

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE LARBI BEN M'HIDI – OUM EL BOUAGHI
FACULTE DES SCIENCES EXACTES ET DES SCIENCES DE LA NATURE ET DE LA VIE
Département DE MATHÉMATIQUES ET INFORMATIQUE

N° D'ordre :

Série :

Thèse

Pour l'obtention du diplôme de

DOCTORAT LMD EN INFORMATIQUE

Option: *Ingénierie des Systèmes Distribués*

Thème

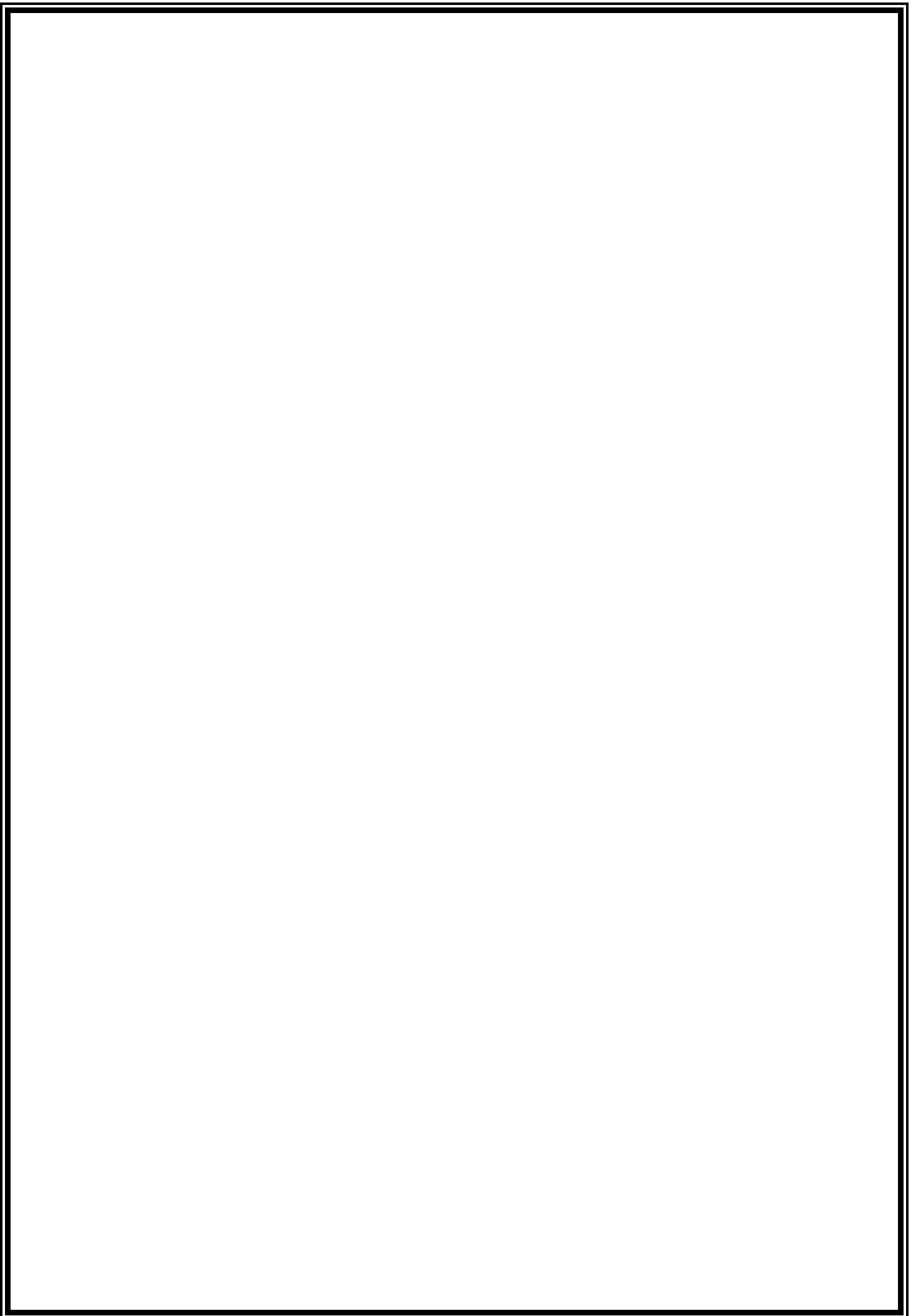
**Une approche basée sur les Cartes
Cognitives Floues pour la prise de décision
collective dans les Systèmes d'Information
Coopératifs**

Présentée par : MAZIZ Asma

Devant le jury composé de :

| | | | |
|-----------------------------|----------------------------|-------------|--|
| Président : | NINI Brahim | Prof | Univ. Oum El Bouaghi |
| Directeur de thèse : | ZAROUR Necereddine | Prof | Univ. Constantine 2- Abdelhamid Mehri |
| Examineurs : | MAAMRI Ramdane | Prof | Univ. Constantine 2- Abdelhamid Mehri |
| | BOUROUIS Abdelhabib | MCA | Univ. Oum El Bouaghi |
| | ZITOUNI Abdelhafid | MCA | Univ. Constantine 2- Abdelhamid Mehri |

2018/2019



Résumé

Actuellement, l'exécution du processus de la prise de décision est l'acte le plus important dans un système d'information; il devient plus difficile et plus complexe dans un contexte coopératif. Notre thèse a pour objectif de proposer la conception et le développement d'une approche basée sur l'outil d'aide à la décision nommé cartes cognitives floues (CCF) et le paradigme des systèmes multi-agents holonique dont le but d'améliorer le processus de prise de décision et éliminer les incohérences issues des décisions partielles dans les systèmes d'information coopératifs (SIC). De plus, le concept d'ontologie est utilisé pour enrichir sémantiquement notre architecture proposée. Nous modélisons chaque sous-SIC par un groupe d'agents holoniques où chacun utilisait une CCF afin de mieux modéliser et analyser les systèmes dynamiques complexes. Ce groupe va essayer de prendre une décision collective pour résoudre n'importe quel problème distribué. Le domaine de la sécurité routière a été choisi pour mettre notre approche en application, dont le but de voir comment éduquer les gens afin de réduire le nombre d'accidents mortels. Enfin, nous validons notre proposition à travers des expériences pour montrer comment elle améliore le processus de prise de décision dans un contexte coopératif.

Mots clés: aide à la décision, systèmes d'information coopératifs (SIC), cartes cognitives floues (CCF), systèmes multi-agents (SMA), systèmes holoniques, ontologie, sécurité routière.

Abstract

Currently, the execution of the decision-making process is the most important act in an information system; it becomes more difficult and more complex in a cooperative context. Our thesis aims to propose the design and development of an approach based fuzzy cognitive maps (FCM) tool and holonic multi-agent paradigm that enhance the decision making process in cooperative information system (CIS). Furthermore, the concept of ontology is used for semantically enrich our architecture. We modeled each sub-CIS by a holonic agents where everyone used a FCM for a more precise analysis of complex dynamic system decisions. This group will try to make a collective decision to solve any given distributed problem. The field of road safety has been chosen to put our approach into practice, whose purpose is to see how to educate people to reduce the fatal accidents number. Finally, we validated our proposition through experiments to show how it improves the decision making process in a cooperative context.

Keywords: Decision Support, Cooperative Information Systems (CIS), Fuzzy Cognitive Maps (FCM), Multi-agent Systems (MAS), Holonic systems, Ontology, Road safety.

الملخص

حالياً، يعتبر تنفيذ عملية صنع القرار أهم عمل في نظام المعلومات؛ فإنه يصبح أكثر صعوبة وأكثر تعقيداً في سياق تعاوني. تهدف أطروحتنا إلى اقتراح تصميم وتطوير نهج يقوم على أداة مساعدة القرار تسمى الخرائط المعرفية الغامضة (CCF) ونموذج نظم متعددة العوامل (SMA) التي تهدف إلى تحسين عملية أخذ القرار والقضاء على أوجه عدم الاتساق الناشئة عن القرارات الجزئية في نظم المعلومات التعاونية (SIC). وبالإضافة إلى ذلك، يتم استخدام مفهوم أونتولوجيا لإثراء بشكل منطقي الهندسة المقترحة. نقوم بتمثيل كل مجموعة فرعية من (SIC) بمجموعة من العوامل الهلونية حيث يستخدم كل منها نموذج CCF لتحليل النظم الديناميكية المعقدة بشكل أفضل. ستحاول هذه المجموعة اتخاذ قرار جماعي لحل أي مشكلة موزعة. تم اختيار مجال السلامة على الطرق لوضع نهجنا موضع التنفيذ، الغرض منه هو معرفة كيفية تثقيف الناس للتقليل من عدد الحوادث المميتة. وأخيراً، قمنا بالتحقق من صحة اقتراحنا من خلال التجارب لإظهار كيف يحسن عملية صنع القرار في سياق تعاوني.

كلمات البحث : دعم القرار، نظم المعلومات التعاونية، الخرائط المعرفية الغامضة، أنظمة متعددة الوكلاء، نظم هولونيك، علم الأنطولوجيا، السلامة على الطرق.

Remerciements

En tout premier lieu, je remercie mon dieu, tout puissant, qui m'a éclairé le bon chemin et qui m'a aidé à réaliser ce travail dans les meilleures conditions.

J'aimerais remercier en particulier mon encadreur Mr ZAROUR Nacereddine Professeur à l'université Constantine 2 – Abdelhamid Mehri de Constantine pour avoir dirigé ce travail, pour son aide, ses conseils, ses encouragements, sa grande disponibilité et son ouverture d'esprit qui m'ont aidé à mener à bien ce travail.

Je tiens à remercier vivement Mr NINI Brahin Professeur à l'université Larbi Ben M'hidi d'Oum El Bouaghi qui me fait l'honneur de présider le jury de soutenance.

Mes vifs remerciements vont également vers tous les membres du jury :

- Mr MAAMRI Ramdane Professeur à l'université Constantine 2 – Abdelhamid Mehri de Constantine,
- Mr BOUROUIS Abdelhabib Maître de Conférences A à l'université Larbi Ben M'hidi d'Oum El Bouaghi,
- Mr ZITOUNI Abdelhafid Maître de Conférences A à l'université Constantine 2 – Abdelhamid Mehri de Constantine,

qui m'ont honoré par leurs acceptations de bien vouloir consacrer de leur temps à évaluer ce modeste travail et à participer à ce jury.

Je ne voudrais pas oublier tous ceux et celles qui au long de ce travail m'ont soutenu moralement, je les remercie de leurs encouragements.

Table des matières

| | |
|---|-----------|
| Introduction générale..... | 1 |
| Chapitre I : Généralités sur les systèmes d'information coopératifs..... | 5 |
| I.1. Introduction..... | 6 |
| I.2. Système d'information..... | 6 |
| I.2.1. Définition..... | 6 |
| I.3. Systèmes d'information coopératifs (SICs)..... | 7 |
| I.3.1. Définition..... | 7 |
| I.3.2. Propriétés et caractéristiques d'un SIC..... | 8 |
| I.3.2.1. Autonomie..... | 9 |
| I.3.2.2. Distribution..... | 9 |
| I.3.2.3. Hétérogénéité..... | 9 |
| I.3.2.4. Evolutivité..... | 10 |
| I.3.2.5. Scalabilité..... | 10 |
| I.3.2.6. Transparence..... | 11 |
| I.3.3. Taxonomie des SICs..... | 11 |
| I.3.3.1. L'approche fédérée..... | 11 |
| I.3.3.2. Les approches à base de médiation..... | 13 |
| I.3.3.3. Comparaison entre les approches fédérées et les approches à base de médiation..... | 17 |
| I.4. Conclusion..... | 21 |
| Chapitre II : Les agents et les systèmes multi-agents..... | 22 |
| II.1. Introduction..... | 23 |
| II.2. Agent..... | 23 |
| II.2.1. Définitions d'un agent..... | 23 |
| II.2.2. Caractéristiques d'un agent..... | 24 |
| II.2.3. Architecture interne d'un agent..... | 25 |
| II.2.3.1. Agents réactifs..... | 25 |
| II.2.3.2. Agents cognitifs..... | 26 |

Table des matières

| | |
|---|-----------|
| II.3. Système multi-agents (SMA)..... | 27 |
| II. 3.1. Définitions d'un SMA..... | 27 |
| II.3.2. Caractéristiques d'un SMA..... | 28 |
| II.3.3. Types des SMA..... | 29 |
| II.3.4. Méthodes de conception de SMA..... | 31 |
| II.3.4.1. MaSE..... | 32 |
| II.3.4.2. AAIL..... | 32 |
| II.3.4.3. DESIRE..... | 32 |
| II.3.4.4. Aalaadin..... | 32 |
| II.3.4.5. MAS-CommonKADS..... | 32 |
| II.3.4.6. GAIA..... | 32 |
| II.3.4.7. Voyelles..... | 33 |
| II.3.4.8. ADELFE..... | 33 |
| II.3.4.9. PASSI..... | 33 |
| II.3.5. Plates-formes de développement des SMA..... | 33 |
| II.3.5.1. Madkit..... | 34 |
| II.3.5.2. RepastHPC | 34 |
| II.3.5.3. D-Mason..... | 34 |
| II.3.5.4. FLAME..... | 34 |
| II.3.5.5. JADE..... | 35 |
| II.3.5.6. Magique..... | 35 |
| II.3.5.7. AgentBuilder..... | 35 |
| II.4. Conclusion..... | 35 |
| Chapitre III : Théorie de la décision et les cartes cognitives floues..... | 37 |
| III.1. Introduction..... | 38 |
| III.2. Définition générale de la théorie de décision..... | 38 |
| III.3. Notions de base de la théorie de décision..... | 38 |
| III.3.1. Décision..... | 38 |
| III.3.2. Décideur..... | 39 |
| III. 3.3. La prise de décision..... | 39 |
| III.3.3.1. Définition..... | 39 |
| III.3.3.2. Processus de la prise de décision..... | 39 |
| III.3.3.3. Les outils d'aide à la prise de décision..... | 40 |
| III.3.3.3.1. Diagrammes d'influence..... | 41 |

Table des matières

| | |
|---|-----------|
| III.3.3.3.2. Arbres de décision..... | 42 |
| III.3.3.3.3. Cartes cognitives floues..... | 43 |
| III.4. Travaux similaires..... | 55 |
| III.4.1. SIC et SMA..... | 55 |
| III.4.2. CCF et SMA..... | 56 |
| III.5. Conclusion..... | 57 |
| Chapitre IV : DCSic-CFSH: l'aide à la Décision Collective dans les SIC basée sur les Cartes Cognitives Floues et les Systèmes Multi-Agents Holoniques..... | 58 |
| IV.1. Introduction..... | 59 |
| IV.2. Intégration des CCF dans les SIC..... | 59 |
| IV.3. Modélisation d'un SIC à base d'agents holoniques..... | 60 |
| IV.4. Architecture globale du système..... | 61 |
| IV.4.1. Types d'agents | 63 |
| IV.4.2. Opérations de base utilisées..... | 66 |
| IV.4.3. Ontologie de domaine | 68 |
| IV.5. Description détaillée du système DCSic-CFSH..... | 69 |
| IV.5.1. Position du problème..... | 69 |
| IV.5.2. Construction de la décision collective (CCF collective) | 69 |
| IV.5.2.1. La sélection de la source d'informations..... | 69 |
| IV.5.2.2. L'acquisition de la carte cognitive..... | 70 |
| IV.5.2.3. La conversion de la carte en forme de matrice..... | 70 |
| IV.5.2.4. Coopération entre les agents SI dans le troisième niveau..... | 70 |
| a. Coefficient d'expertise | 70 |
| b. Exécution de l'union des CCFs entre les agents SI..... | 71 |
| IV.5.2.5. Coopération entre les agents SIC dans le deuxième niveau..... | 71 |
| a. Distribution du coefficient d'émergence sur les agents représentants les sous SICs..... | 71 |
| b. Exécution de l'intersection des CCFs entre les agents SIC..... | 72 |
| IV.5.2.6. Exécution du processus d'analyse et d'inférence sur la CCF collective du système | 76 |
| IV.5.2.7. Construction de la base de connaissance globale (les solutions possibles au problème global posé) | 78 |
| IV.6. Conclusion..... | 78 |

Table des matières

| | |
|---|-----------|
| Chapitre V : Application | 79 |
| V.1. Introduction..... | 80 |
| V.2. Exemple : Sécurité Routière..... | 80 |
| V.3. Environnement d'application..... | 80 |
| V.4. Application..... | 82 |
| 4.1. Position du problème..... | 82 |
| 4.2. Construction de la décision collective (CCF collective) | 82 |
| 4.2.a. Coopération entre les agentsSIs dans le troisième niveau (exécution de l'union des CCFs entre les agents SIs) | 83 |
| 4.2.b. Coopération entre les agentsSICs dans le deuxième niveau (exécution de l'intersection des CCFs entre les agents SICs) | 83 |
| 4.2.c. Exécution du processus d'analyse et l'inférence sur la CCF collective | 85 |
| 4.2.d. La construction de la base de connaissance globale (les solutions possibles au problème posé) | 85 |
| V.5. Expérimentation..... | 91 |
| V.6. Validation..... | 93 |
| Conclusion générale | 95 |
| Bibliographie | 98 |

Liste des figures

| | |
|--|----|
| Figure.I.1. Les cinq niveaux de schémas de l'approche fédérée..... | 12 |
| Figure.I.2. Architecture basée sur un système de médiation..... | 14 |
| Figure II.1. Agent réactif..... | 26 |
| Figure II.2. Agent cognitif..... | 27 |
| Figure II.3. Un exemple de SMAH..... | 31 |
| Figure III.1. Modélisation du processus de décision de Simon [Simon, H.A., 1977].... | 40 |
| Figure III.2. Diagramme d'influence..... | 41 |
| Figure III.3 Exemple d'arbre de décision..... | 42 |
| Figure III.4. Exemple de CCF..... | 44 |
| Figure III.5. Différentes fonctions de normalisation σ d'une CCF..... | 45 |
| Figure III.6. Exemple de deux cartes cognitives de même auteur..... | 48 |
| Figure III.7. Intersection de deux cartes : $\text{Carte1} \cap \text{Carte2} = \text{Carte-Res}(1, 2)$ | 49 |
| Figure III.8. Exemple de l'effet direct..... | 50 |
| Figure III.9. Exemple d'addition..... | 51 |
| Figure III.10. Exemple de carte cognitive [Parenthoen M. et al., 2001] | 54 |
| Figure IV.1. : Architecture globale de DCSic-CFSH..... | 62 |
| Figure IV.2. : L'architecture SMAH modélisant le système initial décrit dans la figure IV.1..... | 62 |
| Figure IV.3. Architecture de l'agentSI..... | 63 |
| Figure IV.4. Architecture de l'agentSIC..... | 64 |
| Figure IV.5. Architecture de l'agentSICG..... | 65 |
| Figure IV.6. Architecture de l'agent analyseur..... | 65 |
| Figure IV.7. Communication entre les différents agents de DCSic-CFSH..... | 66 |
| Figure IV.8. Exemple de deux CCFs de deux agents..... | 67 |
| Figure IV.9. Union de deux cartes: $\text{CCF1} \cup \text{CCF2} = \text{CCF12}$ | 67 |
| Figure IV.10. Exécution de l'opération d'union entre les agents SI (décideurs)..... | 71 |
| Figure IV.11. Architecture d'agent global représentant le SIC (AgentSICG)..... | 72 |

Liste des figures

| | |
|--|----|
| Figure IV.12. Exécution de l'opération d'intersection entre les agents SIC..... | 73 |
| Figure IV.13. Chronogramme de démarrage du processus de la prise de décision..... | 74 |
| Figure IV.14. Chronogramme d'envoi l'agentSIC1 le problème aux agentSIs..... | 74 |
| Figure IV.15. Chronogramme du retour des réponses au niveau du sous SIC1..... | 75 |
| Figure IV.16. Chronogramme du retour des réponses au niveau de l'agentSIC..... | 75 |
| Figure IV.17. Chronogramme du protocole d'interaction..... | 76 |
| Figure IV.18. Communication entre l'agent global (SIC) et l'agent analyseur..... | 76 |
| Figure V.1. Console de la plateforme AgentBuilder..... | 81 |
| Figure V.2. Architecture proposée avec l'exemple de la sécurité routière..... | 82 |
| Figure V.3. Communication entre les différents agents sous AgentBuilder [Maziz, A., Zarour, N., in press]..... | 84 |
| Figure V.4. Carte cognitive floue collective [Maziz, A., Zarour, N., in press..... | 84 |
| Figure V.5. Résultat d'exécution du processus d'analyse et d'inférence [Maziz, A., Zarour, N., in press]..... | 85 |
| Figure V.6. Exemple d'une CCF collective..... | 86 |
| Figure V.7. Exemple de la base de connaissance sous AgentBuilder..... | 88 |
| Figure V.8. Ontologie de la sécurité routière [Maziz, A., Zarour, N., in press]..... | 89 |
| Figure V.9. Modélisation AUML des interactions entre les différents agents..... | 90 |
| Figure V.10. Protocole de la sécurité routière [Maziz, A., Zarour, N., in press]..... | 91 |
| Figure V.11. Résultat de l'exécution sous forme d'histogramme..... | 92 |

Liste des tableaux

| | |
|---|-----------|
| Tableau III.1. Valeurs possibles d'une relation causale..... | 47 |
| Tableau III.2. Règles d'opération d'intersection..... | 48 |
| Tableau III.3. Règles d'opération d'union..... | 49 |
| Tableau III.4. Règles de la multiplication..... | 50 |
| Tableau III.5. Règles de l'addition..... | 51 |
| Tableau V.1. Comparaison de DCSic-CFSH avec deux autres approches..... | 93 |

Introduction générale

La nécessité de prendre de nombreuses décisions et de résoudre des problèmes plus ou moins complexes est un besoin crucial créé chaque jour. Pour une entreprise, la décision est importante car elle est la condition de sa réussite et de son développement.

