque un agent qui a plusieurs contacts a beaucoup plus de chance de satisfaire la demande de nos clients.

6.1.5.3 Discussions

Après affectation des rôles aux agents le nombre de liens entre agents et rôles est identique. Le nombre des liens entre les rôles est toujours supérieur ou égal à celui entre les agents mais jamais inférieur à celui-ci et cela tout au long de l'évolution du système.

En ajoutant de nouveaux rôles à notre monde, le nombre de liens qui s'ajoute entre agents '**liensag**' en conséquence augmente toujours avec un taux inférieur ou égal par apport au nombre des liens qui s'ajoute entre les rôles '**liens**'. En ajoutant des rôles continuellement le nombre de liens entre agent va atteindre un nombre maximal, atteignant ce niveau le nombre de liens entre agents reste fixe même si le nombre de liens entre rôles lui ne cesse d'augmenté. Si le nombre de rôles est fixe le nombre de liens entre rôles '**liens**' reste fixe tandis que le nombre des liens entre agents '**liensag**' lui peut accroitre, baisser ou rester stable mais il reste en tout cas inférieur ou égal au nombre de '**liens**'.

Conclusion et perspectives

I. Conclusion

Ce mémoire décrit notre approche pour le développement d'un mécanisme d'adaptation décentralisé, pour résoudre le problème d'auto-organisation de société d'agents basée organisation des rôles.

Nous avons d'abord commencé par la définition d'agent et de système multi-agent, puis nous avons introduit la notion d'organisation et de réorganisation dans les systèmes multi-agents. Pour examiner ensuite les méthodes existantes décrivant l'auto-organisation de société d'agent (Chapitre 3).

Nous avons établi dans un deuxième temps (Chapitre 4) un survol des travaux en relation avec notre sujet et plus précisément, on s'est focalisé sur les présents approches d'adaptation basé organisation des rôles. Cela consiste d'abord en une exploration horizontale de ces travaux, ou pour chaque travail nous avons précisé le 'quoi' du travail et le 'pourquoi' ainsi que 'l'approche' utilisée pour résoudre le problème. Puis une exploration verticale (plus détaillée et plus approfondie) de ces travaux en plus du 'quoi' du travail et le 'pourquoi' ainsi que 'l'approche' utilisée pour résoudre le problème, nous avons ajouté le 'comment de' la réalisation qui a été faite, quels sont les 'outils et technologies utilisées ainsi que l'identification des' applications' utilisées pour 'l'évaluation' de ces travaux.

En fin nous avons identifié les grands projets, laboratoire et équipes de recherche en relation avec notre travail en décrivant ces projets, équipes et laboratoire de façon succincte ainsi que leurs missions globales. Nous avons également proposé une classification des différents travaux en se basant sur quelques critères de comparaison.

Dans un troisièmes temps (chapitre5), nous avons présenté notre contribution pour résoudre le problème d'adaptation des systèmes multi-agents basés organisation des rôles. Ce qui rend notre approche plus efficace ce sont les traits multicritères, puisqu'elle s'inspire du travail [67] qui porte sur le développement d'organisation des systèmes multi-agents basés rôles. Il propose une approche d'optimisation d'organisation OP (Organization Partitioning) en prenant l'algorithme PA (Particles Approach [68]), qui l'adapte au partitionnement des rôles entre les agents du système. Nous allons à notre tour prendre l'approche d'optimisation d'organisation [67] pour l'appliquer aux systèmes multi-agents adaptatifs.

Pour valider l'approche proposée nous avons imaginé une description d'un 'scénario' que nous avons utiliser par la suite comme application, ou notre choix s'est porté sur un scénario d'application dans le domaine du

commerce électronique. C'est une application de services pour touristes, qui consiste à assurer aux clients un service d'information, fiable et conforme à la réalité, sur des prestataires de services d'hôtels et de compagnies de transport aériens, ferroviaires et maritimes etc. pour une éventuelle réservation. Dans la phase de conception où nous décrivons la méthodologie de développement de notre exemple SMA, et pour cela nous avons choisi la méthodologie MASA-Method [65] comme processus méthodologique qui est une méthodologie complète.