Les systèmes d'aide à la décision (SAD) sont des systèmes informatiques spécialisés pour prendre en charge certaines, plusieurs ou toutes les phases d'un processus décisionnel. Le SAD peut être utilisé en utilisant des mécanismes traditionnels et intelligents [Mora et al., 2014]. La prise de décision est le processus qui permet d'obtenir le meilleur choix en établissant des objectifs et en recueillant de l'information. L'amélioration et la bonne exécution du processus de prise de décision est l'acte le plus important car un système d'information (SI) peut l'appliquer dans le temps et l'espace. En fait, cela devient plus complexe et difficile dans un contexte coopératif.

Actuellement, la coopération des SIs est devenue un besoin important en particulier dans le contexte du développement rapide des technologies de la communication et de la croissance des échanges d'informations. Cette coopération est un mécanisme dans lequel les acteurs, avec des intérêts différents, travaillent ensemble pour décider de trouver des solutions à un problème commun.

Depuis quelques dernières années, le phénomène concernant la résolution distribuée des problèmes a vu un intérêt accru en particulier dans le domaine de la prise de décision dans les organisations [Adla, A., 2010 ; Mathieu, L., 2017 ; Uglev, V., 2014]. Par exemple, la prise de décision médicale peut être considérée comme un processus combinant à la fois la cognition analytique et l'intuition. Il s'agit de raisonner à l'intérieur de modèles causaux complexes de concepts multiples, généralement décrits par des informations incertaines, imprécises et / ou incomplètes [Iakovidis, D. K., 2011].

Problématique

Le SI actuel (d'une entreprise) a été caractérisé en tant que système matériel / logiciel / personnel par la mondialisation du commerce et la diversification des acteurs. Pour persister dans un tel

environnement, le SI doit supporter un ensemble de partenaires forts, et donc intégrer une structure externe. Tous les SIs prennent généralement la forme de système d'information coopératif (SIC). Ce dernier est un domaine de recherche relativement jeune qui cherche à maximiser la facilité de construction, la souplesse et la robustesse grâce à des combinaisons de techniques issues de l'informatique distribuée [Kessi, K. et al., 2015].

Les recherches en intelligence artificielle démontrent un constant intérêt pour étudier et appliquer les techniques de prise de décisions. Par exemple dans [Mingchua, Z. et al., 2012], il est proposé d'étudier un mécanisme de décision coopérative pour les multi-nœuds autonomes (NAs) et aussi de présenter un processus décisionnel coopératif correspondant dans le réseau cognitif. Aussi dans [Ughev, V., 2014], l'auteur a suggéré la mise en œuvre de méthodes de prise de décision dans un système éducatif automatisé intelligent axé sur l'individualisation complète de l'apprentissage.

Dans les SICs, des nombreux projets sont à la charge pour résoudre les problèmes relatifs à l'hétérogénéité des données, mais il y a peu de travaux qui traitent les problèmes de l'hétérogénéité des décisions partielles des SIs composants afin d'obtenir une décision globale, dont le but de résoudre un problème posé. C'est l'objectif principal de notre travail.

Donc, nous sommes intéressés à la prise de décision dans les SICs. Dans ce domaine nous cherchons également à améliorer la capacité des SIs afin de prendre les meilleures décisions favorisant le fonctionnement du SIC.

Les SIs doivent agir de manière coopérative dans un SIC, leurs décisions sont prises avec comme objectif l'amélioration du rendement du SIC en tant que tel. Généralement, lorsque nous devons prendre une décision, nous le faisons plutôt d'une manière qualitative en analysant les conséquences qui peuvent en découler. C'est pourquoi il nous a paru utile d'intégrer le raisonnement causal d'une manière qualitative dans les SICs.

Donc, dans notre contexte, nous nous sommes intéressés aux SICs où les problèmes de prise de décision augmentent. La principale difficulté réside dans le fait que le SIC global ne contrôle pas toutes les spécificités des différents sous- SIC. Cela implique la nécessité de trouver des moyens de coordination qui pourraient limiter les incohérences découlant de la prise de décision.

Objectifs

Le but principal de notre travail est d'étudier les cartes cognitives de type flou pour les utiliser comme un outil d'aide à la décision collective dans les systèmes d'information coopératifs. Dans ce contexte les principaux objectifs de notre travail sont :

- Présenter les généralités de base sur les SICs, les différentes approches pour la prise de décisions en particulier les cartes cognitives floues et les systèmes multi agents;

- Proposer une architecture permettant l'aide à la décision collective dans les SICs en se basant sur les cartes cognitives floues comme module de prise de décisions et les SMA comme module de modélisation et d'implémentation;
- Illustrer la faisabilité de notre proposition par l'application d'une étude de cas.

Contribution

Pour améliorer le processus décisionnel, éliminer les incohérences des décisions partielles dans les SICs et trouver des solutions plus précises et réalistes, l'approche proposée est basée sur les cartes cognitives floues (CCFs). L'efficacité de tels systèmes dépend de l'univers dans lequel évolue le décideur humain ou virtuel. En général, tout décideur utilise un système d'aide à la décision pour faciliter le travail d'analyse du problème. Le rôle principal de ces systèmes est de compléter, améliorer et aider le décideur à réduire le temps d'exécution et le taux d'erreur.

Dans notre approche, nous proposons d'associer à chaque SI (que ce soit un individu ou une entreprise) une CCF, qui sera considérée comme un système d'aide à la communication, à l'analyse des idées, à la prise de décision pour résoudre un problème distribué donné, aussi contribue également à améliorer la base de connaissances des SIs et de réduire les problèmes générés par leurs vues partielles. En outre, nous utilisons le paradigme nommé système multi agents holonique (SMAH) [Beheshti, R. et al., 2016], qui permet de modéliser et de mettre en œuvre les SICs en tant que systèmes distribués, dans lesquels différents acteurs agissent indépendamment pour atteindre un objectif spécifique, et interagir afin d'atteindre un objectif global. De plus, une ontologie de domaine est associée à notre architecture proposée pour définir le domaine des connaissances nécessaires par le décideur agent dans son environnement. Plus spécifiquement, l'ontologie de domaine est utilisée pour enrichir notre architecture et minimiser les concepts générés sur le problème distribué dans la CCF collective.

Donc, la combinaison de SIC, SMAH, CCF et le concept d'ontologie nous fournit les éléments nécessaires pour la modélisation pertinente des comportements et des interactions dans un contexte coopératif. En d'autres termes, nous avons modélisé un SIC par SMAH, dont chaque agent holonique avec l'ontologie de domaine a une CCF afin de prendre les bonnes décisions et donc résoudre le problème global posé.

Organisation de la thèse

Compte tenu de nos objectifs, cette thèse est organisée de la manière suivante :

Nous exposons dans un premier temps dans le **chapitre I** quelques généralités de base sur les systèmes d'information coopératifs ; la première partie de ce chapitre définit un système

d'information, puis dans la deuxième partie, les concepts de base relatifs à ce contexte sont présentés.

Le **chapitre II** est consacré à l'étude des systèmes multi agents dont lequel nous l'abordons d'abord sous l'aspect « agent » en présentant ses définitions, ses caractéristiques et aussi ses architectures. Puis, nous passons à la dimension collective du système en présentant les définitions, les caractéristiques, les différents types d'un SMA. Nous terminons le chapitre par quelques méthodes de conception ainsi que les plates-formes de développement des SMA les plus utilisées.

Puis dans le **chapitre III**, nous nous intéressons à la théorie de la décision. Cette dernière est un domaine très large, nous nous limiterons uniquement à quelques notions de bases qui sont : décision, décideur, prise de décision et outils d'aide à la décision les plus connus. Nous décrivons en profondeur un des outils d'aide à la décision du type qualitatif : cartes cognitives floues, où nous parlerons de leur historique, leurs définitions, l'algèbre causale sous-jacente qui permet de les manipuler, le processus de leur construction, et finalement dans la dernière section nous présentons quelques travaux de la littérature ayant traité la problématique considérée.

Le **chapitre IV** va présenter en détail notre contribution qui consiste à construire une architecture basée sur l'utilisation des cartes cognitives floues dans les SICs pour améliorer le processus de la prise de décision collective et aussi basée sur d'autres paradigmes comme les SMAH et le concept d'ontologie.

Dans le **chapitre V**, nous exposons la partie conception et implémentation de notre travail, où nous commençons par la description de notre cas d'étude sur le domaine de la sécurité routière. Par la suite nous citons les outils utilisés, l'application et quelques expérimentations avec une comparaison avec d'autres travaux pour illustrer la faisabilité et l'applicabilité de notre proposition.

La conclusion de cette thèse nous permet de présenter les avantages d'implémenter notre approche et aussi de dresser les principales perspectives des travaux auxquels nous pensons nous investir dans le futur.

Chapitre I :
*Généralités sur les systèmes
d'information coopératifs*

1.1. Introduction

Actuellement, la coopération des systèmes d'information est devenue un besoin important surtout dans le cadre du développement rapide des technologies de communication et la croissance d'échange d'information. Cette coopération est un mécanisme dont lequel les acteurs, ayant des intérêts variés, travaillent ensemble pour trouver des solutions à un problème commun. Elle peut être considérée dans plusieurs niveaux. La consolidation des sources d'information préexistantes et la fusion des différents points de vues dans les systèmes d'information est aujourd'hui une réalité, offrant un accès unifié à des informations distribuées et hétérogènes. Dans ce chapitre, nous nous intéressons aux systèmes d'information coopératifs (SICs); on va tout d'abord définir un système d'information, puis on va présenter les concepts de base relatives à un SIC qui est notre contexte de travail.

1.2. Système d'information

1.2.1. Définition

La complexité des problèmes d'économie et les développements obtenus dans les technologies de l'information sont la cause originale de l'émergence du concept du système d'information (SI). Ce dernier est considéré comme un élément important dans les organisations, il est utilisé pour collecter, stocker, manipuler et présenter l'information nécessaire à la coordination, au contrôle et à la prise de décision au sein d'une organisation [Hicks, J., 1993]. Il existe plusieurs définitions de ce concept, par exemple:

Dans [Rolland, C. et al, 1988] : « Un SI est un artefact qui supporte un réseau de flux d'informations nécessaires pour organiser, mettre en œuvre, gérer et maintenir les activités d'une organisation. C'est un instrument de communication qui sert aux échanges informationnels entre les partenaires de l'organisation et qui accroît leur efficacité».

Le SI est défini dans [Tardieu, H. et al, 1994] comme « l'ensemble des ressources matérielles, humaines et informatiques contribuant à la collecte, la mémorisation, la recherche, la communication et l'utilisation des données nécessaires et suffisantes pour un pilotage des opérations au sein du système de l'entreprise ».

Dans [Zacklad, M., 1999], on donne deux définitions la première est que « Le SI est l'ensemble des instruments, individuels ou collectifs, qui participent au processus de gestion des informations au sein de l'organisation ». Et la deuxième : « Organe situé entre un système opérant où se déroule les processus opérationnels et un système de pilotage chargé de prendre les décisions. ».

Les SIs aident les acteurs à analyser les problèmes, visualiser les sujets complexes et créer de nouveaux produits. Le rôle des SIs est devenu plus complexe et plus large au cours des décennies, allant du support d'une activité spécifique et d'un ensemble limité d'utilisateurs au support des processus métiers et des collaborations entre acteurs dans un environnement décentralisé et

distribué [Verharen, E., 1997]. Un SI n'est pas unique tel que préconisé par les méthodes classiques [Tardieu, H. et al, 1994], mais plusieurs sous-ensembles du SI qui ont des objectifs différents et sont automatisés à l'aide de méthodes de conception différentes [Zarour, k., 2007]. Cette nouvelle structuration d'un SI nous amène à considérer que la connaissance nécessaire à la prise de décision dans une entreprise est constituée non seulement de la connaissance de ses membres mais aussi de la connaissance émergente de leurs interactions les uns avec les autres [Baghdadi, Y., 1997]. Dès la fin des années 80, l'objectif des premières propositions relatives à la coopération de systèmes d'information qui sont ouverts, distribués, hétérogènes et de plus en plus complexe. Son objectif consiste à traiter les différentes techniques structurelles et syntaxiques entre les SIs, aussi de maîtriser les différentes spécificités relatives à l'hétérogénéité soit de sources d'information par exemple, ou bien des décisions partielles décrivant les points de vues des SIs pour résoudre un problème donné.

1.3. Systèmes d'information coopératifs

1.3.1. Définition

Les systèmes d'information coopératifs (SICs) sont un domaine de recherche issu de la synthèse des bases de données et de l'intelligence artificielle. Ce genre de systèmes occupe de nos jours une place centrale dans les organisations. Plusieurs définitions de ce paradigme sont proposées, par exemple :

Les définitions de SIC des années 1980 se sont concentrées sur la coopération avec un utilisateur simple, par exemple, par des interfaces de langage naturel ou l'interrogation sémantique traitant [Bolc, L., et al, 1986]. Et au début des années 90, les SICs firent leur apparition grâce à [Boulanger, D., 1991; Papazoglou, M.P. et al, 1992]. Cet axe de recherche est issu principalement de l'Intelligence Artificielle Distribuée (IAD) et les Bases de Données.

Dans l'approche des bases de données distribuées, un SIC est un système intégrant des sources d'informations distribuées, des bases de données et/ou systèmes à bases de connaissances, pouvant utiliser des représentations de connaissances et de données hétérogènes [Brodie, M., Ceri, S., 1992].

La notion de SIC a été aussi introduite au cours de la conférence CoopIS' 1994 [De Michelis, G. et al, 1994] qui donne la définition comme suit : « Les SICs englobent un nombre important de SIs distribués au travers d'un grand nombre de réseaux d'ordinateurs et de communication complexes. Ces systèmes supportent toutes les catégories d'applications, des plus simples au plus complexes. De tels SIs gèrent et accèdent à une grande quantité d'information et de services automatisés. Ils supportent également le travail humain individuel ou de groupe. L'exécution des tâches s'effectue de manière concurrente sur des réseaux à l'aide de logiciels spécifiques. Les informations et les services sont disponibles sous plusieurs formes grâce à des référentiels. La communication entre les

systèmes compétents peut s'effectuer de façon centralisée ou distribuée grâce à des protocoles de communication».

D'après [De Michelis, G. et al, 1996] « Un SI est dit coopératif s'il partage des buts communs avec d'autres SIs, agents humains ou l'organisation même et contribue positivement à l'accomplissement de ces buts ».

Dans le domaine du CSCW (Computer Supported Cooperative Work), Boulanger et Dubois [Boulanger, D., Dubois, G., 1997] proposent une définition plus fédératrice dans laquelle un SIC est perçu comme « un ensemble de composants plus ou moins autonomes, souvent préexistants qui travaillent de manière synergique en échangeant information, expertise et en coordonnant leur activités ».

Une autre définition proposée dans [De Michelis, G. et al, 1997] qui considère le concept de SIC comme un cadre architectural qui maintient l'uniformité parmi une variété de systèmes informatisés, de groupes d'utilisateurs, et d'objectifs organisationnels pendant que le tout évolue dans le temps.

Pour [Michael, H., 1997] on catégorise le développement des SICs sur quatre générations, système d'information centralisé, architecture client/serveur, architecture Peer to Peer et finalement les systèmes coopératifs.

Selon la communauté de l'Intelligence Artificielle Distribuée, un SIC est défini comme « un ensemble d'agents (programmes) qui partagent continuellement des objectifs avec d'autres systèmes d'information, des agents humains aussi bien qu'avec l'organisation au sein de leur environnement opérationnel » [Herin, D. et al, 2001].

Un SIC est un système complexe, lieu de multiples interactions, ouverts sur de nombreux systèmes interconnectés, distribués, hétérogènes et autonomes, qui interagit de façon permanente avec son environnement. L'émergence de connaissances nouvelles, est le résultat de la coopération de ses composants [Zarour, K., 2007]. Et finalement dans [Kessi, K. et al., 2015], Un SIC est un système complexe; il implique la coopération de nombreuses parties prenantes dans un but commun et chacune avec son propre point de vue. Tout de long de notre travail nous nous basons sur ces deux dernières définitions.

Parmi les objectifs principaux d'un SIC c'est : résoudre les différents types d'hétérogénéité ou les conflits inhérents à la coopération de systèmes d'information [Frendi, M., 2013], collecter les différentes décisions partielles pour résoudre un problème donné, et aussi traiter les requêtes soumises au SIC en respectant ses caractéristiques ou propriétés définies dans la section suivante.

3.2. Propriétés et caractéristiques d'un SIC

D'après [Sciore, E. et al, 1994 ; Liu, L., Lu, C., 1995; Zarour, N., 2004; Zarour, K., 2007 ; Frendi, M., 2013], un SIC possède plusieurs propriétés fondamentales, on peut citer les plus distinctives

par exemple : l'autonomie, la distribution, l'hétérogénéité, l'évolutivité, la scalabilité et la transparence. Dans ce qui suit on va décrire chacune comme suit :

1.3.2.1. Autonomie

Un système est autonome s'il a la capacité de travailler, de changer, de contrôler, de communiquer et aussi a la capacité d'intégrer ou d'abandonner sans altération du fonctionnement global du SIC. Dans notre travail, nous nous intéressons à traiter l'autonomie des décisions partielles des différents systèmes composants d'un SIC, dont le but de créer la cohérence de ces décisions pour arriver à une décision globale pour résoudre n'importe quel problème distribué. Selon [Frendi, M., 2013 ; Scheuermann, P. et al, 1990], il existe plusieurs types d'autonomie, on peut citer :

- a. L'autonomie de conception : signifie que chaque SI composant dans un SIC est libre de choisir sa propre conception. Ce principe interdit tout changement dans leur conception et leur implémentation; c'est des boîtes noires. Il permet de garantir la continuité des applications héritées.
- b. L'autonomie de communication : c'est la capacité d'un SI composant de préciser la manière de communiquer ou non avec les autres systèmes, et aussi la capacité de choisir les protocoles de communication.
- c. L'autonomie d'exécution : assure qu'un système a la liberté de traiter les requêtes externes provenant par d'autres systèmes composants, aussi a la capacité de choisir l'ordre de ces requêtes ou refuser certaines d'entre elles.
- d. L'autonomie d'association : permet à un système composant de pouvoir partager ou non ses fonctionnalités ou ressources avec d'autres systèmes composants.

1.3.2.2. Distribution

La distribution physique des données indique que les informations sont réparties entre des sites géographiquement éloignés mais peuvent, aussi, être localisées sur un même site [Frendi, M., 2013]. Le problème de distribution des données est résolu par des techniques nombreuses et disparates telles Hyper Text Transfer Protocol (HTTP), Common Object Request Broker Architecture (CORBA)/Internet Inter-ORB Protocol (IIOP), ces techniques permettent de prendre en charge cette distribution. Dans notre travail, cette caractéristique est devenue un besoin crucial, qui est aussi résolu dans notre approche. Cette dernière permet l'accès unifié à des informations distribuées et hétérogènes.

1.3.2.3. Hétérogénéité

L'hétérogénéité est le résultat d'évolution autonome des systèmes composants qui sont différents soit au niveau de leur structure, leur type et aussi au niveau du mode de représentation des

informations. Notre travail est concentré sur l'hétérogénéité surtout des décisions partielles des différents SIs composants du SIC globale. Notre but est donc comment maîtriser ces hétérogénéités des points de vue afin d'obtenir une décision globale pour résoudre un problème posé.