Contrairement à d'autres approches de réorganisation qui sont généralement focalisées sur un seul paramètre par exemple à maximiser une fonction d'utilité ou encore à minimiser une fonction de coût de réorganisation. L'adaptation dans notre travail se fait en adoptant une méthode multicritères en introduisant des contraintes explicites, variées et qui peuvent être conflictuelles combinées ensemble pour ne constituer qu'un seul critérium qui est le résultat de la force, ce qui nous permet de prendre en considération, d'une façon beaucoup plus large et plus complète, plusieurs paramètres dans la mise en œuvre de l'adaptation du système. Ces critères donc seront calculés et mis-à-jour à chaque cycle d'exécution pour servir à la résolution de conflit pour une meilleure coopération des agents de l'organisation

L'approche que nous avons proposée reste ouverte et peut être appliquée dans différents domaines (industrie, contrôle aérien, transport etc.). Elle est facilement implémentée dans un environnement de modélisation de systèmes complexes (simulant des phénomènes naturels et sociaux) évoluant au cours du temps.

II. Perspectives

Il est à noter qu'un effort doit être fait pour prendre en considération l'aspect d'inter-blocage qui peut survenir lors des réservations. Effectivement deux clients peuvent solliciter le même service, avec exactement les mêmes paramètres et en même temps, par le biais de deux agents distincts. Nous prévoyons prendre en charge cet aspect dans un travail futur.

Bibliographies

[1]Bond, A. H. and Gasser, L., "Readings in Distributed Artificial Intelligence", Morgan Kaufmann, Burlington, Massachusetts, 1988.

[2]Huaglory Tianfield "A New Framework of Holonic Self-Organization for Multi-Agent Systems" Huaglory Tianfield, Member, IEEE, 2007 IEEE.

[3]Eric Malville "L'auto-organisation de groupes pour l'allocation de tâches dans les Systèmes Multi-Agents: Application à CORBA" ; Thèse 1999 Université SAVOIE France.

[4]G. Picarda J. F. Hübnera O. Boissiera M.-P. Gleizesb "Réorganisation et auto-organisation dans les systèmes multi-agents"

[5]Georgé J-P. Gleizes, M-P., Glize, P., "Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie Amas", Revue RIA n°4/2003, 2003.

[6]Hans-Ulrich Heiss, Marcus Dormanns, "Mapping Tasks to Processors with the Aid of Kohonen Networks" 1994.

[7]J. Ferber. Les systèmes multi-agents, vers une intelligence collective. InterEditions, 1995.

[8]N. R. Jennings, M. Wooldridge, and K. Sycara. A roadmap of agent research and development. Int Journal of Autonomous Agents and Multi-Agent Systems, 1(1) :7- 38, 1998.

[9]B. Chaib-draa. 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.

[10]B. Chaib-draa. Industrial applications of distributed AI. Communications of the ACM, 38(11) :49-53, 1995. [BCD, 1994] B. Chaib-draa. Interaction between agents in routine, familiar and unfamiliar situations. International Journal of Intelligent and Cooperative Information Systems, 1(5):7-20, 1996.

[11]B. Moulin and B. Chaib-draa. An overview of distributed artificial intelligence. In G. M. P. O'Hare and N. R. Jennings, editors, Foundations of Distributed AI, pages 3-54. John Wiley & Sons : Chichester, England, 1996.

- [12]F. Bergenti, M.P. Gleizes, and F. Zambonelli. Methodologies and Software Engineering for Agent Systems. Kluwer, 2004.
- [13]J. S. Sichman, R. Conte, Y. Demazeau, and C. Castelfranchi. A social reasoning mechanism based on dependence networks. In Proceedings of the 11th European Conference on Artificial Intelligence, pages 188–192, 1994.
- [14]Schmidt, K, " Cooperative Work: A Conceptual Framework ". In Distributed Decision Making: Cognitive Models for Cooperative Work, J. Rasmussen, B. Brehmer and J. Leplat (Eds.), John Wiley & Sons Ltd, pp. 75-110, 1991
- [15]Morin E. " La méthode (1) : La nature de la Nature. " Le seuil; 1977.
- [16]Fox, M.S. " An organizational view of distributed systems ". IEEE Trans. Syst., Man, Cybern., SMC-11, pp70-80, 1981.
- [17]Groupe MARCIA. Auto-organisation := Evolution de structure(s); Actes de la Journée PRC-GDR Intelligence Artificielle sur le thème des Systèmes Multi-Agents, Toulouse, 2 février 1996.
- [18]Rasmussen, J., " Modelling Distributed Decision Making. " In Distributed Decision Making: Cognitive Models for Cooperative Work, edited by J. Rasmussen, B. Brehmer and J. Leplat, John Wiley & Sons Ltd, 1991.
- [19]Ünsal Cem "Self-organization in large populations of mobile robots" Master of Science in Electrical Engineering, May 1993, Blacksburg, Virginia <http://armyant.ee.vt.edu/unsalWWW/cemthesis.html>
- [20]Varela Fransico J. « Autonomie et connaissance :essai sur le vivant ». Edition du seuil 1988.
- [21]Krippendorff Klaus, « Dictionary of Cybernetic », 27 janvier 1997, <http://pespmcl.vub.ac.be/SELFORG.html>.
- [22]Marcenac Pierre, Calderoni Stéphane, « Self-organisation in Agent-Based Simulation », Proceedings of MAAMAW'97, Ronneby, Sweden, May 1997 <http://www.univ-reunion.fr/~marcenac/Publications/MAAMAW97.ps>.
- [23]Bonabeau Eric, Theraulaz Guy, " Auto-organisation et comportement collectifs: la modélisation des societies d'insectes" dans " Auto-organisation et comportement" Coordinateurs Guy Théraulaz et François Spitz – Editions Hermès 1997 -). 91 – 140
- [24]Christine Piquemal-Baluard, Pierre Glize. Des aptitudes non cognitivistes d'agents pour l'auto-organisation. Actes de la Journée PRC-GDR Intelligence Artificielle sur le thème des Systèmes Multi-Agents, Toulouse, 2 février 1996.