Selon [Goh, C. et al, 1994 ; Jouanot, F., 2001 ; Kim, W., Seo, J., 1991 ; Sheth, A., Kashyap, V., 1992; Sheth, A., 1999] plusieurs classifications ont été proposées. Donc, l'hétérogénéité a été classée en trois niveaux, à savoir :

- L'hétérogénéité technique se réfère à l'hétérogénéité des plates-formes, i.e. des systèmes d'exploitation et des systèmes matériels, sur lesquelles sont situés les systèmes composants.
- L'hétérogénéité structurelle traduit entre les systèmes locaux une différence :
 - ✓ de modèle de données (e.g., réseau, relationnel, ou orienté objet),
 - ✓ de langage de requêtes (e.g., Langage de manipulation des données , DAML, SQL ou OQL),
 - ✓ de représentation des informations entre composants coopérants, ce qui engendrent plusieurs types de conflits lors de cette coopération.
- Hétérogénéité sémantique : représente les différences dans l'interprétation ou la signification d'une même donnée. Aussi, cette hétérogénéité sémantique symbolise les conflits de domaine de définition. Elle englobe les conflits soit de noms, de valeurs, de précision de données, ou les conflits d'échelle [Freundi, M., 2013].

1.3.2.4. Evolutivité

Cette caractéristique désigne la capacité de prendre en charge les modifications qui peuvent apparaître au sein d'une architecture coopérative. Ces modifications interviennent à deux niveaux, soit au niveau du SIC (ajout / retrait d'un composant participant), ou soit au niveau des systèmes composants (modification d'un schéma, de l'implémentation ou de la sémantique d'un composant,...). Notre système proposé va prendre en compte cette caractéristique, puisque il existe la possibilité de changer le point de vue d'un SI composant pour s'abonner ou non au SIC mondial pour résoudre un problème distribué posé.

1.3.2.5. Scalabilité

Ce concept signifie qu'un SIC a la faculté de gérer un nombre important de SIs composants. La scalabilité permet d'assurer une performance optimale dans l'exécution d'une requête lorsque le nombre de sources d'information est élevé. Cette caractéristique aussi est assurée dans notre travail, puisque notre système ne pose pas des contraintes sur le nombre des SIs composants ; Quel que soit le nombre de SIs composants est important, notre système proposé va réduire ce nombre par des opérations, on va les voir en détail dans le chapitre IV.

1.3.2.6. Transparence

Un SI est transparent désigne un système intégré ; autrement dit, il doit être vu un système centralisé, fonctionnel, cohérent et aussi homogène. Selon [Couturier, V., 2004 ; Frendi, M., 2013], on distingue trois types de transparence, à savoir :

- ❖ La transparence de localisation : permet à l'utilisateur d'ignorer la localisation physique de l'information.
- ❖ La transparence au niveau de la modélisation des données : les utilisateurs n'ont pas besoin de connaître les différents noms, des entités ou attributs des sources d'information.
- ❖ La transparence de langage : permet aux utilisateurs ou l'apprentissage d'un langage de requête d'extraction des données qui n'est pas obligatoire pour les utilisateurs. La transparence implique la présence d'un niveau d'abstraction entre l'utilisateur et les sources d'information.

1.3.3. Taxonomie des systèmes d'information coopératifs

Dans le but de résoudre les problèmes de l'hétérogénéité structurelle et garantir l'autonomie des sources d'information locale, plusieurs solutions sont en charge. Ces solutions doivent assurer la bonne coopération des sources d'informations. Selon plusieurs travaux sur les SICs, ces derniers peuvent être classés en deux approches : l'approche fédérée et les approches à base de médiation. Dans ce qui suit on va présenter l'essentiel sur ces deux approches.

1.3.3.1. L'approche fédérée

D'après [Sheth, A., Larson, J., 1990], un système fédéré contrôle et coordonne la manipulation d'une collection de systèmes de gestion de bases de données autonomes. Chacun des systèmes de base de données participant à une fédération continue à être utilisé par un ensemble d'applications, de manière locale. L'approche fédérée est orientée vers l'intégration partielle des structures de données.

Les principes fondamentaux de l'approche fédérée nécessitent la présence de cinq niveaux de schémas suivants [Sheth, A., Larson, J., 1990] (Figure.I.1.) :

- Le schéma local représente le schéma de chaque base de fédération dans son modèle local.
- Le schéma composant résulte de la traduction d'un schéma local en un schéma exprimé à l'aide d'un modèle de données commun, appelé modèle canonique. Cette transformation permet de résoudre les problèmes d'hétérogénéité des modèles de données.
- Le schéma exporté est une vue d'un schéma composant spécifiant les données qui seront mises à disposition de la fédération.

- Le schéma fédéré peut consister à intégrer statiquement des schémas d'export ou peut être une vue dynamique sur plusieurs schémas fédérés répartis en fonction des applications clientes ou des utilisateurs. Un schéma fédéré correspond à un domaine d'application.
- Le schéma externe consiste en une vue d'un schéma fédéré adapté à une application cliente ou une catégorie d'utilisateurs. Les schémas externes permettent ainsi de limiter l'étendue et la complexité des schémas fédérés.

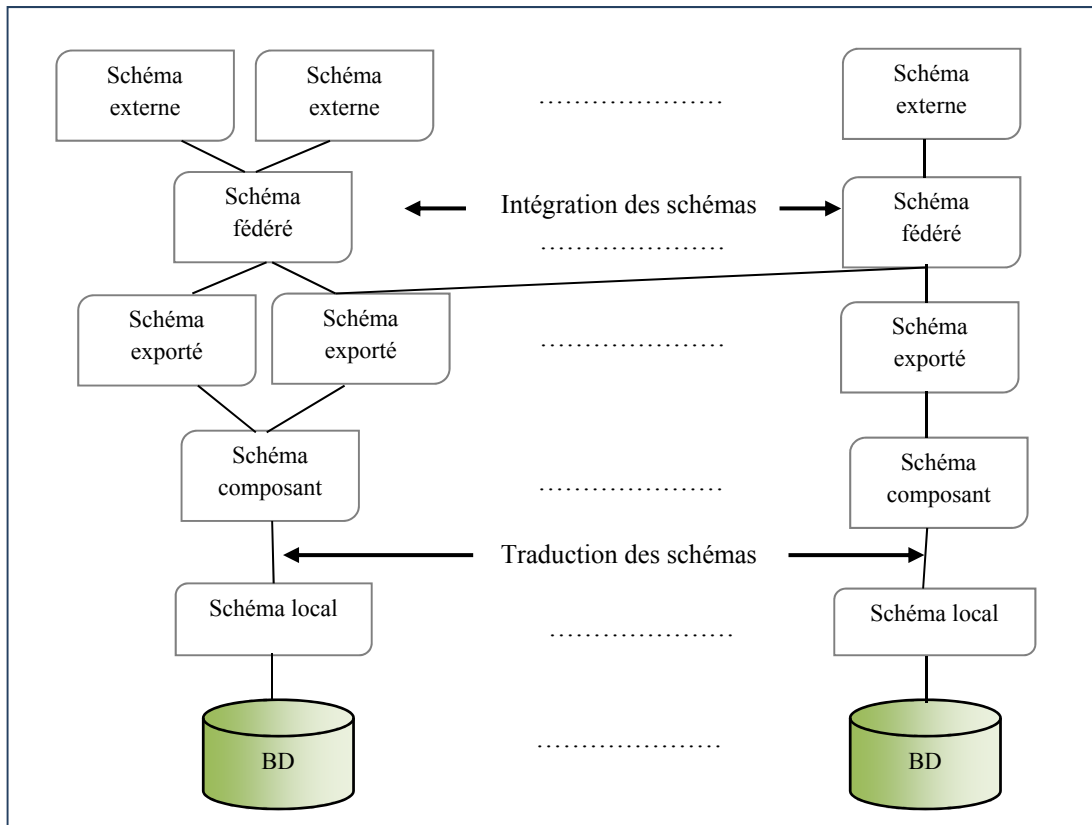


Figure.I.1. Les cinq niveaux de schémas de l'approche fédérée

Un modèle canonique est utilisé dans les quatre premiers niveaux de schémas, qui permettent d'homogénéiser les schémas locaux représentés dans différents modèles de données (relationnel, hiérarchique, objet, ...). Ce type de modèle est en général un modèle orienté objet, en raison de ses capacités d'abstraction et de représentation. Le modèle canonique assure la résolution de certains problèmes d'hétérogénéité structurelle.

Dans l'approche fédérée, il est nécessaire la présence d'un dictionnaire de données (répertoire des méta-données des schémas, ...), appelé dictionnaire de fédération, afin de pouvoir réaliser l'intégration. Selon [Sheth, A., Larson, J., 1990], il existe deux types d'intégration ; l'intégration sous forme de schémas fédérés (couplage fort) et l'intégration sous forme de vues fédérant plusieurs SIs (couplage faible).

➤ *Les systèmes fédérés fortement intégrés (couplés) :*

Ce type de systèmes permet aux applications ou utilisateurs de formuler leurs requêtes sur un schéma fédéré « global » unifié. Les schémas exportés par les différentes sources de données sont ainsi intégrés en un schéma fédéré unique. La définition d'un schéma fédéré global demande, d'une part, l'analyse et la comparaison des différents schémas et, d'autre part, la résolution des conflits éventuels [Kim, W. et al, 1994].

Un tel système possède les caractéristiques suivantes [Frendi, M., 2013] :

- ✓ La transparence de localisation des données et la distribution des données,
- ✓ L'autonomie des sources locales n'est pas pleinement respectée,
- ✓ L'administration de la fédération a un contrôle total sur la création et la maintenance des schémas fédérés et des schémas exportés,
- ✓ Les schémas fédérés sont statiques et donc rarement modifiés.

➤ *Les systèmes fédérés faiblement intégrés (couplés) :*

Les systèmes fédérés faiblement intégrés ne demandent pas la définition d'un schéma global. L'intégration des informations des différents systèmes composants est réalisée en fonction des besoins des utilisateurs. Ce type de systèmes permet l'intégration de nouvelles sources de données.

Les caractéristiques de tels systèmes sont [Frendi, M., 2013] :

- ✓ Les schémas fédérés peuvent être créés ou supprimés selon les besoins,
- ✓ L'utilisateur de la fédération est responsable de la création et de la maintenance de la fédération,
- ✓ L'utilisateur doit être bien informé sur la structure et l'information contenue dans les schémas exportés,
- ✓ Les composants locaux sont autonomes mais uniquement disponibles en lecture seule,
- ✓ L'évolutivité des composants locaux.

1.3.3.2. Les approches à base de médiation

Les approches à base de médiation fournissent une organisation modulaire des SIs de manière à tirer partie de l'accès à de nombreuses sources de données connectées par des réseaux. La médiation repose sur un composant central, appelé **médiateur**, qui simplifie, abstrait, réduit, combine et décrit les données [Wiederhold, G., 1992 ; Wiederhold, G., Genesereth, M. 1995 ; Frendi, M., 2013]. Le médiateur permet d'encoder les connaissances nécessaires aux transformations sur les données des sources. Il peut être spécialisé pour un ensemble de sources ou un domaine d'application. Plusieurs médiateurs peuvent être composés en une hiérarchie de modules entre l'application et un grand nombre de sources de données.

La fusion des connaissances relatives aux systèmes de bases de données fédérés est l'objectif principal de l'infrastructure de médiation, qui va permettre l'intégration des données dans des composants utilisables par plusieurs applications. Cette infrastructure possède la propriété de la modularité qui permet facilement l'évolution et l'intégration de nouvelles sources.

D'après Wiederhold [Wiederhold, G., 1992], les idées et les concepts relatifs aux systèmes de bases de données fédérés ont été repris par la plupart des projets de médiation qui ont adopté une même architecture pour leur SIC. La figure (Figure.I.2.) décrit cette architecture basée sur trois niveaux : le premier niveau concerne les SIs locaux, le deuxième niveau englobe les différents outils pour traiter les requêtes et résoudre les conflits et le troisième niveau est dédié aux applications et utilisateurs.

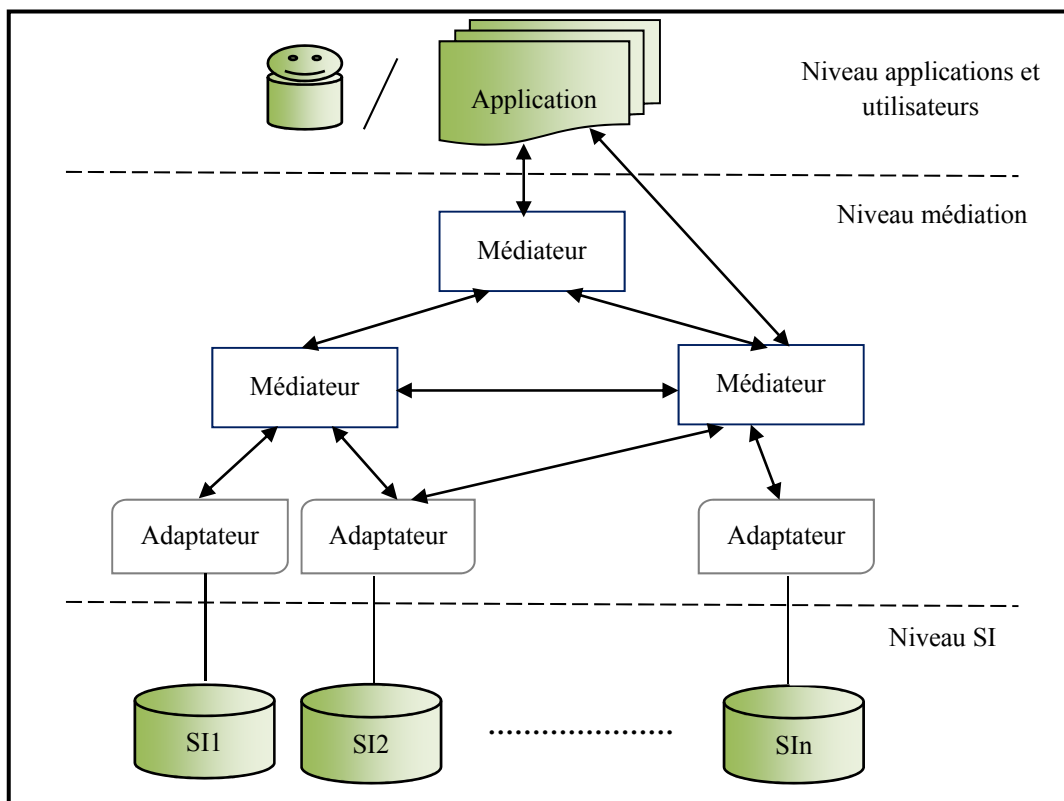


Figure.I.2. Architecture basée sur un système de médiation

Le deuxième niveau d'un système de médiation est composé par un ensemble des médiateurs et des adaptateurs. Au niveau de chaque source de données est associé un adaptateur. Chaque adaptateur fournit la même interface d'accès vers les sources des données et masque ainsi l'hétérogénéité de la source associée. Le médiateur fournit un point d'entrée unique et une vue uniforme sur un ensemble de sources de données. L'application permet de soumettre de requêtes déclaratives au médiateur dont le but d'obtenir les données demandées par l'utilisateur.

Un adaptateur doit être développé à chaque intégration d'une nouvelle source de données. Le médiateur va ensuite enregistrer cet adaptateur. Lorsque l'application soumet une requête, celle-ci

est optimisée par le médiateur, elle est ensuite exécutée. Le moteur d'exécution du médiateur envoie de sous-requêtes aux sources de données par l'intermédiaire des adaptateurs : chaque sous-requête envoyée par le médiateur est transformée par un adaptateur en une requête locale au format de la source de données. La réponse de la source est ensuite transformée par l'adaptateur au format du médiateur. Dans une dernière étape, le médiateur rassemble les résultats fournis par toutes les sources de données impliquées dans la requête, les combine et compose la réponse qu'il renvoie à l'application. Lors de cette étape, le médiateur résout certains conflits structurels et sémantiques [Couturier, V., 2004 ; Frendi, M., 2013]. L'évolution du nombre d'adaptateurs est le résultat d'intégration de nouvelles sources de données. Un système de médiation doit ainsi fournir les outils pour le développement rapide d'adaptateurs.

Dans les projets récents, les concepts de médiateur et d'adaptateur sont utilisés sous formes d'agents intelligents à cause de leurs fonctionnalités de distribution des connaissances et de l'exécution des tâches [Chawathe, S. et al, 1994 ; Genesereth, M.R. 1997].

Selon la manière de la gestion des requêtes et de l'utilisation de la sémantique dans le processus de médiation, on peut distinguer deux types de médiation [Jouanot, K., 2000 ; Jouanot, F., 2001] : la médiation de schémas et la médiation de contextes.

a. La médiation de schémas

Ce type de médiation va construire une base d'informations prenant en compte les SIs participants. Il associe un ensemble de connaissances au médiateur qui localisent et intègrent les informations pertinentes. Le schéma de médiation est formé d'interfaces qui peuvent être soit :

- ❖ Des interfaces objet : ces interfaces sont définies sous forme d'un ensemble de classes virtuelles. Le modèle ODMG'93 [Cattel, R. et al, 1993] c'est le modèle choisi pour les représenter. Les instances des classes virtuelles sont des instances d'objets distants attachés à un SI. Il existe une base de connaissances qui met en correspondance ces classes virtuelles avec les éléments des schémas locaux. Parmi les solutions mettant en œuvre des interfaces objet par exemple, DIOM [Liu, L., Pu, C., 1995] ou DISCO [Tomasich, A. et al, 1995] qui utilisent un modèle objet basé sur ODMG'93 ou le projet ACSIS [Boulanger, D., Dubois, G., 1997 ; Boulanger, D., Dubois, G., 2001 ; Dubois, G. et al, 2002] qui repose sur l'application d'un modèle objet spécifique.
- ❖ Des interfaces logiques : ce type d'interfaces basé sur l'utilisation d'un modèle logique. Les solutions mettant en œuvre des interfaces logiques sont, par exemple, TSIMMIS [Garcia-Molina, H. et al, 1997] ou YAT [Simeon, J., 1999].
- ❖ Des interfaces objet XML : le langage XML est utilisé comme modèle commun pour définir les interfaces de coopération et remplace le modèle objet classique. Parmi les projets mettant en œuvre des interfaces objet XML, on peut citer MIX [Ludascher, B. et al, 1999].

La médiation de schémas est une extension directe de l'approche fédérée avec une meilleure évolutivité et souvent une meilleure scalabilité, notamment souvent permise par l'utilisation d'agents intelligents et de systèmes multi-agents [Couturier, V., 2004].

b. La médiation de contextes

La médiation de contextes est basée sur une représentation explicite de la sémantique des données à travers la notion de contexte. Ce dernier peut être décrit à l'aide d'outils de type méta-donné [Libourel, Th. et al, 2003] ou ontologies. Un contexte montre un ensemble de connaissances sur la structure, les valeurs ou les fonctionnalités d'un objet.

Ce type de médiation repose sur l'intégration dynamique des informations en fonction du contexte de l'application ou de l'utilisateur. Le contexte de ces informations est décrit par chaque source d'information et doit se définir par rapport à un modèle de référence. Un contexte d'utilisation du modèle de référence est spécifié par chaque système composant. Dans la majorité des approches, le médiateur qui va utiliser les concepts et le vocabulaire d'une ontologie est associé à un contexte d'application, l'accès à un médiateur particulier définit alors le contexte d'une application. On peut distinguer deux types de médiation de contextes :

- L'approche mono-domaine repose sur un domaine unique dont le but de décrire la sémantique des informations. Pour représenter la connaissance du domaine d'application, un médiateur est associé à une ontologie unique. Les SICs basés sur la médiation de contextes et mettant en œuvre une approche mono-domaine sont, par exemple, SIMS [Arens, Y. et al, 1993] et COIN [Goh, C., 1997].
- L'approche multi-domaines repose sur la distribution des informations sur des domaines de connaissances différents. Un médiateur devra disposer d'outils spécifiques pour comparer les domaines distincts. Ces différents outils devront permettre de définir les relations de synonymie, d'homonymie ou d'antonymie entre les termes désignant les concepts d'un domaine (termes d'une ontologie). Et aussi les relations inter-ontologies explicitant les règles de mise en correspondance entre les concepts de domaines différents. Le médiateur doit prendre en charge les ontologies liées aux différents domaines de connaissances. Les solutions de médiation de contextes multi-domaines sont, par exemple, OBSERVER [Mena, E., 1999] et INFOSLEUTH [Bayardo, R. et al, 1997].

Dans la médiation de contextes, la résolution dynamique des conflits est assurée et la connaissance a priori des SI participants est ignorée (pas de schéma de médiation).

1.3.3.3. Comparaison entre les approches fédérées et les approches à base de médiation

Dans cette section, on va montrer une comparaison entre les différentes approches coopératives présentées dans les sections précédentes, dont le but de faire une étude sur le développement de la coopération, de découvrir les concepts et les techniques de base utilisés au sein des SICs. Aussi pour montrer la nature de l'approche utilisée selon nos besoins dans notre travail.

Selon [Frendi, M., 2013], on peut dire que chaque approche est évaluée en fonction des points suivants, qu'on va les détailler par la suite :

- ✓ Modèle canonique utilisé,
- ✓ Langage d'interrogation de la coopération,
- ✓ Représentation de la sémantique du domaine d'application,
- ✓ Respect des propriétés pour la coopération,
- ✓ Types d'hétérogénéité prise en compte,
- ✓ Modèle d'architecture.

Ces différents points de comparaison permettent de présenter les points communs ou les différences entre les différentes approches coopératives ou entre les SICs. D'après [Ouzzani, M., Bouguattaya, B., 2004], les approches, ou plus exactement les SICs, peuvent être comparés selon d'autres points, comme par exemple, les techniques d'optimisation ou de traitement des requêtes, etc.

a. Modèle canonique

Tout simplement, un modèle canonique est un modèle commun de représentation des données qui assure l'homogénéité des modèles locaux. Les schémas des sources de données locales sont traduits en schémas composants à l'aide de ce modèle. Ce modèle peut être :

- Relationnel,
- Orienté-objet : modèles objet issus des standards ODL (Object Definition Language) de l'ODMG (Object Database Management Group), OGM (Open Geodata Model) , ...
- Logique : c'est un modèle basé sur la logique terminologique tels KL-ONE, LOOM ou KIF (Knowledge Interface Format) ou modèle basé sur la logique déductive tel LDL (Logical Data Language , Frame-Logic ou Prolog),
- Objet XML
- Propriétaire : c'est une extension d'un modèle objet ou logique avec des concepts spécifiques adaptés à la représentation de données particulières : par exemple, FDS (Formal Data Structure) pour les données spatiales, ...

Généralement, un certains SIC, relatifs aux approches à base de médiation de contextes, n'utilisent pas de modèles canoniques.

b. Langage d'interrogation de la coopération (langage global)

Un Langage d'interrogation de la coopération permet l'accès uniforme aux informations distribuées (données et connaissances). Il est lié au modèle utilisé pour représenter les données (modèle canonique) et peut être :

- Un langage de programmation : souvent est associé à un modèle relationnel ou orienté-objet et souvent est utilisé en combinaison avec un langage déclaratif. Par exemple : C, C++, Java. Ce dernier c'est le langage utilisé dans notre approche.
- Un langage déclaratif : est associé à un modèle relationnel (SQL), orienté-objet (OQL, GeoOQL (GeoPOMQueryLanguage)), voire logique (par exemple, SQL associé à un modèle de logique terminologique KIF ou à un modèle de logique déductive LDL).
- Un langage logique : ce langage est associé à un modèle relationnel dans certains projets. Par exemple : Prolog pour un modèle déductif ou KL-ONE, LOOM ou KIF pour un modèle terminologique, ...

Un certains SIC soit n'utilisent pas un langage d'interrogation de la coopération, soit combinent plusieurs types de langages (par exemple, un langage déclaratif et un langage logique).

c. Représentation de la sémantique du domaine d'application

Pour la représentation de la sémantique du domaine d'application, il est distingué quatre types qui sont :

- Le schéma : permet de décrire la manière pour structurer les informations d'un domaine. Dont le but d'exprimer le domaine d'application, plusieurs projets va utiliser un schéma de données.
- L'interface objet : c'est le point d'accès au système pour l'interroger. Elle permet de décrire la sémantique du domaine d'application et nécessite l'utilisation d'un langage spécifique.
- L'annuaire : selon une thématique particulière, l'annuaire permet de localiser un SI et aussi permet de regrouper une description résumée des différents systèmes d'un environnement de coopération.
- L'ontologie : permet de définir les concepts d'un domaine d'application. L'ontologie peut soit être limitée à un vocabulaire de référence, soit utiliser un schéma terminologique ou logique. Chaque projet, basé sur l'usage d'ontologies, utilise un langage de règles pour formaliser les concepts et relations entre ces concepts : F-Logic, KIF, ONTOLINGUA, Dans notre travail, nous avons utilisé une ontologie de domaine qui va permettre de définir les principaux concepts utilisés par notre système.

d. Respect des propriétés de la coopération

Les différents projets respectent, à des degrés différents, les propriétés requises pour la coopération, à savoir, l'autonomie, la distribution, l'hétérogénéité, la transparence, l'évolutivité et la scalabilité (voire section 3.2.).

- Dans la majorité des projets et surtout de type médiation, les propriétés de distribution et d'hétérogénéité sont bien respectées. Et aussi pour l'autonomie ; l'autonomie de conception est respectés à cause d'utiliser les schémas d'export ou d'adaptateurs. L'autonomie d'association est assurée principalement dans les approches de type médiation de contextes.
- La transparence permet à l'utilisateur de la coopération de formuler une requête globale sans intéresser à la modélisation et au langage d'interrogation des données. L'utilisation du schéma global dans les approches fédérées à couplage fort permet de résoudre les problèmes de localisation et de modélisation des données, aussi permet d'utiliser un langage d'interrogation unique. Dans ce cas, la gestion de la transparence est à la charge de l'administrateur de la fédération. Et dans le cas des approches fédérées à couplage faible, la gestion de la transparence est gérée par l'utilisateur et est donc moins bien respectée. Plusieurs types d'outils sont fournis dans les solutions de type médiation ce qui va permettre d'assurer la transparence à la localisation et de respecter l'indépendance du modèle local de représentation des données.
- L'intégration dynamique des informations en fonction de la sémantique de la requête ou de la source d'information permet de créer l'évolutivité. Cette dernière est une caractéristique de base des systèmes de médiation de contextes et pour les autres approches, il est demandé une modification des connaissances globales. Autrement dit, l'évolutivité sera ainsi respectée en fonction de la présence ou non d'outils pour simplifier la gestion de ces modifications. En général, s'il existe une quantité importante de sources, l'évolutivité dans les approches fédérées à couplage faible nécessite des mises à jour importantes au niveau local. Et dans le cas des approches fédérées à couplage fort, l'administrateur de la fédération devra procéder à une mise à jour manuelle de la coopération qui pourra être facilitée par la présence d'outils.
- La scalabilité est une propriété résultante par les différentes caractéristiques relatives à la résolution dynamique des requêtes, et aussi par l'utilisation d'une ou plusieurs ontologies. Donc c'est une propriété de base des systèmes à base de médiation de contextes. Un degré de scalabilité est assuré dans les approches fédérées faiblement couplées du fait que l'ajout de nouvelles sources reste relativement aisé en raison de la simplicité de leur modèle de données. Contrairement dans les approches à base de médiation de schémas nécessitent lors de l'ajout de systèmes composant, la mise à jour d'un ou plusieurs médiateurs sous forme de hiérarchies ou la présence d'outils spécifiques permet cependant d'accroître la scalabilité de telles architectures.

Pour garantir que notre système soit bien coopératif dont le but de résoudre n'importe quelle problème distribué, les différentes propriétés sont respectées à des degrés différents dans notre approche.

e. Types d'hétérogénéité pris en compte

La résolution des conflits de modèles de données et de langages d'interrogation est assurée dans l'approche fédérée. Les conflits structurels et certains conflits sémantiques sont traités dans les approches à base de médiation de schémas ou médiation de contextes.

f. Modèles d'architecture

Les architectures proposées ont généralement suivi l'évolution des techniques informatiques. Plusieurs types d'architectures peuvent être distingués :

- L'architecture 3-tiers : l'application cliente manipule un module applicatif qui est le seul à accéder aux sources d'information et à traiter la requête globale.
- Les objets distribués : ce type d'architecture est une version évoluée des applications 3-tiers, où les fonctions du module applicatif sont réparties sur plusieurs plates-formes
- Les adaptateurs (wrappers) : dans le cas des systèmes à base de wrapper et selon une interface uniforme représentée dans un modèle commun, l'application cliente accède directement aux sources d'information locales. Ce type d'architecture n'est pas adapté à la résolution des problèmes d'hétérogénéité sémantique.
- Les médiateurs : l'accès transparent aux différentes sources de données est assuré par le médiateur. Les architectures peuvent éventuellement être composées de plusieurs médiateurs regroupés par domaine d'application.
- Les agents artificiels : dans les projets de SIC (tel le cas dans notre approche), les agents artificiels sont les plus utilisés. Selon Briot et Demazeau [Briot, J.P., Demazeau, Y., 2001], un agent artificiel est « une entité logicielle ou physique à qui est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents ». Différents points de vue sur les agents artificiels correspondant à différents types d'agents peuvent être rencontrés dans la littérature, on va les détailler dans le chapitre II.

Peu importe le type de SIC, l'amélioration du processus de la prise de décision dans les SICs est l'objectif principale de notre système. Donc, parmi les points essentielles qui peut être dans notre SIC sont :

- ✓ Un langage d'interrogation de la coopération est utilisé,
- ✓ Pour la représentation de la sémantique du domaine d'application, le concept d'ontologie est utilisé qui permet de définir les concepts du domaine l'application.

- ✓ La majorité des propriétés pour la coopération sont utilisées dans notre système,
- ✓ Et pour le modèle d'architecture, les agents artificiels sont l'architecture utilisés dans notre système.....

1.4. Conclusion

Ce premier chapitre nous a permis de présenter les fondements de base des Systèmes d'Information Coopératifs. Nous avons tout d'abord présenté la définition d'un SI. Ensuite, nous avons présenté le terme de SIC ; sa définition, leurs principales propriétés et caractéristiques et enfin leur taxonomie. L'objectif principal de ce chapitre est de présenter les concepts et les techniques de base utilisés au sein des SICs. Dans ces derniers, nous cherchons également à améliorer l'exécution du processus de la prise de décisions afin de trouver les meilleures solutions à n'importe quel problème distribué posé.

Le chapitre suivant est consacré à un autre paradigme que nous avons utilisé dans notre solution, c'est l'étude sur l'environnement multi agents, dont lequel nous pouvons utiliser et appliquer les cartes cognitives comme un outil d'aide à la décision dans le contexte des SICs.

Chapitre II :

Les agents et les systèmes multi- agents

II.1. Introduction

Le besoin de la distribution des connaissances et de contrôle dans les systèmes d'intelligence artificielle a donné naissance au paradigme des Systèmes Multi-Agents (SMA). Ces derniers sont apparus à la fin des années 70 et au début des années 80. Ce chapitre a pour vocation de présenter le contexte du domaine des SMA. Nous l'abordons d'abord sous l'aspect « agent » en présentant les définitions, les caractéristiques et aussi les architectures relatives à cette unité de base du SMA. Puis, nous passons à la dimension collective du système en présentant les définitions, les caractéristiques telles que l'autonomie, la coopération, la coordination, l'organisation, la négociation...etc. et les différents types d'un SMA. Nous terminons le chapitre par quelques méthodes de conception ainsi que les plates-formes de développement des SMA les plus utilisées.

II.2. Agent

Pour comprendre le paradigme du SMA, il est important de définir tout d'abord le concept d'agent. Dans ce qui suit, on va présenter les grandes lignes de ce concept.

II.2.1. Définitions

Il n'existe pas une définition standard du concept d'agent relative à l'implémentation. Elle reste encore très controversée à propos de sa signification. Même si les efforts de standardisation des aspects opérationnels de la conception SMA sont de plus en plus reconnus.

Plusieurs définitions ont été proposées dans le domaine des SMA. Par exemple selon *Ferber*, un agent est une entité physique ou virtuelle [Ferber, J., 1995]:

1. capable d'agir sur elle-même et sur son environnement ;
2. capable de percevoir son environnement (avec une représentation partielle) ;
3. capable de communiquer avec d'autres agents ;
4. qui poursuit un objectif individuel ;
5. qui possède des compétences et peut éventuellement se reproduire ;
6. dont le comportement est la conséquence de toutes les propriétés énumérées ci- dessus.

D'après Yves Demazeau [Demazeau, Y., 1996] « *un agent est une entité réelle ou virtuelle dont le comportement est autonome, évoluant dans un environnement qu'il est capable de percevoir et sur lequel il est capable d'agir et d'interagir avec les autres agents* ».

D'autres auteurs comme Wang propose la définition suivante: « *An agent is a piece of autonomous software created by and acting on behalf of a user (or some other agent). It is set up to achieve a modest goal, with the characteristics of autonomy, interaction, reactivity to environment, as well as proactiveness* » [Wang, A.I. et al., 1999].

Pour Mickael Wooldridge « *un agent est un système informatique, situé dans un environnement, et qui est capable d'agir de façon autonome dans cet environnement pour atteindre les objectifs (buts) pour lesquels il a été conçu* » [Wooldridge, M., 2002].

D'un point de vue organisationnel, la fonction principale d'un agent est l'adoption d'un rôle qui contribue à l'objectif global de la société (communauté). L'objectif de la société peut être de déterminer les rôles des agents et les normes des interactions. Les agents peuvent être des acteurs exécutant des rôles décrits dans la structure organisationnelle de la communauté, ses capacités propres et ses objectifs déterminent de façon spécifique la manière avec laquelle il peut jouer un rôle. Plusieurs auteurs ont préconisé des approches axées sur le développement du rôle des agents dans une société [Zambonelli, F. et al., 2001].

Le comportement d'un agent résulte de ses perceptions, de ses connaissances, de ses compétences, et bien évidemment de son but [Jérémy, B., 2014].

Peu importe la définition que nous préférons, la plupart des auteurs sont d'accord sur le fait que tous les agents ont des perceptions et sont capables d'effectuer des actions. Bien entendu, la partie centrale dans la construction d'un agent est d'avoir un mécanisme de raisonnement efficace, lui permettant d'assurer le passage des perceptions jusqu'aux actions.

II.2.2. Caractéristiques

Il existe plusieurs caractéristiques relatives aux agents qui procurent la force au paradigme agent et qui les distinguent des autres paradigmes logiciels tels que les systèmes orientés objets, les systèmes distribués et les systèmes experts. Un agent ne doit pas posséder toutes ces caractéristiques en même temps, mais il suffit qu'il soit doté de certaines d'entre elles.

Dans ce qui suit, nous citons les plus intéressantes afin d'avoir une meilleure vision d'un agent [Ferber, J., 1995 ; Gouiza, F., 2016 ; Graja, Z., 2015 ; Mazyad, H., 2013 ; Nchet, B., 2014]:

- **L'autonomie** : un agent est autonome, c'est-à-dire qu'il doit pouvoir agir seul, prendre une décision seul, en fonction d'informations lui provenant d'autres agents, de l'environnement. Il gère lui-même son état interne en fonction des informations qui lui parviennent ;
- **La réactivité** : signifie que l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis. L'agent perçoit son environnement qui peut être l'utilisateur à travers une interface graphique, un ensemble d'agents, etc. il doit répondre dans les temps impartis aux changements qui surviennent sur cet environnement ;
- **La pro-activité (intentionnalité)** : un agent ne réagit pas simplement aux changements de l'environnement mais il fait preuve d'un comportement dirigé vers ses buts et prend des initiatives lorsqu'elles lui semblent appropriées ;
- **La perception** : l'agent peut avoir une vision très locale de son environnement, mais aussi une représentation plus large de cet environnement et notamment des agents qui l'entourent ;

- **La capacité à agir** : un agent est mu par un certain nombre d'objectifs qui guident ses actions, il ne répond pas simplement aux sollicitations de son environnement ;
- **La rationalité** : les agents rationnels disposent de critères d'évaluation de leurs actions et de choix des meilleures pour atteindre le but ;
- **L'adaptabilité ou la flexibilité** : un agent adaptable est un agent capable de moduler ses aptitudes selon l'état de l'environnement. L'agent est capable d'apprendre et sait résoudre un nouveau problème à partir de son expérience ;
- **La sociabilité** : un agent est capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin d'accomplir ses tâches et aide les autres à atteindre leurs buts ;
- **La compétition** : l'agent est capable d'agir dans un environnement où d'autres agents interviennent. Le but est le même pour tous les agents présents mais un seul l'atteindra, les autres échoueront ;
- **La communication** : est indispensable à l'agent avec d'autres agents (comportement social).

II.2.3. Architecture interne d'un agent

Le terme d'architecture interne désigne l'ensemble des structures de données et des processus internes à un agent lui permettant de prendre une décision consistant à choisir une action en vue de modifier l'environnement [Jennings, N. et al., 1998]. Généralement, un agent évolue dans un environnement. Par des récepteurs, il est capable de recevoir des informations de cet environnement, capable d'agir sur ce même environnement par des effecteurs, suivant un comportement décidé selon le raisonnement de l'agent. L'agent est caractérisé par son architecture et son comportement.

Il existe plusieurs architectures d'agents se distinguant et se différenciant les uns des autres soit par leurs caractéristiques, leur mode de communication ou par leur mode d'action. On peut cependant les regrouper en deux grandes catégories [Ferber, J., 1995 ; Atohou, B.C., 2013], les autres catégories n'étant que des sous-catégories ou des déclinaisons d'une de ces deux catégories [Gouiza, F., 2016]. Ces deux catégories fondamentales sont constituées des agents réactifs et des agents cognitifs.

La distinction qu'on peut faire entre cognitif et réactif tient essentiellement du processus décisionnel chez l'agent et de la représentation du monde dont il dispose. Les titres suivants décrivent ces deux approches comme suit :

II.2.3.1. Agents réactifs

Généralement, les agents réactifs ont un comportement du type « stimulus – réponse » (Figure II.1). Ils ne disposent ni de représentation du monde, ni de représentation de leurs actions qui consistent surtout en une réaction rapide face à un stimulus. Ils ont une faible capacité de

coopération qui est limitée à la communication par message, le plus souvent de façon indirecte. Un ensemble d'agents réactifs peut avoir un comportement complexe. Donc, Les agents à capacités réactives ne disposent que d'un protocole et d'un langage de communication réduit, ils n'ont pas une représentation explicite de leur environnement et ne sont pas capables de tenir compte de leurs actions passées. Ils ne possèdent pas de moyen de mémorisation. En effet, dès qu'ils perçoivent une modification de leur environnement, ils répondent par une action programmée.

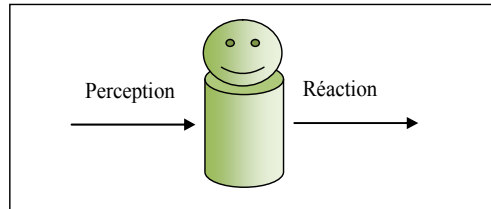


Figure II.1. Agent réactif

Les caractéristiques d'un agent réactif peuvent être définies par les points suivants [Ferber, J., 1995]:

- ✓ La prise de décision d'un agent est réalisée à travers un ensemble de modules comportementaux correspondant à la tâche à réaliser. Chaque comportement est implémenté sous la forme de la règle suivante : situation \rightarrow action ;
- ✓ Lorsque plusieurs comportements peuvent être déclenchés à un instant donné, la résolution de conflits peut être assurée par l'arrangement des modules d'action dans une hiérarchie.

Un des exemples d'architectures utilisées pour concevoir des agents réactifs est l'architecture de subsomption proposée par Brooks [Brooks, R.A., 1991]. Le principe de cette architecture est de concevoir le processus décisionnel en considérant un ordre de priorité entre des modules qui représentent les tâches pouvant être accomplies par l'agent. Les conditions d'activation de chaque module sont alors évaluées suivant cet ordre de manière à sélectionner la tâche à effectuer en priorité.

II.2.3.2. Agents cognitifs

Les agents cognitifs ont un comportement du type « perception- décision – action» (Figure II.2). Ils sont le résultat direct des recherches menées dans le domaine de l'intelligence artificielle. Ils disposent de représentation globale de leur environnement et des autres agents avec lesquels ils communiquent. Ils savent tenir compte de leur passé et s'organisent autour d'un mode social d'organisation. Au niveau des agents cognitifs le raisonnement est développée (décision), une base de connaissances est associé qui comprenant les diverses informations liées à leurs domaines d'expertise et à la gestion des interactions avec les autres agents et leur environnement, et aussi l'interaction permet aux agents de communiquer, de collaborer et d'agir... De ce fait, ils sont

capables de prendre des décisions à partir des informations dont ils disposent et de planifier leurs actions à l'avance.