- [25] Reid G. Smith. The contract net protocol : Highlevel communication and control in a distributed problem solver. *IEEE Transaction on Computers*, 29(12) :1104–1113, 1980.
- [26] COLLINOT et A. DROGOUL : La méthode de conception multi-agent CASSIOPEE : application à la robotique collective. *Revue d'intelligence artificielle*, 12(1):125–147, 1998.
- [27] DROGOUL : Systèmes multi-agents situés. Mémoire d'habilitation à diriger des recherches, Université Paris 6, 2000.
- [28] F. Zambonelli, M. Wooldridge. Developing Multiagent Systems : The Gaia Methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3): 317–370, 2003.
- [29] M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
- [30] ZOUACHE Djaafar, AGENTS AUTONOMES POUR CONSTRUIRE DES SYSTEMES ADAPTATIFS, Université Mentouri de Constantine Soutenu le 26 /12 / 2008.
- [31] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7 :83–124, 1997.
- [32] V. Lesser, K. Decker, T. Wagner, N. Carver, A. Garvey, B. Horling, D. Neiman, R. Podorozhny, M. NagendraPrasad, A. Raja, R. Vincent, P. Xuan, and X.Q.. Zhang. Evolution of the gpgp/taems domain independent coordination framework. *Autonomous Agents and Multi-Agent Systems*, 9(1) :87–143, July 2004. Kluwer Academic Publishers.
- [33] J. F. Hübner, J. S. Sichman, and O. Boissier. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence (SBIA'02)*, volume 2507 of LNAI, pages 118–128. Springer, 2002.
- [34] M. Esteva, J. A. Rodriguez-Aguiar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specification of electronic institutions. In *Proceedings of the Agentmediated Electronic Commerce*, LNAI 1191, pages 126–147, Berlin, 2001. Springer.
- [35] J. Ferber and O. Gutknecht. A meta-model for the analysis and design of organizations in multiagents systems. In *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS' 98)*, pages 128–135. IEEE Press, 1998.

- [36] Amiguet, M. MOCA: un modèle componentiel dynamique pour les systèmes multi-agents organisationnels, thèse de doctorat 2003, Université de Neuchâtel (Switzerland).
- [37] J.F. Hübner, J. S. Sichman et O. Boissier. Spécification structurelle, fonctionnelle et déontique d'organisations dans les SMA. Dans J.-P. Müller (éditeur), JFIADSMA'02. Hermes, 2002.
- [38] Schillo, M., "Self-organization and adjustable autonomy: Two sides of the same medal?", In H. Hexmoor and R. Falcone (Eds.), Proceedings of the AAAI 2002 Workshop on Autonomy, Delegation and Control: From Inter-agent to Groups. Connection Science 14(4): pp. 345-360, 2003.
- [39] Schillo, M., Bürckert, H., Fischer, K. and Klusch, M., "Towards a definition of robustness for market-style open multiagent systems", Proceedings of the 5th International Conference on Autonomous Agents (AA'2001), Montreal, Canada, 2001, New York, pp.75 -76.
- [40] E Yu, "Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering ", Proc. Of 3rd Int. Symp. on Requirements Engineering, 1997.
- [41] F. Zambonelli, N. Jennings, M. Wooldridge, "Organizational Abstractions in the Analysis and Design of Multi-agent Systems", Proc. Of 1st International Workshop on AOSE 2000.
- [42] Xinjun Mao, Ji Wang and Jiajia Chen, "Modeling Organization Structure of Multi-Agent System", Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05).
- [43] Wooldridge, M., Jennings, N. R., & Kinny, D. (1999). A methodology for agent-oriented analysis and design. In Proceedings of the third annual conference on autonomous agents (pp. 69–76).
- [44] M. Wooldridge, N. R. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
- [45] J. Ferber and O. Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In *Third International Conference on Multi-Agent Systems*, pages 128–135. IEEE Computer Press, 1998.
- [46] J. Ferber, O. Gutknecht, and M. Fabien. From agents to organizations: An organizational view of multi-agent systems. In P. G. et al. Editor, *AOSE 2003*, number LNCS 2935, pages 214–230, 2004.

- [47]G. Cabri, L. Ferrari, and L. Leonardi. Brain: A framework for flexible role-based interactions in multiagent systems. In Proceedings of the 2003 Conference on Cooperative Information Systems (CoopIS), 2003.
- [48]GAUTHIER PICARD,' Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente'.thèse soutenue a l'Université Paul Sabatier de Toulouse III le 10 décembre 2004
- [49]G. Cabri, L. Ferrari, and L. Leonardi. Agent role-based collaboration and coordination: a survey about existing approaches. In Proc. Of the Man and Cybernetics Conf., 2004.
- [50]Eric Matson, Scott DeLoach,' Enabling Intra-Robotic Capabilities Adaptation Using An Organization-Based Multiagent System'. Proceeding of the 2004 IEEE Int.matlonai Conference on Robotics Automation New Orleans, U\ * April 2004.
- [51]Lucia R.D. Bastos, Jaelson F. B. Castro 'From Requirements to Multi-Agent Architecture Using Organizational Concepts' May 15–16, 2005, St. Louis, Missouri, USA. Copyright 2005 ACM.
- [52]Mirko Viroli, Alessandro Ricci, Andrea Omicini,' An Organization Infrastructure for Multi-Agent Systems based on Agent Coordination Contexts'. AAMAS'05, July 25–29, 2005, Utrecht, Netherlands. Copyright 2005 ACM
- [53]Zhiyong Zhang, Jiexin Pu and Hongyi Zhang 'Collaboration Access Control Model for MAS Based on Role and Agent Cooperative Scenarios', Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation June 25 - 28, 2006, Luoyang, China.
- [54]Xiaoqin Zhang, Haiping Xu, Bhavesh Shrestha 'Developing Multi-Agent Systems with Automatic Agent Generation and Dynamic Task Allocation Mechanisms' AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.Copyright 2007 IFAAMAS.
- [55]Scott A. DeLoach · Walamitien H. Oyen · Eric T. Matson.'A capabilities-based model for adaptive organizations' Auton Agent Multi-Agent Syst (2008).
- [56]R. Kota, N. Gibbins and N. R. Jennings ,'Self-Organising Agent Organizations', , Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX. Copyright _c 2009, In

- [57]Roman Neruda, Ondřej Kazík 'Role-Based Design of Computational Intelligence Multi-Agent System'. MEDES'10 October 26-29, 2010, Bangkok, Thailand.
- [58]Centeno, R., R. Hermoso, V.T. da Silva. 2010. Role evolution in open multi-agent systems as an information source for trust. In Proceedings of the 9th international conference on autonomous agents and multiagent systems (AAMAS 2010). Toronto: ACM.
- [59]DANNY WEYNS, ROBRECHT HAESEVOETS, and ALEXANDER HELLEBOOGH,' The MACODO Organization Model for Context-Driven Dynamic Agent Organizations' ACM Transactions on Autonomous and Adaptive Systems, Vol. 5, No. 4, Article 16, Pub. Date: November 2010.
- [60]Haibin Zhu, Senior Member, IEEE 'Group Role Assignment with Conflicting Agent Constraints' 2011 IEEE.
- [61]M. Becht, T. Gurzki, J. Klarmann, M. Muscholl,' ROPE: Role Oriented Programming Environment for Multiagent Systems'. University of Stuttgart Breitwiesenstrasse 20-22 D-70565 Stuttgart.
- [62]Juan M. Alberola, Vicente Julian, and Ana Garcia-Fornes. 'Cost-Aware Reorganization Service for Multiagent Systems', ITMAS 2011: Infrastructures and Tools for Multiagent Systems.
- [63]Juan M. Alberola, Vicente Julian, and Ana Garcia-Fornes. 'Multidimensional Adaptation in MAS Organizations'. IEEE TRANSACTIONS ON CYBERNETICS, VOL. 43, NO. 2, APRIL 2013.
- [64]Juan M. Alberola · Vicente Julian · Ana Garcia-Fornes. 'Challenges for adaptation in agent societies' Springer-Verlag London 2012.
- [65]Lahlouhi Ammar, Thèse de Doctorat d'état 'Modélisation multi-agent du Processus logiciel' Université Mentouri – Constantine Juin 2006.
- [66]Lahlouhi, Ammar and Z. Sahnoun, 'Multi-Agent Methodologies. Incoherencies.', J. Debenham, B. Henderson-Sellers, N. Jennings & J.Odell (Eds). Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies. ISBN 0-9581915-0-6. COTAR (2002) soft binding.
- [67]Lahlouhi Ammar 'Organizations Partitioning Optimization' The 9th International Conference on Autonomous Agents and Multiagent Systems May 10-14, 2010 Toronto, Canada.
- [68] Heiss, H.-U. and Schmitz, M.: Decentralized Dynamic Load Balancing: The Particles Approach. Information Sciences 84(1-2), pages 115-128 (1995)