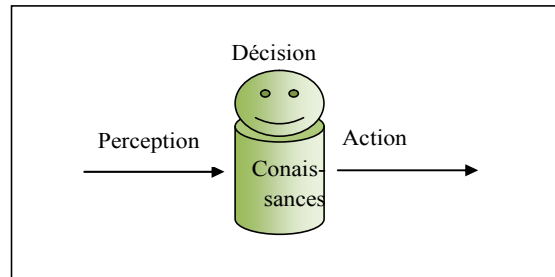


Figure II.2. Agent cognitif

L'une des architectures cognitives les plus connues est sans doute l'architecture BDI : Belief (Croyance), Desire (Désir), Intention (Intention) [Rao, A., Georgeff, M., 1995] qui comme son nom l'indique, est basée sur les notions que sont la croyance, le désir et l'intention :

- Les croyances correspondent aux informations dont dispose l'agent sur son environnement.
- Les désirs correspondent aux états de l'environnement que l'agent souhaiterait voir réalisés.
- Les intentions correspondent aux projets de l'agent pour satisfaire ses désirs.

II.3. Système multi-agents (SMA)

Pour pallier la limite d'un agent individuel, l'architecture multi-agents permet la répartition de la complexité d'un problème donné sur un ensemble d'agents au lieu de la centraliser au niveau d'un seul agent.

II.3.1. Définitions d'un SMA

Tout simplement, un SMA peut être considéré comme un ensemble d'agents partageant un environnement commun. La diversité des travaux de recherche sur les SMA a donné lieu à plusieurs définitions de ce paradigme. Dans ce qui suit, on va présenter quelques définitions.

Définition 1 : selon J. Ferber [Ferber, J., 1995], les SMA sont la conjonction de plusieurs disciplines de recherches qui sont l'intelligence artificielle pour l'aspect prise de décision de l'agent, les systèmes distribués et l'intelligence artificielle distribuée pour les interactions et la distribution de l'exécution, et finalement le génie logiciel pour l'évolution vers des composants logiciels de plus en plus autonomes et proactifs [Ferber, J., 1995]. Il définit un SMA comme un système composé des éléments suivants :

- ✓ Un environnement E, c'est-à-dire un espace disposant généralement d'une métrique.
- ✓ Un ensemble d'objets O. 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.

- ✓ Un ensemble A d'agents, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.
- ✓ Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
- ✓ Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O .
- ✓ 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.

Cette définition met en évidence la relation entre les agents pouvant influencer leurs objectifs dans un espace partagé appelé plus communément environnement.

Définition 2: Geen, S. et al., dans [Geen, S. et al., 1997] définissent un SMA en tant qu'« un réseau d'agents (solveurs) faiblement couplés qui **coopèrent ensemble** pour résoudre des problèmes qui dépassent les capacités ou les connaissances individuelles de chaque agent. Les agents sont **autonomes** et peuvent être de **natures hétérogènes** ».

Définition 3 : Un SMA est défini par K. Sycara et al. [Sycara, K. et al., 2010] comme l'émergence d'un comportement global produit par un ensemble d'interactions entre agents afin de résoudre des problèmes qui dépassent individuellement leurs capacités de raisonnements.

Définition 4 : Un SMA est tout au moins un système formé d'agents de même type ou non, qui coopèrent et interagissent de façon coordonnée dans le souci d'atteindre un but ou un objectif commun dans un environnement dont ils ont la même perception ou point de vue ou des perceptions ou points de vues complémentaires [Atohou, B.C., 2013].

Définition 5 : Un SMA est un ensemble d'agents en interaction dans un environnement commun, agissant pour résoudre une tâche commune et cohérente. Ce dernier point est important car il implique l'unité du SMA, malgré le fait que chaque agent possède son propre but individuel qui peut éventuellement entrer en conflit avec celui des autres. Cette composition particulière donne aux SMA des propriétés très intéressantes [Jérémy, B., 2014].

II.3.2. Caractéristiques d'un SMA

Selon les propriétés des agents énoncés dans la section 2.2, quelques caractéristiques d'un SMA peuvent être définies de la façon suivante [Ferber, J., 1999 ; Gasser, L., 1990 ; Gouiza, F., 2016 ; Graja, Z., 2015 ; Jennings, N. et al., 1998 ; Mazyad, H., 2013 ; Vernadet, F., Azena, P., 1992]:

- ❖ **L'autonomie :** le SMA est autonome lorsqu'aucune entité externe ne contrôle ce système. L'autonomie du système dérive de l'autonomie de ses agents.

- ❖ **La distribution:** dans un SMA, les données sont distribuées entre les différents agents que ce soit relatives à l'environnement ou relatives aux compétences des agents.
- ❖ **Système ouvert/système fermé :** un SMA est ouvert lorsqu'il est possible d'en rajouter/retirer des agents. Dans le cas contraire, il est dit fermé. Un agent est capable de se reproduire ou de se suicider.
- ❖ **Système homogène/système hétérogène :** le système est homogène lorsque les agents qui le composent ont le même type de perception et d'action ainsi que les mêmes compétences.
- ❖ **L'organisation :** un SMA est caractérisé par son organisation sociale, c'est à dire la manière dont le groupe est constitué, à un instant donné, pour pouvoir fonctionner.
- ❖ **L'interaction :** c'est la caractéristique principale d'un SMA, il désigne la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques.
- ❖ **La coopération :** dans un SMA, la coopération est la forme générale d'interaction. Une activité de coopération prend place si un ensemble particulier d'agents converge vers un but global par l'achèvement de leurs propres buts locaux.
- ❖ **La négociation :** c'est le processus d'améliorer les accords en réduisant les inconsistances et l'incertitude sur des points de vue communs ou des plans d'actions grâce à l'échange structuré d'informations pertinentes.
- ❖ **La coordination :** c'est un ensemble des actions individuelles accomplies par chacun des agents de manière à ce que l'ensemble aboutisse à un tout cohérent et performant. La coordination permet d'améliorer l'action de groupe en diminuant les conflits et en augmentant les performances.
- ❖ **La communication :** la communication dans les SMA est à la base des interactions et de l'organisation des agents mais aussi de la résolution coopérative des problèmes. L'absence de la communication implique l'absence de coopération, de négociation, de coordination des actions ou de réaliser des tâches en commun.

II.3.3. Types des SMA

Dans la littérature sur les SMA, Adam a proposé quatre types de système [Kenfack Dongmo, C.V., 2010] :

- **Les systèmes hiérarchiques :** qui représentent une architecture dans laquelle un niveau supérieur définit les contraintes et les objectifs à atteindre par le niveau suivant.
- **Les systèmes distribués :** qui est une architecture dans laquelle les agents sont connectés par l'intermédiaire d'un réseau pour atteindre un objectif donné. Ils sont physiquement distribués.

➤ **Les systèmes ouverts** : représentés par une architecture dans laquelle les agents n'ont pas besoin d'être conceptualisés ensemble pour partager un objectif commun. Ils ont la capacité de pouvoir dynamiquement quitter ou intégrer le système. Les agents peuvent être hétérogènes.

➤ **Les systèmes holoniques** :

Les systèmes multi-agents holoniques (SMAH) ont leurs propres propriétés qui les distinguent des SMA généraux. Un système holonique est une architecture dont la structure résulte de la combinaison de deux structures hiérarchique ou plus. On va détailler un peu cette architecture, puisqu'elle est exploitée dans notre travail.

La théorie des structures holoniques a été proposée par Arthur Koestler en 1967. Cette théorie implique une structure qui est composée de composants appelés «Holons». Un holon est une structure auto-similaire ou fractale qui est stable et cohérente et se compose de plusieurs holons comme sous-structures [Koestler, A., 1968]. Chaque holon peut contenir plusieurs sous-holons, et peut faire partie d'un holon plus grand lui-même. De cette manière, une structure hiérarchique se forme [Arabani, A.B. et al., 2009 ; Fleyeh, H., Davami, E., 2013].

Le concept d'Holon résulte de l'association du terme grec « holos » qui signifie le « tout » et du suffixe « -on » qui réfère à l'entité ou la particule comme dans neutron, proton ou électron.

Les SMAH sont des systèmes où les agents sont composés d'agents et s'emboîtent les uns dans les autres. Comme il est montré dans la figure II.3, l'idée de ce type de systèmes considère que des agents peuvent se regrouper pour créer un agent de niveau supérieur ou qu'un agent peut être décomposé en plusieurs agents de niveau inférieur, c-à-d que les holons de niveau n qui sont regroupés en organisation au niveau $n+1$ peuvent être considérés comme des entités individuelles. Inversement des holons considérés comme des entités individuelles de niveau $n + 1$ peuvent être considérés comme des organisations au niveau n .

Dans les années 90, le modèle holonique a été utilisé comme un nouveau paradigme de modélisation. La naissance de ce paradigme est de fait qu'un groupe, une organisation ou une société d'agents peuvent à un certain niveau d'abstraction, se comporter comme une entité individuelle. De nombreuses approches ont tenté de modéliser ce type des systèmes. Plusieurs noms sont spécifiés à ce type d'agent holonique « agents *collectifs* et *individuels*, *méta-agent*, *agents intermédiaires*, *agents récursifs*».

L'idée principale de SMAH est d'assigner une tâche qu'un seul agent ne peut accomplir à un holon. Ce dernier décide ensuite comment effectuer cette tâche. De cette façon, un objectif se divise en plusieurs sous-objectifs, et chaque holon est responsable d'un ou plusieurs sous-objectifs [Beheshti, R. et al., 2016].

L'autonomie limitée est la propriété clé des agents holoniques. Dans un SMAH, les agents sont rationnels et décident quoi faire en fonction de leurs propres préférences comme agents généraux dans un SMA. Mais, à partir du moment où un agent joint un holon, il ne peut pas

faire ce qu'il veut. Il doit obéir aux engagements de Holon après cela. Cela ne signifie pas que chaque action que l'agent fait est déterminée par la tête du holon comme SMA pleinement coopératif. Un agent peut habituellement décider quand rejoindre un holon, ou laisser le holon.

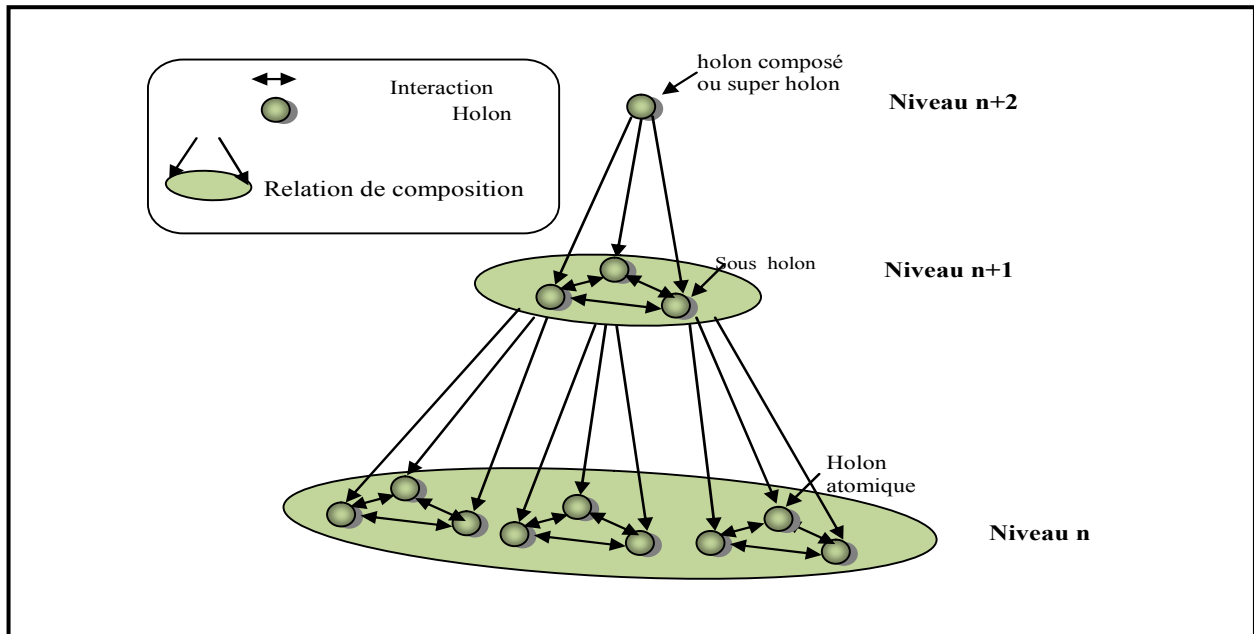


Figure II.3. Un exemple de SMAH

Les systèmes holoniques ont été appliqués dans de nombreux autres domaines tels que la santé, les systèmes de transport, les problèmes de couverture dans les réseaux radio-mobiles, les robots footballeurs, ou encore le travail collaboratif.

II.3.4. Méthodes de conception de SMA

Pour développer et réaliser les SMA, il est nécessaire d'appliquer des méthodes et d'outils. Dans la littérature, il existe une multitude de méthodes de développement orienté agent. Une méthode de conception [Picard, G., 2004] est constituée d'un processus de développement et de notations permettant la modélisation ainsi que des outils d'aide à la conception, à la vérification et/ou au suivi du processus. L'intérêt principal d'une méthode pour les SMA consiste à proposer une démarche qui permettra au concepteur d'être guidé afin de passer d'un cahier des charges à une implémentation et de pouvoir gérer le cycle de vie global d'une application. A cet effet, les chercheurs dans ce domaine se sont basés sur des modèles, méthodes et outils déjà existants. Suite à cela, plusieurs méthodes de conception de SMA ont vu le jour [Gleizes, M.P. et al., 2008 ; Gutknecht, O., 2001 ; Jarras, I., Chaib-draa, B., 2002 ; Mazyad, H., 2013 ; Nachet, B., 2014 ; Picard, G., 2004]. Dans ce qui, on va présenter quelques méthodes les plus connues.

II.3.4.1. MaSE

MaSE (Multiagent Software Engineering) est une approche complète pour le développement des SMA de l'analyse au déploiement [Deloach, S., 2001]. L'objectif principal de cette approche est d'aider le concepteur à l'analyse, la conception et aussi l'implémentation d'un SMA à partir d'un cahier de charge initial.

II.3.4.2. AAI

La méthodologie AAI (Australian Artificial Intelligence Institute) a été développée en se basant sur l'expérience accumulée dans les travaux effectués pour la gestion du trafic aérien [Kinny, D. et al., 1996]. Elle constitue un mélange de concepts des méthodologies orientées objet avec certains concepts orientés agents. Cette méthodologie permet la construction d'un ensemble de modèles qui définissent des spécifications du système d'agent à base d'architecture BDI.

II.3.4.3. DESIRE

DESIRE [Brazier, F.M.T. et al., 1997] (Design and Specification of Interacting Reasoning components) est directement issue de l'ingénierie des connaissances. Un modèle est la base de cette approche qui permet de traiter la connaissance, l'interaction, la coordination des tâches et les possibilités de raisonnement dans les SMA.

II.3.4.4. Aalaadin

Aalaadin est une approche organisationnelle pour le développement des SMA fondée sur le concept AGR (Agent, Groupes et Rôles) [Ferber, J. et al., 2003]. Elle permet de définir les indications méthodologiques et aussi un environnement de prototypage et d'exécution pour des agents qui se basent sur les notions du modèle AGR.

II.3.4.5. MAS-CommonKADS

La méthode MAS-CommonKads [Iglesias, C.A. et al., 1998] est une extension de CommonKads qui ajoute, à des techniques venant des méthodes orientées objet, des techniques de conception des protocoles dans le but de modéliser les agents et les interactions entre eux. Plusieurs modèles existent dans cette méthode: le modèle d'agent, le modèle d'expertise, et le modèle de conception.

II.3.4.6. GAIA

GAIA est une méthodologie qui utilise une approche centrée sur l'organisation pour analyser et concevoir un SMA. Elle reprend et développe les concepts d'Aalaadin [Wooldridge, M.J., 2000]. Cette méthodologie s'inspire de la méthodologie orientée objet FUSION [Coleman, D. et al.,

1994]. Les principaux concepts de GAIA sont : Rôle, Permission, Responsabilité, Protocole, Propriété d'animation et Propriété de sécurité.

II.3.4.7. Voyelles

Voyelles est une approche proposée par Demazeau [Briot, J.P., Demazeau, Y., 2001], elle est une méthode de haut niveau très souvent citée dans la documentation car reposant sur des principes purement multi-agents. Elle se base sur la décomposition de la vue d'un système suivant cinq dimensions : Agent, Environnement, Interaction, Organisation et Utilisateurs. L'approche Voyelles est l'approche utilisée dans notre travail.

II.3.4.8. ADELFE

La méthode ADELFE (Atelier de DEveloppement de Logiciel à Fonctionnalité Emergente) [Picard, G., Gleizes, M.P., 2004] est dédiée à la conception de SMA adaptatifs. Elle se caractérise par l'absence de structure organisationnelle, règles d'organisation, rôles etc. le concepteur doit exprimer sa structure organisationnelle sous forme de règles de coopération propres à chaque agent. De plus, l'absence du concept « rôle » rend moins flexible la définition du comportement de l'agent [Nachet, B., 2014].

II.3.4.9. PASSI

La méthodologie PASSI (Process for Agent Societies Specification and Implementation) [Mazyad, H., 2013] est basée sur les modèles de conception et concepts provenant de l'ingénierie orientée objet et de l'intelligence artificielle en utilisant UML. Elle a la particularité de prendre en considération la modélisation des agents mobiles. Cette modélisation respecte les standards définis par FIPA.

II.3.5. Plates-formes de développement des SMA

Une plateforme de développement des SMA est une infrastructure de logiciels utilisée comme environnement pour le déploiement et l'exécution d'un ensemble d'agents. Elle devrait fournir des fonctionnalités confortables pour créer et tester des agents, elle peut être vue comme une collection de services offerts aux développeurs, mais également aux agents en exécution.

Il existe un certain nombre de plates-formes fournies comme logiciels libres. Dans ce qui suit, on va présenter quelques plates-formes. Dont les trois dernières représentent les plateformes utilisées tout au long de notre travail.

II.3.5.1. Madkit

MadKit [Madkit, 2018] (Multi-Agents Développement Kit) est une plateforme multi-agents modulaire écrite en Java et conçue selon le modèle d'organisation Alaadin ou AGR (Agent/Group/Role). MadKit est développée en 1996 par Olivier GETKNECHT et Michel FERBER au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) de l'Université Montpellier II. MadKit est un logiciel libre de type "Open Source" avec une licence mixte GPL/LGPL. Il fonctionne en mode distribué de manière transparente à partir d'une architecture "peer to peer" sans nécessiter de serveur dédié. Il est ainsi possible de faire communiquer des agents à distance sans avoir à se préoccuper des problèmes de communication qui sont gérés par la plateforme.

II.3.5.2. RepastHPC

RepastHPC [Collier, N. et North, M., 2011] est développé par le laboratoire national d'Argonne (USA). Il fait partie de la série de plateformes de simulation multi-agents RepastJ, RepastSymphony. Il est spécialement conçu pour les environnements hauts performance. RepastHPC reprend le cor de RepastSymphony, c'est-à-dire l'utilisation des projections (grid, network) et des contextes mais en les adaptant à l'environnement parallèle et distribué. Le langage utilisé pour implémenter une simulation agent est le C++ ou le ReLogo, un dérivé du langage Netlogo. Pour la communication, la plateforme RepastHPC utilise MPI par l'intermédiaire de la librairie Boost.

II.3.5.3. D-Mason

D-Mason (Distributed Mason) [Cordasco, G. et al., 2011] est développé par l'université de Salerne. D-Mason est la version distribuée de la plateforme de simulation multi-agent MASON. Les auteurs ont souhaité élaborer la version distribuée de Mason afin de fournir une solution qui n'oblige pas les utilisateurs à réécrire les simulations qu'ils ont déjà développées pour repousser le nombre maximal d'agents. D-Mason utilise JMS ActiveMQ pour la communication, malgré le fait qu'elle ne soit pas la plus extensible des solutions dans un environnement HPC. D-Mason est basé sur le langage Java.

II.3.5.4. FLAME

FLAME [Coakley, S. et al, 2012] est développé principalement à l'Université de Sheffield. elle a été conçue pour permettre une large gamme de modèles d'agents. elle fournit des spécifications sous la forme d'un cadre formel qui peut être utilisé par les développeurs pour créer des modèles et des outils. FLAME permet la parallélisation à l'aide de MPI. L'implémentation d'une simulation

FLAME est basée sur la définition de X-machines qui sont définies comme des automates d'états finis. Ils peuvent, en plus, recevoir et envoyer des messages à l'entrée et à la sortie de chaque état.

II.3.5.5. JADE

JADE [Jade, 2014] (Java Agent DEveloppement) est un framework de développement de SMA, open-source et basé sur le langage Java. Elle est conforme aux normes et spécifications de FIPA (Foundation For Intelligent Physical Agents) [FIPA, 2002], elle hérite de son architecture : protocoles, services et mécanismes de transport et de communication. JADE est compatible avec la plupart des configurations matérielles. La version de Java (JRE ou JDK) est le seul système nécessaire pour son fonctionnement. L'agent JADE est implémenté en Java. Ce dernier est un système expert qui permet d'implémenter le raisonnement à base de règles. Les agents JADE communiquent à travers le langage ACL (Agent Communication Language) de FIPA.

II.3.5.6. MAGIQUE

MAGIQUE [Magique, 2015] est une plate-forme pour agents physiquement distribués écrite en Java et fournissant un modèle de communication original d'appel à la cantonade. Dans MAGIQUE, les compétences sont dissociées des agents. L'architecture d'agents et les différentes compétences sont développées séparément, les compétences sont ensuite "pluguées" dans les agents au grès du concepteur. Cette plate-forme est développée au sein du LIFL.

II.3.5.7. AgentBuilder

AgentBuilder [AgentBuilder, 2016] est une suite intégrée d'outils permettant de construire des agents intelligents, elle a été développée en JAVA par Reticular Systems Inc. L'élaboration du comportement des agents se fait à partir du modèle BDI et du langage AGENT-0. KQML est utilisé comme langage de communication entre les agents. AgentBuilder est composé d'une interface graphique et d'un langage orienté agent permettant de définir des croyances, des engagements et des actions. Il permet également de définir des ontologies et des protocoles de communications inter-agents. Un agent créé avec cet outil est typiquement un agent d'interface, chargé de faciliter la recherche d'information ou la réalisation de certaines tâches à la place de son utilisateur. Un tel agent sera capable de filtrer l'information, de négocier des services avec d'autres agents et dialoguer avec son utilisateur.

II.4. Conclusion

Dans ce chapitre, nous avons présenté les grandes lignes sur le paradigme du SMA. Nous avons tout d'abord présenté le concept d'agent dans la première partie. Et dans la deuxième partie, nous avons montré les concepts de bases relatives au SMA ; ses définitions, ses caractéristiques, ses

différents types, quelques méthodes de conception, et finalement le chapitre est conclu par quelques plates-formes de développement des SMA utilisées tout au long de notre travail. Donc, ce chapitre a pour vocation de présenter le contexte du domaine SMA. Ce dernier est le paradigme que nous utilisons pour concevoir notre solution.

Dans le chapitre suivant, on va présenter la théorie de décision ou on va montrer les concepts de base relatifs à la décision et au processus de la prise de décision. Plus précisément, le chapitre va renfermer les définitions de base nécessaires pour comprendre la prise de décision et l'outil d'aide à la décision utilisé dans notre travail.

Chapitre III :

*Théorie de la décision et les cartes
cognitives floues*

III.1. Introduction

Chaque jour, nous sommes tous amenés à prendre des décisions. Certaines sont faciles à prendre, d'autres demandent la réflexion et parfois du temps. Dans ce chapitre, nous nous intéressons à la théorie de décision. Cette dernière est un domaine très vaste, nous nous limiterons uniquement à quelques notions de bases qui sont : décision, décideur, prise de décision et outils d'aide à la décision les plus connus, ce qui permet de nous aider à mieux comprendre l'outil utilisé dans notre travail. Par la suite, nous présentons quelques travaux qui combinent les trois paradigmes utilisés dans notre travail.

III.2. Définition générale de la théorie de décision

Selon John Von Neumann et Oskar Morgenstern, la théorie de la décision n'est qu'une combinaison de deux théories qui sont la théorie de la probabilité avec celle de l'utilité. Elle permet de combiner les convictions et les désirs de l'agent, définissant la meilleure action comme celle qui maximise l'utilité attendue [Russell, S., Norving, P., 1995]. L'idée principale de la théorie de la décision est de déterminer à partir de différents types des données le comportement optimal dans une situation donnée à l'aide de modèles mathématiques.

III.3. Notions de base de la théorie de décision

III.3.1. Décision

Plusieurs définitions sont proposées pour le concept de décision :

« Une décision est une action qui est prise pour faire face à une difficulté ou répondre à une modification de l'environnement, c'est-à-dire, pour résoudre un problème qui se pose à l'individu ou à l'organisation » [Lévine, P., Pomerol, J., 1989].

« Une décision est le résultat d'un processus mental qui choisit une parmi plusieurs alternatives mutuellement exclusives » [Holtzman, S., 1989].

Une décision est aussi définie comme « un choix entre plusieurs alternatives », ou encore par « le fait qu'elle concerne aussi le processus de sélection de buts et d'alternatives » [Schneider, D.K., 1994].

Une décision est souvent vue comme le fait d'un individu isolé (le décideur) exerçant un choix entre plusieurs possibilités d'actions à un moment donné [Roy, B., 2000].

La décision peut être définie comme étant l'action de décider après délibération, et que l'acteur exerce un rôle important. Ce n'est donc pas un acte élémentaire et simple, mais c'est plutôt l'aboutissement de tout un processus de décision [Nachet, B., 2014].

On peut déduire que décider c'est sortir de l'incertitude, rompre avec les comportements habituels, résoudre un problème, saisir une opportunité. D'une manière générale, il s'agit de faire un choix parmi plusieurs alternatives dont le but est d'atteindre un objectif.

III.3.2. Décideur

D'après [Souici, I., 2016], on peut dire qu'un décideur :

« Est un organisme ancré dans un environnement : c'est un acteur social. Il doit être décrit et traité comme un système ouvert qui saisit l'information, qui la traite et qui agit ... ».

Généralement, le décideur est la personne physique ou morale qui est appelée à prendre une décision concernant le financement d'un projet par exemple. Le rôle du décideur n'est pas une tâche facile. Ceci est dû au fait que le décideur intervient dans le processus de prise de décision très souvent d'une manière humaine, c'est-à-dire que son raisonnement est influencé par des aspects propres aux humains tels que :

- ✓ Le cadre de décision : la décision est prise en fonction de la manière dont est présenté le problème.
- ✓ Le choix sans raisonnement : souvent, on constate que les gens prennent des décisions sans vraiment savoir pourquoi.
- ✓ L'expérience vécue : Le décideur humain fait toujours une révision rapide dans sa mémoire pour voir s'il a déjà vécu un cas similaire.

Selon la définition précédente, décider c'est agir. Dans le deuxième chapitre, Nous avons montré qu'un agent a la capacité d'agir et selon une situation donnée il est capable de modéliser le comportement d'un décideur.

III.3.3. La prise de décision

III.3.3.1. Définition

Selon [Barthélemy, J.P., 1997 ; Mullet, E. et al., 1996], la prise de décision est classiquement subdivisée en deux tâches de natures différentes : choix d'une (ou plusieurs) alternative(s) parmi un ensemble, et jugement qui traduit une opinion.

III.3.3.2. Processus de la prise de décision

Le processus de décision permet de déterminer les étapes à suivre par un décideur afin d'arriver à opter pour une décision comme solution à un problème posé. Dans [Simon, H.A., 1957], l'auteur propose le modèle IDC décomposant le processus de décision en trois étapes (Intelligence – Design

– Choice) ou en français « Information – Conception – Choix ». Il existe aussi une quatrième phase complémentaire qui intervient pendant la durée de la prise de décision, c'est l'étape (Review) « Evaluation ». Cette dernière peut être vue comme une étape d'évaluation du processus afin de valider ou non la décision à appliquer (Figure III.1.). Depuis, ce modèle reste à ce jour une référence pour la modélisation de la décision [Souici, I., 2016].

- a. Recherche d'information (Intelligence) : c'est une phase pour identifier le problème. Il s'agit de présenter les objectifs ou buts du décideur et de définir le problème à résoudre. Pour cela, il est nécessaire de rechercher les informations pertinentes en fonction des préoccupations du décideur.
- b. Conception et formulation (Design) : c'est une phase de modélisation proprement dite. Le décideur construit des solutions et imagine des scénarios. Cette phase aboutit aux différents chemins possibles à la résolution du problème.
- c. Choix d'un mode d'action (Choice) : permet la sélection d'un mode d'action particulier, c'est-à-dire la prise d'une décision.
- d. Contrôle ou évaluation (Review) : permet d'évaluer la solution choisie (la décision prise). Elle peut amener à un retour arrière vers l'une des trois phases précédentes ou, au contraire, à la validation de la solution.

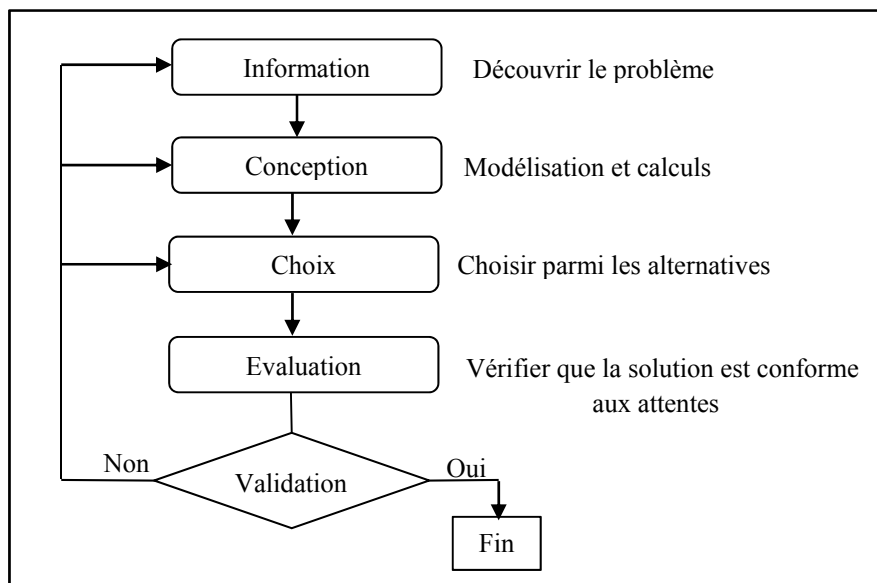


Figure III.1. Modélisation du processus de décision de Simon [Simon, H.A., 1977]

III.3.3.3. Outils d'aide à la prise de décision

Il existe plusieurs outils d'aide à la décision dans la théorie de décision. Généralement, le décideur utilise ces outils afin de faciliter le travail d'analyse du problème. Le rôle principal de ces outils n'est pas de remplacer le décideur mais de compléter, améliorer et aider le décideur afin de réduire son temps et son pourcentage d'erreur.

Dans l'intelligence artificielle, on peut distinguer généralement quelques outils d'aide à la décision définis dans la section suivante:

III.3.3.3.1. Diagrammes d'influence

D'après [Howard, R. et Matheson, J., 1984], un diagramme d'influence (schéma de pertinence, diagramme de décision ou réseau de décision) est une représentation graphique et mathématique compacte d'une situation de décision. A la fin des années 70, le diagramme d'influence est utilisé dans l'analyse de décisions.

Généralement, un diagramme d'influence comme le montre la figure (Figure III.2.) incluant des variables aléatoires A_1, \dots, A_p et des variables de décision D_1, \dots, D_n est un graphe orienté sans circuit $G = (N, E)$ tel que:

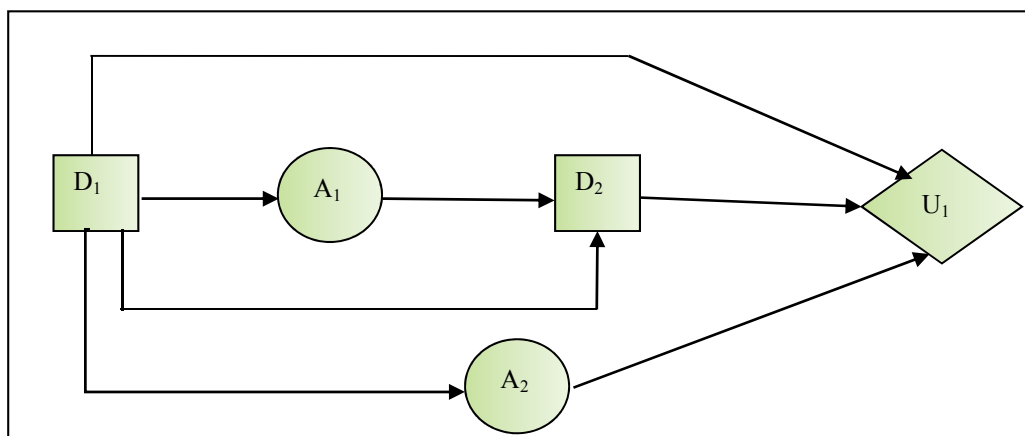


Figure III.2. Diagramme d'influence

- l'ensemble des variables N se partitionne en trois sous-ensembles:
 - $D = D_1, \dots, D_n$: ensemble de variables de décision (représentées par des carrés),
 - $A = A_1, \dots, A_p$: ensemble de variables aléatoires (représentées par des cercles),
 - $U = U_1, \dots, U_m$: ensemble de variables d'utilité (représentés par des losanges);
- l'ensemble E représentant les arcs se partitionne en deux sous-ensembles :
 - un ensemble d'arcs fonctionnels allant d'une variable de décision ou d'une variable aléatoire vers une variable aléatoire ou un variable d'utilité (arcs représentant des dépendances),
 - un ensemble d'arcs informationnels allant d'une variable de décision ou d'une variable aléatoire vers une variable de décision (arcs représentant les variables observées avant de prendre une décision).

Il n'existe pas vraiment de stratégie pour construire les diagrammes d'influence. Etant donné que le but est de structurer une décision qui n'est pas toujours facile, la meilleure façon pour en construire un est de commencer par faire une version simple du diagramme

et par la suite d'ajouter les détails nécessaires jusqu'à ce que le diagramme renferme tous les aspects importants du problème [Avila, F. M., 2002].

III.3.3.3.2. Arbres de décision

Dans les diagrammes d'influence, la représentation des structures de décisions est faite d'une manière propre, mais ils cachent beaucoup de détails. Ce qui est disponible dans un autre type d'outils nommé « arbres de décisions ».

Comme le diagramme d'influence, dans l'arbre de décision, on utilise :

- un rectangle pour représenter une décision,
- un cercle pour représenter un événement aléatoire,
- les branches qui sortent du rectangle représentent les choix que le décideur peut prendre,
- les branches qui sortent d'un cercle représentent les résultats possibles que l'événement aléatoire peut prendre,
- le dernier élément utilisé est le nœud de gain qui est indiqué à la fin de chaque branche.

La figure III.3 présente un simple exemple pour mieux comprendre l'arbre de décision.

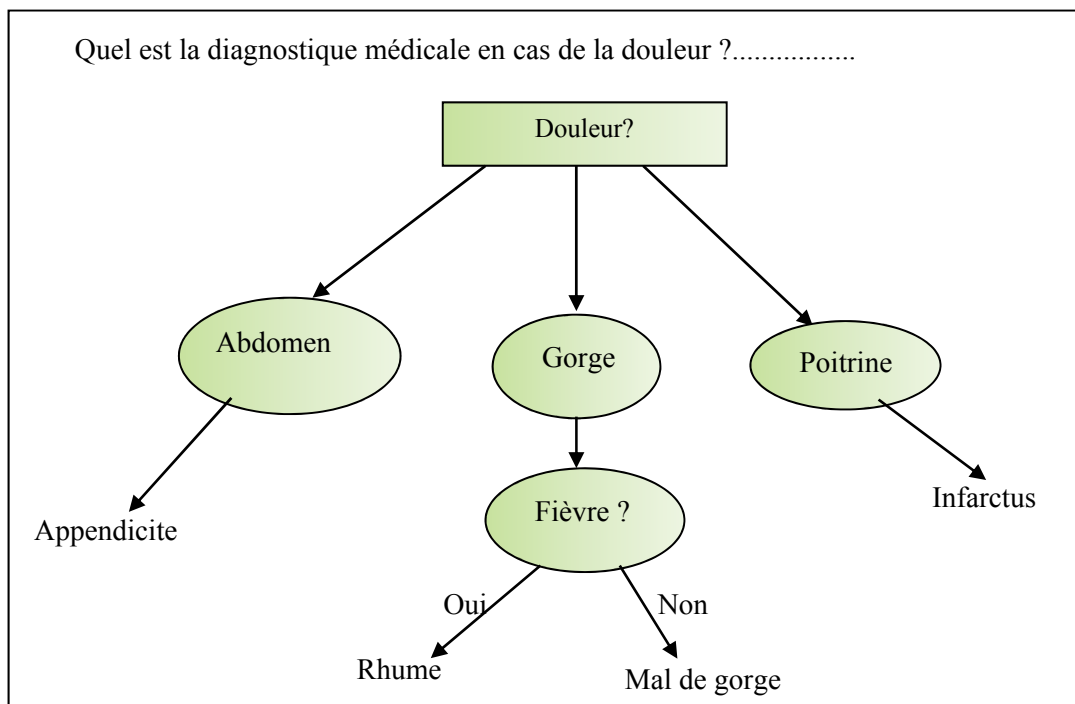


Figure III.3 Exemple d'arbre de décision

Dans un arbre de décisions, on trouve tous les choix possibles qu'un décideur peut faire, ainsi les nœuds sont dans un ordre chronologique (de gauche à droite ou de haut en bas). Généralement, le premier élément d'un arbre est un nœud de décision.

Les diagrammes d'influence et les arbres de décisions sont deux approches qui permettent la modélisation des systèmes de décisions. Les arbres de décision montrent plus d'informations que les diagrammes d'influence. Un inconvénient des arbres de décisions est qu'ils peuvent exploser rapidement d'un point de vue taille. Ces deux approches sont de type quantitatif, et puisque le raisonnement de l'être humain est plus qualitatif que quantitatif, il existe un autre outil proposé dans notre travail qui est nommé carte cognitive. Les sections suivantes vont donner plus de détails sur cette approche. Et dans le chapitre IV, on va motiver l'utilisation des cartes cognitives floues dans notre approche.

III.3.3.3. Cartes cognitives

Le graphe est le mode de représentation utilisé dans la majorité des techniques d'analyse de décisions, tel est le cas dans les arbres de décision et les diagrammes d'influence. Les cartes cognitives, sujet de notre recherche, est une autre approche qui utilise aussi les graphes pour représenter les connaissances mais cette fois-ci d'une manière qualitative. Plus spécifiquement, les nœuds et les arcs des cartes cognitives sont utilisés pour représenter le système de croyances d'une personne, d'un agent ou d'un domaine particulier.

a. Historique

Le terme de Carte Cognitive apparaît pour la première fois en 1948 dans un article de E. Tolman sous le titre *cognitive maps in rats and men* [Tolman, E., 1948] pour décrire la représentation mentale abstraite de l'espace construit par des rats entraînés à se déplacer dans un labyrinthe. Puis dans les années 1970, le sociologue Robert Axelrod [Axelrod, R., 1976], a développé un modèle graphique de connaissance appelé carte cognitive. Il s'agit alors d'un graphe de causalité qui représente un modèle de croyance d'un individu. Axelrod a appliqué ces cartes pour l'aide à la décision dans le domaine de la politique. En 1986, Kosko a introduit la logique floue au modèle d'Axelrod pour obtenir les cartes cognitives floues "CCFs" [Kosko, B., 1986].

b. Définitions

D'après [Laroche, H., Nioche, J.P., 1994], une carte cognitive « consiste à représenter les processus de pensée des décideurs comme orientés par des agencements d'items liés entre eux par des relations ».

Les cartes cognitives (cartes cognitives floues simples ou cartes causales) sont une représentation graphique des croyances d'une personne, d'un agent ou d'un domaine particulier. Le but est de capturer les relations de causalités existantes entre les concepts d'un système donné d'une manière simple [Avila, F.M., 2002].

Une carte cognitive (en anglais Cognitive Maps) est un modèle du monde (ou d'un système) qui se présente sous la forme d'un graphe dont les nœuds représentent les concepts du monde et les arcs les relations de causalité entre ces concepts. Cependant, il ne s'agit pas d'une représentation statique du monde; une carte cognitive est un système « inférenciel » dont lequel des calculs vont être effectués en vue d'évaluer les conséquences d'un état du système (faire des prédictions) : c'est un support d'aide à la décision [Obernesser, C., 2003].

On désigne par K l'un des anneaux \mathbb{Z} ou \mathbb{R} , par δ l'un des nombres 0 ou 1, par V l'un des ensembles $\{0, 1\}$, $\{-1, 0, 1\}$, ou $[-\delta, 1]$. Soient $(n, t_0) \in \mathbb{N}^2$ et $k \in \mathbb{R}^*_{+}$.