[69] A Baciú, A Nagy, "Coordination and Reorganization in Organizational Multi-Agent Systems" 2003 Suisse.

[70] Y. So and E. H. Durfee. An organizational self design model for organizational change. In Proceedings of AAAI93 Workshop on AI and Theories of Groups and Organizations, 1993.

[71] GAUTHIER PICARD, 'Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente'. Thèse présentée devant l'Université Paul Sabatier de Toulouse III décembre 2004.

ملخص:

كل يوم الإعلام الآلي يحتل مجالات تطبيقية جديدة أكثر فأكثر تعقيداً، هذه التطبيقات لديها ملتزمات تتغير مع الوقت والفضاء. إنه من الصعب جداً الأخذ بعين الاعتبار هذه التغيرات خلال طور التطوير. إذا من اللازم تزويد هذه النظم بقدرة التكيف الذاتية الشئ الذي سيمكنها من أخذ هيئة لا يمكن تثبيتها مباشرة في بادئ طور التطوير. لقد تم عرض عدة طرق من أجل التصميم الفعلي لتكييف النظم. إلا أن الإشكال يبقى مفتوح ولا يزال الكثير من العمل في هذا المجال. في عملنا هذا نهدف التكيف الذاتي بطريقة متعددة العملاء.

قد أستعمل العديد من الطرق من أجل تطوير النظم المتعددة العملاء. والأكثر نجاعة منها هي التي تعتمد على تنظيم الأدوار. في مذكرتنا هذه نعرض طريقة من أجل تكيف النظم المتعددة العملاء معتمدة على تنظيم الأدوار، طريقة بإمكانها التعامل مع شروط محيط مفتوح، ديناميكي و دائم التطور. ولأجل هذا نأخذ كقاعدة مبدأ تكيف النظم المتعددة العملاء بإعادة تنظيم تكليف الأدوار على العملاء. كوسيلة تقنية سنستعمل النظم المتعددة العملاء المعتمدة على إعادة التنظيم الذاتي للأدوار، الشغل العام للنظام هو عبارة عن دالة تطفو وسيطها شروط المنظمة.

كلمات المفاتيح: التكيف، النظم المتعددة العملاء، منظمة متعددة العملاء.

RESUME

L'informatique envahie chaque jour de nouveaux domaines d'applications de plus en plus complexes. De telles applications ont des besoins qui varient dans le temps et dans l'espace. Tenir compte de cette variabilité au cours du développement est très difficile. Il faut alors doter de tels systèmes de capacités d'auto-adaptation leur permettant d'adopter des aspects ne pouvant être fixés directement dans le développement initial. Plusieurs approches ont été proposées pour mettre en œuvre l'adaptation de systèmes. Cependant, le problème encore ouvert et beaucoup de travail reste à faire dans ce domaine. Dans ce projet, nous visons une approche multi-agent pour l'auto-adaptation.

Plusieurs inspirations ont été utilisées pour le développement de systèmes multi-agent. La plus concrète d'entre elles est celle basée organisation de rôles. Dans notre mémoire on propose une approche pour l'adaptation des systèmes multi-agent basés organisation de rôles, une approche capable de répondre aux contraintes d'un environnement ouvert, dynamique et en constante évolution. Pour cela nous prenons pour base le concept d'adaptation du système multi-agent par une réorganisation d'assignation des rôles aux agents. Comme moyen technique nous utilisons les systèmes multi-agents basés organisation (auto-organisation) des rôles, la fonction globale du système est une fonction émergente guidée par des contraintes d'organisation.

Mots clés : Adaptation, Systèmes multi-agent, Organisation multi-agent.