Mathématiquement [Papageorgiou, E.I., 2012, Parenthoen M. et al., 2001], une CCF F est un sextuplé (C, A, M, A, fa, R) , où (Figure III.4) :

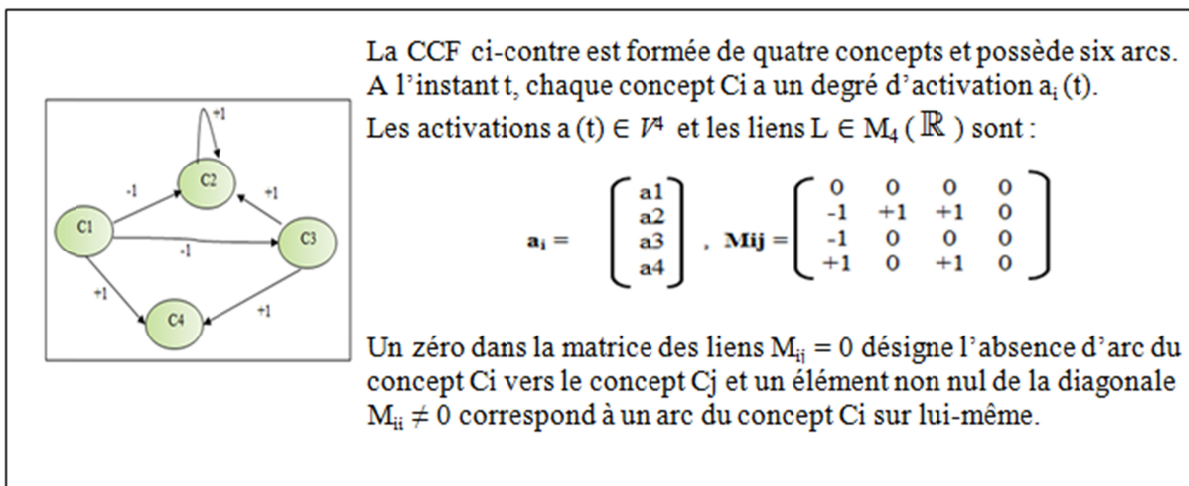


Figure III.4. Exemple de CCF

1. $C = \{C_1, \dots, C_n\}$ est l'ensemble des n concepts formant les nœuds d'un graphe.
2. $A \subset C \times C$ est l'ensemble des arcs $(C_i; C_j)$ orientés de C_i vers C_j .
3. $M : \begin{cases} C \times C \rightarrow K \\ (C_i, C_j) \mapsto M_{ij} \end{cases}$ est une fonction de $C \times C$ vers K associant M_{ij} à un couple de concepts (C_i, C_j) , avec $M_{ij} = 0$ si $(C_i, C_j) \notin A$, ou avec M_{ij} égale le poids de l'arc orienté de C_i vers C_j si $(C_i, C_j) \in A$. Remarquons que : $M(C \times C) = (M_{ij}) \in K^{n \times n}$ est une matrice de la CCF F .
4. $A : \begin{cases} C \rightarrow V^{\mathbb{N}} \\ C_i \mapsto a_i \end{cases}$ est une fonction qui à chaque concept C_i associe la suite de ses degrés d'activation telle que pour $t \in \mathbb{N}$, $a_i(t) \in V$ soit son degré d'activation à l'instant t . On notera $a(t) = [(a_i(t))_{i \in [1, n]}]^T$ le vecteur des activations à l'instant t .
5. $fa \in (\mathbb{R}^n)^{\mathbb{N}}$ une suite de vecteurs d'activations forcées tels que pour $i \in [1, n]$ et $t \geq t_0$, $fa_i(t)$ soit l'activation forcée du concept C_i à l'instant t .

6. R est une relation de récurrence sur $t \geq t_0$ entre $a_i(t+1)$, $a_i(t)$ et $fa_i(t)$ pour $i \in [1, n]$ traduisant la dynamique de la CCF F . R est expliqué dans la figure suivante (Figure III.5) [Papageorgiou, 2012, Parenthoen et al., 2001].

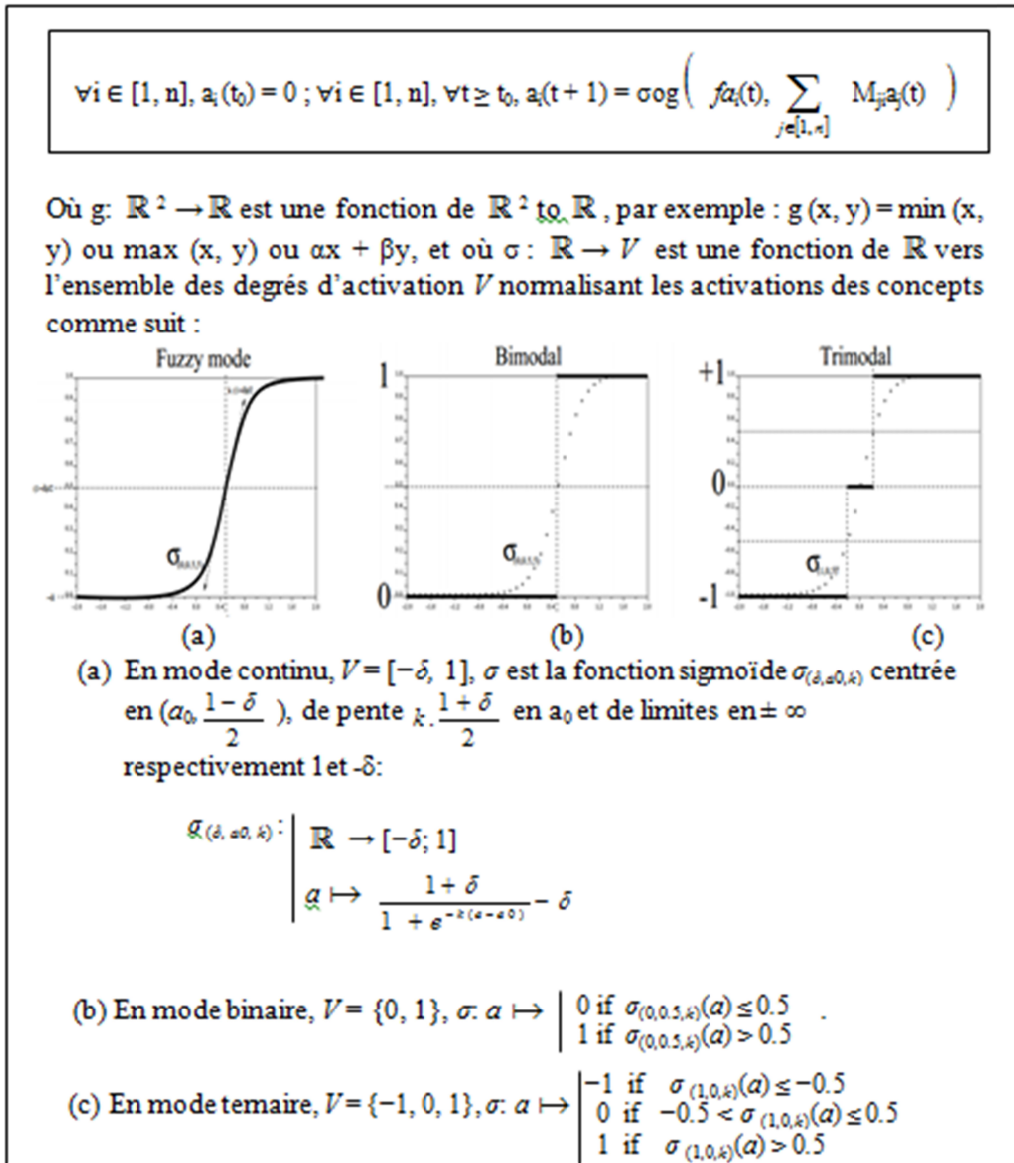


Figure III.5. Différentes fonctions de normalisation σ d'une CCF

Nous avons appliqué la relation de récurrence (fonction de seuil) de deuxième mode (mode binaire) dans le processus d'analyse et d'inférence tout au long de notre travail. Selon les besoins de notre approche, les informations les plus intéressantes pour notre travail sont les suivantes :

- ✓ Dans une CCF simple, les poids peuvent prendre les valeurs $\{-1, 0, 1\}$ et les concepts les valeurs 0 = inactif ou 1 = actif.
- ✓ Chaque configuration particulière des activités des concepts représente un état du système.
- ✓ L'état du système sera lui représenté par une matrice de taille $1 \times n$ (un vecteur).
- ✓ la dynamique est calculée mathématiquement par le produit matriciel normalisé.

- ✓ Chaque arc entre deux concepts de C_i vers C_j est associé à un poids M_{ij} traduisant une relation d'inhibition ($M_{ij} < 0$) ou d'excitation ($M_{ij} > 0$) du concept C_i vers le concept C_j .
- ✓ Le vecteur des activations évolue dans le temps selon les liens existants entre les concepts.
- ✓ La dynamique d'une CCF se résume pour un cycle (passage de t à $t + 1$) en une mise à jour du vecteur des activations comme étant la normalisation du couple d'activations ($f_a(t)$, $M^T \cdot a(t)$).
- ✓ Une CCF aide à prédire l'évolution du système (simulation de comportement), peut être utilisée pour trouver des points fixes ou des cycles limites dans les états futurs.
- ✓ Dans notre cas des cartes cognitives floues (CCFs) [Miao, Y. et al., 2001], les poids des connexions peuvent prendre toutes les valeurs dans l'intervalle $[-1 \dots 1]$ ou $[0 \dots 1]$, ce qui permet d'exprimer de façon plus précise et plus réaliste la force de la relation de causalité entre deux concepts. Les valeurs d'activités des concepts peuvent prendre comme valeur 0 ou 1 dénotant ainsi pour un concept-cause la "force" de la cause, et pour un concept-effet la "force" de l'effet.

c. L'algèbre causale

Le raisonnement des cartes cognitives nécessite l'algèbre causale. Ce dernier permet d'établir les opérations mathématiques de base largement utilisées depuis plusieurs siècles. Dans ce qui suit on va présenter ces opérations, dont le but de voir la façon dont nous pouvons les utiliser dans notre contexte.

Tout d'abord, on va commencer par présenter toutes les valeurs qu'une relation causale peut prendre.

❖ Valeurs des relations causales

Nous avons déjà vu qu'il existe trois types de relations causales fondamentales : $\{+, -, 0\}$. Mais pour avoir un ensemble plus complet de relations causales nous présentons dans le tableau suivant l'ensemble des huit valeurs résultantes de toutes les combinaisons logiques possibles du positif, négatif et zéro [Axelrod, R., 1976]: $\{+, -, 0, N_n, N_p, \pm, ?, a\}$.

| <i>Relation causale</i> | <i>Signification</i> |
|-------------------------|---------------------------|
| + | Positif |
| - | Négatif |
| 0 | Zéro |
| Nn | non négatif, {0, +} |
| Np | non positif, {0, -} |
| ± | non zéro, {+, -} |
| ? | universel, {+, -, 0} |
| A | ambivalent, ensemble vide |

Tableau III.1. Valeurs possibles d'une relation causale

Pour manipuler ces huit valeurs, il convient de savoir qu'ils existent quatre opérations de base : l'union (\cup), l'intersection (\cap), l'addition (\oplus), et la multiplication (\otimes). Les sections suivantes sont consacrées pour expliquer l'utilisation de ces quatre opérations causales, ainsi que les règles correspondantes à chacune d'elles.

❖ *Operations*

Dans ce qui suit, nous dénotons que le terme « y » signifie que y peut prendre n'importe quelle valeur.

➤ *Opération entre deux cartes cognitives*

Généralement, les opérations entre deux cartes causales sont nécessaires lorsqu'une personne a fait plusieurs assertions sur un même sujet dans des documents différents, ou lorsque nous désirons unir l'information contenue dans différentes cartes cognitives faites par différents auteurs sur un même domaine [Avila, F.M., 2002].

a) *Intersection*

Dont le but d'avoir une seule carte, l'opération d'intersection est utilisée dans des situations où ils existent deux cartes causales ou plus du même auteur sur un domaine particulier. Ce cas est essentiellement utilisé pour réviser la cohérence des assertions de l'auteur, c'est pour cela que toutes les relations qui se contredisent vont être identifiées par le signe d'ambivalence « a ». Le tableau suivant montre les règles pour l'opération d'intersection :

| N° | Règle |
|-----|---------------------------|
| (1) | $0 \cap + = a$ |
| (2) | $0 \cap - = a$ |
| (3) | $+ \cap - = a$ |
| (4) | $y \cap y = y$ |
| (5) | $y \cap a = a \cap y = a$ |

Tableau III.2. Règles d'opération d'intersection

Dans ce qui suit, on va donner un exemple de deux cartes cognitives présentées dans la figure III.6, dont lesquelles représentent les relations causales d'un auteur sur un même sujet. Nous voulons vérifier si ses idées sont cohérentes par rapport à ses deux articles.

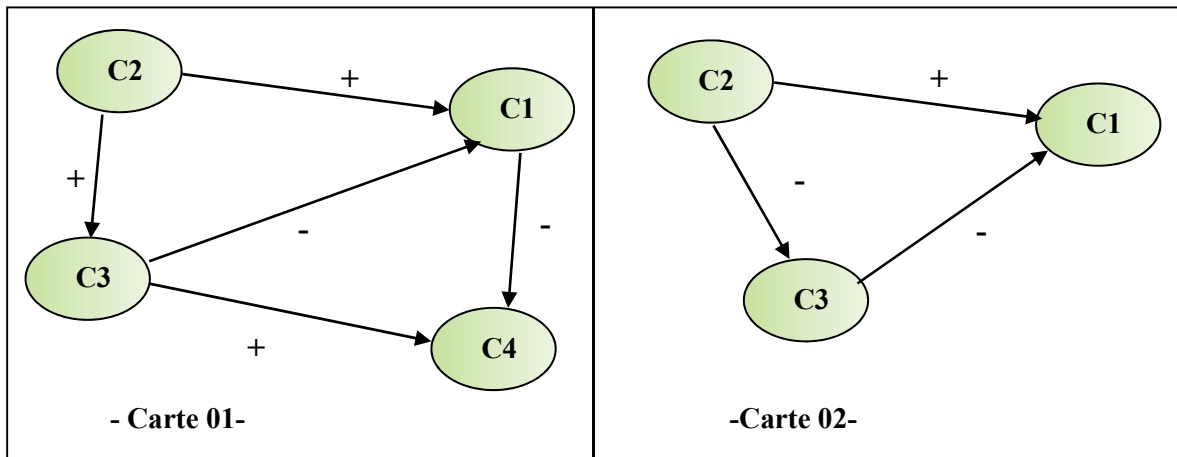


Figure III.6. Exemple de deux cartes cognitives de même auteur

Généralement, si on a un ensemble des cartes cognitives de même auteur alors l'opération d'intersection est exécutée sur les cartes cognitives deux à deux. Autrement dit ; par exemple si on a trois cartes ; on doit en premier temps sélectionner deux des trois cartes cognitives et faire leur intersection, ensuite faire une deuxième intersection avec la carte résultante de la première intersection et la carte restante, (toujours deux cartes à la fois [Kosko, B., 1992]).

Le résultat d'intersection est présenté dans la figure III.7. L'intersection comprend seulement les concepts et les relations qui se trouvent dans les deux cartes.

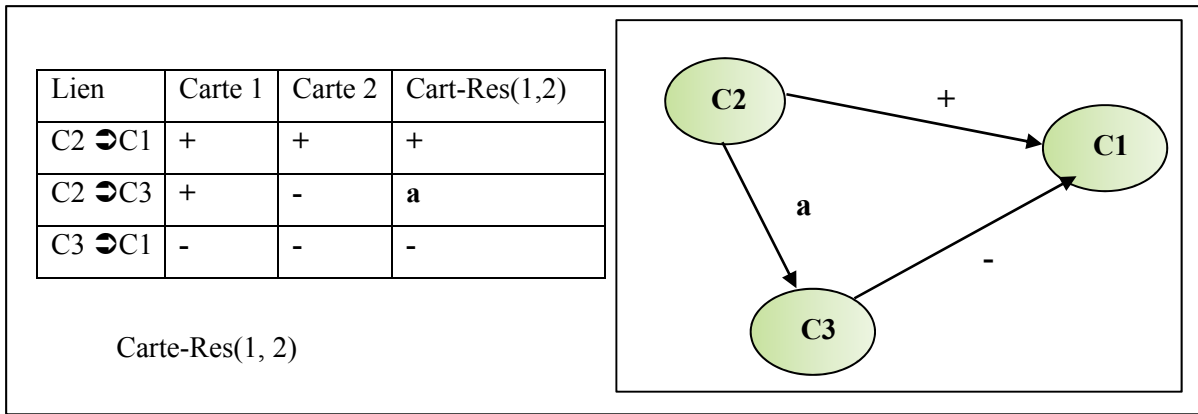


Figure III.7. Intersection de deux cartes : $Carte1 \cap Carte2 = Carte-Res(1, 2)$.

b) Union

L'opération de l'union est utilisée lorsqu'il est nécessaire de rassembler dans une seule carte l'information de plusieurs cartes cognitives effectuées par différents auteurs sur un même domaine [Avila, F.M., 2002]. Cependant, il existe une condition importante à respecter « avoir au moins une relation causale en commun (indépendamment du signe) afin d'établir le lien entre les deux cartes cognitives ». Le tableau suivant montre les règles pour l'opération d'union :

| N° | Règle |
|-----|-----------------------|
| (1) | $0 \cup + = Nn$ |
| (2) | $0 \cup - = Np$ |
| (3) | $+ \cup - = \pm$ |
| (4) | $y \cup y = y$ |
| (5) | $0 \cup + \cup - = ?$ |

Tableau III.3. Règles d'opération d'union

La procédure utilisée pour faire l'union entre deux cartes est la même que celle utilisée pour l'intersection, elle doit être appliquée en groupe de deux cartes cognitives à la fois.

➤ **Opérations à l'intérieur de la carte cognitive**

Une fois la carte causale représentée en forme graphique nous pouvons commencer à l'analyser. Pour connaître l'influence qu'exerce un concept sur un autre, même si les concepts ne sont pas directement reliés, nous devons utiliser les opérations de multiplication et d'addition. Grâce à ses opérations nous pouvons calculer les effets indirects d'un chemin et les effets totaux de plusieurs chemins [Axelrod, R., 1976].

a) Multiplication

L'opération de la multiplication dans les cartes cognitives est utilisée pour calculer les effets indirects. Par exemple, si un chemin est formé par plusieurs concepts, alors il existe un effet indirect du premier au dernier concept. Voici un exemple (en appliquant la règle numéro (1) de la multiplication) dans la figure (figure III.8) suivante :

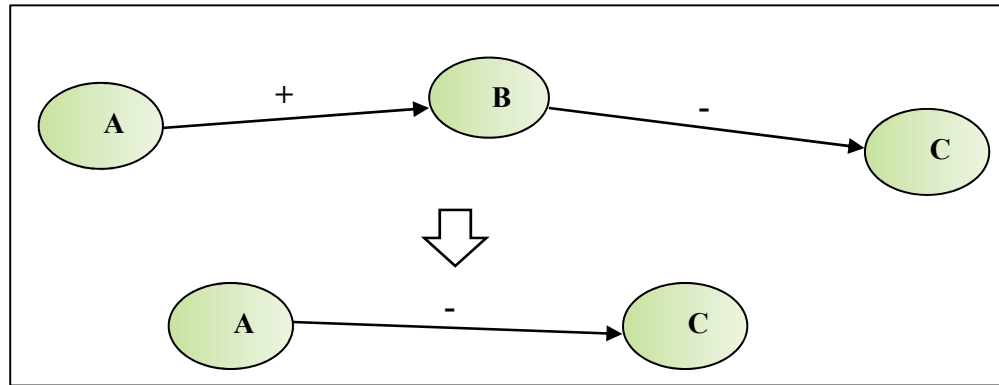


Figure III.8. Exemple de l'effet direct

Le tableau suivant montre les six règles de la multiplication :

| N° | Règle |
|-----|--|
| (1) | $+ * y = y$ |
| (2) | $0 * y = y$ |
| (3) | $a * y = a$ |
| (4) | $- * - = +$ |
| (5) | * distribue par rapport à \cup et à \cap |
| (6) | * est commutative : $x * y = y * x$ |

Tableau III.4. Règles de la multiplication

b) Addition

L'objectif de l'addition dans les cartes cognitives vise en fait à déterminer l'effet total existant d'un concept-cause C_i sur un concept-effet C_j à travers tous les chemins qui conduisent de C_i à C_j . Quand une carte cognitive possède plusieurs chemins pour arriver d'un concept C_i à un concept C_j , ses valeurs peuvent être additionnées pour obtenir l'effet total du concept C_i sur le concept C_j . Les règles de l'addition peuvent être appliquées une fois que tous les effets indirects des différents chemins qui conduisent de C_i à C_j ont été déterminés.

Les six règles de l'addition sont présentées dans le tableau suivant :

| N° | Règle |
|-----|--|
| (1) | $0 y = y$ |
| (2) | $a y = a$ |
| (3) | $y y = y$ |
| (4) | $+ - = ?$ |
| (5) | $ $ distribue par rapport à \cup et à \cap |
| (6) | $ $ est commutative : $x y = y x$ |

Tableau III.5. Règles de l'addition

Dans la quatrième règle, le résultat peut avoir la valeur indéterminée (?). C'est un des inconvénients majeur d'appliquer l'opération d'addition dans les cartes cognitives. Plus clairement, s'il existe plusieurs liens causaux qui partent d'un concept C_i et qui finissent dans le même concept C_j mais avec des valeurs différentes, alors il est impossible de déterminer quelle sera la nouvelle valeur attribuée au lien qui résumera l'effet total des liens causaux du concept C_i au concept C_j vu qu'il existe des contradictions. Ce problème n'est pas résolu dans les cartes cognitives simples. Pour cela, des approches comme les cartes cognitives floues ou celles qui tiennent compte de la longueur des chemins ou de l'importance des chemins ont réglé ce problème.

Dans l'exemple de la figure III.9, nous effectuons une petite comparaison du résultat obtenu d'un effet total du concept **A** au concept **B** avec des valeurs causales et avec des valeurs floues.

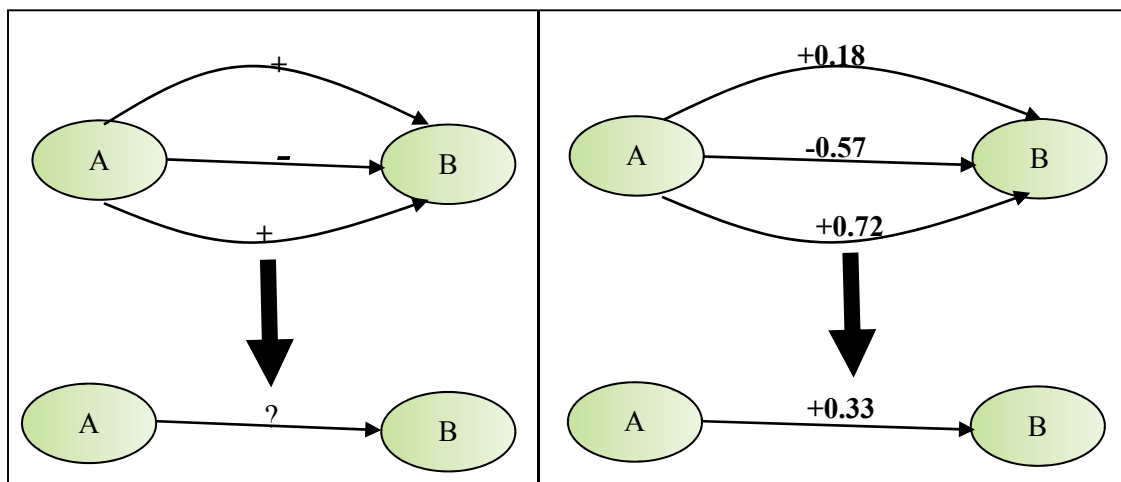


Figure III.9. Exemple d'addition

d. Processus de construction d'une carte cognitive

Dans les sections précédentes, on a vu quelques notions de base relatives à la carte cognitive. Maintenant on doit voir les différentes techniques possibles pour sa construction. Selon Robert Taber [Taber, R., 1991], il existe cinq phases pour l'obtention d'une carte cognitive qui sont les suivantes :

1. la sélection de la source d'information,
2. l'acquisition de la carte,
3. la conversion de la carte en forme de matrice,
4. l'inférence
5. la construction de la base de connaissances globale.

Ces principales phases seront traitées dans les sections suivantes selon l'ordre proposé par Taber.

d.1 : La sélection de la source d'informations

D'après les méthodes proposées par Axelrod dans la deuxième phase « L'acquisition de la carte cognitive », soit on va :

- a) Dériver une carte à partir d'un document.
- b) Dériver une carte à l'aide d'un groupe d'experts (par questionnaire).

Donc on a deux sources d'information : les documents et les experts.

Selon Axelrod [Axelrod, R., 1976], la dérivation d'une carte cognitive à partir d'un document est une méthode très coûteuse en terme de temps puisqu'elle exige la codification d'un texte donné, phrase par phrase et elle nécessite un codeur assez compétent. Donc dans cette méthode, le principal déterminant est la longueur du document. Pour la deuxième source qui est les experts, le facteur majeur est le temps des répondants (experts). Le nombre de questions autour des liens causaux est $2^n - n$, où n représente le nombre de concepts. Dans cette méthode, l'utilisation d'un panneau d'experts pour créer une carte cognitive est un bon choix. Dans ce cas, la carte cognitive est généralement plus précise qu'une carte construite par un seul expert. Tout au long de notre travail, nous avons appliqué deuxième méthode.

d.2 : L'acquisition de la carte cognitive

d.2.1 : L'acquisition à partir d'un document

Une fois le bon document est choisi, il faut suivre les étapes de ce processus :

- 1) Produire deux listes, une pour les concepts et autre pour les assertions causales, où ces deux listes sont extraites à partir du document et reflètent les concepts et les relations de cause à effet.
- 2) Placer les concepts pour construire la carte.
- 3) Relier les concepts par les relations causales.

d.2.2 : L'acquisition à l'aide d'un groupe d'experts

Dans ce cas, pour construire la carte cognitive il faut suivre les étapes suivantes [Axelrod, R., 1976] :

- 1) Identifier les concepts pertinents, où on peut demander aux experts d'écrire une liste de tous les concepts pertinents pour le domaine à analyser.
- 2) Limiter le nombre de concepts en évaluant leur importance.
- 3) Choisir les arcs et ses poids (les relations causales).

Cette étape se fait comme suit :

- ❖ Distribuer des questionnaires au panneau d'expert avec toutes les combinaisons possibles des concepts précédemment sélectionnés.
 - ❖ Les experts doivent assigner un signe parmi $\{-, 0, +\}$ pour chaque paire de concepts différents.
 - ❖ Rassembler les cartes de chaque expert et les intégrer dans une seule carte cognitive.
- 4) Retourner la carte résultante aux experts pour des discussions et modifications. Les désaccords devraient être identifiés et analysés. La carte est généralement modifiée plusieurs fois avant que les experts soient satisfaits du résultat final.

d.3 : La conversion de la carte en forme de matrice

La forme graphique de la carte cognitive permet de faciliter sa conception et sa compréhension. Pour exploiter la carte cognitive il faut la transformer sous une forme matricielle, où tous les mécanismes d'inférence se basent sur la représentation matricielle. Donc avant de passer à la phase suivante « Inférence », il faut extraire la matrice d'adjacence de la carte cognitive.

d.4 : L'inférence

Une carte cognitive va permettre – par une série d'opérations algébriques – d'évaluer les conséquences d'un état du système qu'elle représente; ces conséquences se manifestent sous la forme d'un autre état [Avila, F.M., 2002]. Une CCF doit être vue comme un système dynamique qui – à partir d'un état initial – va donner de nouveaux états descriptifs du système à chaque pas de temps. Ces opérations représentent le processus d'inférence, qui donne des réponses aux questions de type "*qu'arrivera t'il si ?*" (*What if*) au système.

Voici un exemple (**Figure III.10**) concret d'une carte causale pour montrer et expliquer au même temps le processus d'inférence, qui permet de représenter la fuite d'un agent en fonction de la distance qui le sépare de son ennemi.

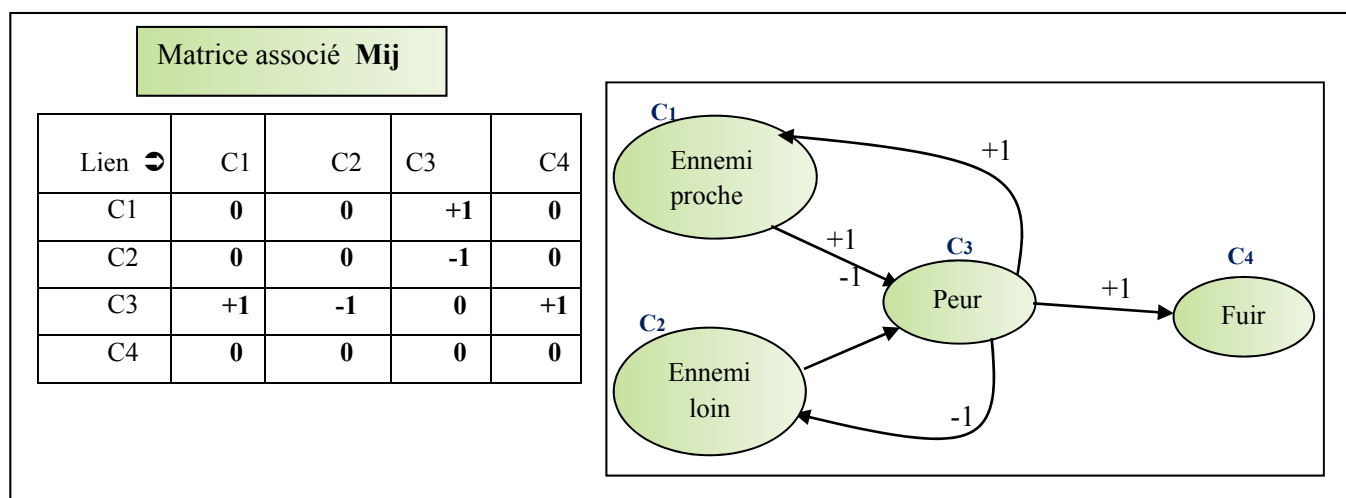


Figure III.10. Exemple de carte cognitive [Parenthoen M. et al., 2001]

Selon [Chauvin, L., 2010], on peut résumer le processus d'inférence par l'algorithme suivant:

- Supposons qu'on a une carte cognitive de **n** concepts (dans notre exemple **n=4**)

1. Extraire la matrice d'adjacence **M_{ij}**.
2. Préparer le vecteur exciteur **a₁**.
 - a. Initialiser le vecteur exciteur à zéro (0).
/ Chaque valeur *i* représente l'état de concept *i*/
 - b. Mettre à **1** les concepts voulus. C'est-à-dire les concepts qu'on veut voir ; qu'arrivera-t-il s'ils sont activés.
 - c. Ajouter **a₁** à la mémoire de travail MT (cette mémoire contient tous les vecteurs-résultants du processus d'inférence).
3. Faire le produit matriciel **a_{i+1} = a_i × M_{ij}** / *i=1,2...*
4. **Si** **a_{i+1}** appartient à la mémoire de travail alors on a obtenu un comportement typique.

Sinon

- a. Remettre à **1** les concepts voulus du départ pour le vecteur **a_{i+1}**.
- b. Mettre à « **0** » où comme Taber [Taber, R., 1991] et Perusich [Perusich, K., 1996] « **-1** » aux valeurs négatives pour le vecteur **a_{i+1}**.
- c. Ajouter **a_{i+1}** à la mémoire de travail.
- d. Incrémenter *i* et aller à 3.

Le résumé de ce processus est comme suit:

$\mathbf{a}_{i+1} = \mathbf{S} [\mathbf{a}_i \times \mathbf{M}_{ij}] / i=1 \dots \text{Imax}$ où \mathbf{S} est la fonction de seuil (la fonction de seuil est représentée par les deux premier point après le « **sinon** » dans le l'algorithme ci-dessus). Soit on laisse l'algorithme se stabiliser à soi-même ou on le fixe par un nombre d'itérations maximal Imax .

Cet algorithme donne comme sortie un vecteur-résultant de dimension $1 \times n$ qui contient la liste des réponses. Chaque élément i de ce vecteur qui est différent à 0 indique que le concept i va avoir une augmentation si les concepts voulus du départ ont une augmentation et vice-versa.

Avec notre exemple, si on veut connaitre « qu'arrivera-t-il si le concept *Ennemi proche* est activé », on fait l'inférence comme suit :

Vecteur exciteur $\mathbf{a}_1 = (1, 0, 0, 0)$.

1- $\mathbf{a}_1 \times \mathbf{M}_{ij} = (0, 0, 1, 0) \Rightarrow$ Fonction de seuil $\Rightarrow (1, 0, 1, 0) = \mathbf{a}_2$

2- $\mathbf{a}_2 \times \mathbf{M}_{ij} = (1, -1, 1, 1) \Rightarrow$ Fonction de seuil $\Rightarrow (1, 0, 1, 1) = \mathbf{a}_3$

3- $\mathbf{a}_3 \times \mathbf{M}_{ij} = (1, -1, 2, 1) \Rightarrow$ Fonction de seuil $\Rightarrow (1, 0, 1, 1) = \mathbf{a}_4 = \mathbf{a}_3$

On peut interpréter le résultat comme suit : si le concept « *Ennemi proche* » est activé, alors les concepts « *Peur* » et « *Fuir* » auraient une augmentation. Autrement dit, plus l'ennemi est proche de l'agent, plus sa peur s'accroît et plus il a la sensation de fuir.

d.5 : La construction de la base de connaissance globale

Enfin, on peut construire notre base de connaissance globale par l'inférence sur tous les scénarios possibles, c'est-à-dire, voir qu'arrivera-t-il si n'importe quelle combinaison possible de concepts est activée.

III.4. Travaux similaires

L'approche proposée est situé dans l'intersection des grands domaines de recherche qui sont les SIC, SMA, CCF et les systèmes holoniques qui sont présentés dans les trois premiers chapitres. Dans ce qui suit, nous mentionnons quelques travaux dans quelques combinaisons de ces domaines.

III.4.1. SIC et SMA

Plusieurs travaux ont combiné ces deux paradigmes (SIC et SMA), par exemple, M. Séguran a proposé l'utilisation du SMA dans le SIC pour résoudre les conflits sémantiques à partir de bases de données hétérogènes [Séguran, M., 2003]. N. Zarour et al ont proposé de soutenir les systèmes d'entreprise virtuelles à l'aide de coordination d'agents [Zarour, N. et al., 2005]. M.A. Kamoun et al ont proposé un SIC basé sur les agents pour les itinéraires multimodaux [Kamoun, M.A. et al.,

2005]. Presque dans le même contexte, L. Ouahrani et ses collègues ont suggéré un modèle basé sur un agent pour un accès uniforme aux ressources dans les CIS [Ouahrani, L., Alimazighi, Z., 2009]. K. Zarour et al ont proposé de combiner les SIC et les SMA dans le domaine médicale [Zarour, K. et al., 2010]. J.L.R. Sousa et ses collègues ont proposé un modèle de socio-matérialité des systèmes d'information et ont présenté leur impact sur le développement organisationnel coopératif de l'environnement [Sousa, J.L.R. et al., 2015]. B. Nachet et A. Adla [Nachet, B., Adla, A., 2014] ont proposé un modèle multi-agent pour la conception du système de prise de décision collective. Le point de comparaison est situé dans l'étape de décision pour résoudre le problème posé. Le détail de la comparaison est présenté dans le chapitre V. L. Munan et H. Fen ont présenté une description formelle des organisations dans le système d'information de soins de santé omniprésente, c'est une perspective du système multi-agent [Munan, L., Fen, H., 2016]. Enfin, dans [Dotoli, M. et al., 2017] l'auteur a proposé un système d'information avancé de voyageur multi-agent pour une planification de voyage optimale dans un cadre co-modal.

III.4.2. CCF et SMA

Les cartes cognitives ont été utilisées dans plusieurs domaines, on peut citer par exemple en 1988, Styblinski et Meyer utilisent les CCFs pour analyser des circuits électriques [Styblinski, M.A., Meyer, B. D., 1988]. Dans [Dickerson, J. A., Kosko, B., 1986; Kosko, B., 1992] (l'origine de l'utilisation des CCFs), les auteurs ont démontré une nouvelle approche: l'utilisation des CCFs pour la modélisation des acteurs dans un monde virtuel. C. Stylios et P. Groumpos [Stylios, C., Groumpos, P., 1998] ont utilisé les CCFs dans le défi de la modélisation des systèmes de supervision. La contribution de [Jung, J., Geun-Sik, J., 2000] suggère l'utilisation de la CCF dans le contexte du commerce. Plus proche de notre travail, F. M. Avila [Avila, F.M., 2002] a proposé l'utilisation de cartes causales dans un environnement multi-agent comme une alternative pour représenter la connaissance de manière qualitative et pour gérer la prise de décision d'un agent. G. Acampora et V. Loia ont introduit une nouvelle méthodologie de conception de systèmes AmI qui exploite le paradigme multi-agent et une nouvelle extension de la théorie des CCFs bénéficiant de la théorie des automates chronométrés. Cette méthodologie permet de créer une collection d'agents dynamiques intelligents qui utilisent l'informatique cognitive pour définir des modèles d'actions capables de maximiser les paramètres environnementaux comme, par exemple, le confort de l'utilisateur ou l'économie d'énergie [Acampora, G., Loia, V., 2009]. V. Rodin et al ont développé un graphique d'influence floue pour modéliser le comportement cellulaire en biologie des systèmes par la voie biochimique intracellulaire [Rodin, V. et al., 2009]. M. Stula et al ont développé une carte cognitive floue (ABFCM) pour l'injection du concept de système multi-agent (SMA) dans la CCF et les différents algorithmes d'inférence dans chaque nœud permettant la simulation de systèmes à divers concepts comportementaux [Stula, M. et al., 2010]. Aussi, A.M. Gadallah et al

ont proposé un SMA basé sur une approche dynamique de la CCF [Gadallah , A.M. et al., 2010]. N. Lee et ses collègues ont proposé un système de cartographie cognitive basé sur un agent pour l'analyse des opportunités de vente [Lee, N. et al., 2011]. Presque dans le même contexte, K.C. Lee et al ont utilisé une approche CCF basée sur les agents pour la planification stratégique du marketing pour les entreprises industrielles [Lee, K.C. et al., 2013]. S. Nalchigar et al ont proposé la simulation du processus de planification stratégique des systèmes d'information à l'aide d'une CCF [Nalchigar, S. et al., 2011]. S. Mei et al ont fourni une étude de CCF dans la prise de décision individuelle [Mei, S. et al., 2014]. N. Tomá a décrit une nouvelle approche à la modélisation du modèle de vie artificielle à base individuelle basée sur les CCFs. Le concept proposé se concentre sur l'optimisation de l'intelligence artificielle des individus dans les modèles multi-agents et leur adaptation à l'environnement [Tomá, N., 2015]. Pour reconstruire les réseaux de régulation génique (RRG) à grande échelle avec une grande précision, les auteurs dans [Jing, L. et al., 2016] ont proposé un algorithme génétique multi agent dynamique (AGMA_d) à partir de profils d'expression temporels basés sur les CCFs. Enfin, P.J. Giabbanelli et ses collègues ont proposé de combiner les CCFs avec la modélisation à base d'agents, dont le but de montrer les cadres et les pièges d'une approche de modélisation hybride puissant pour comprendre les interactions entre l'environnement et l'humain [Giabbanelli, P.J. et al., 2017].

Nous avons essayé de répondre aux diverses insuffisances des propositions mentionnées ci-dessus et donc d'améliorer le processus décisionnel en particulier dans un contexte coopératif, en combinant les domaines de SIC, SMA, CCF, systèmes holoniques, et le concept d'ontologie. Les avantages de cette association sont présentés dans le chapitre suivant.

III.5. Conclusion

Tout au long de ce chapitre, nous avons présenté une vue globale sur la théorie de décision, où on a montré les concepts de base relatifs à la décision et au processus de la prise de décision. Plus précisément, le chapitre a renfermé les définitions de base nécessaires pour comprendre l'outil d'aide à la décision utilisé dans notre contexte, qui est nommé carte cognitive floue (CCF). Nous avons présenté son historique, ses définitions, ensuite on a étudié l'algèbre causale qui permet la manipulation des cartes cognitives ainsi que les quatre opérations qui permettent d'agrandir, de simplifier et de réviser les différentes cartes. Enfin, on a décrit le processus pour sa construction. Et dans la dernière section nous avons présenté quelques travaux voisins.

Le chapitre suivant est consacré pour faire l'intégration des trois premiers chapitres, pour arriver à présenter le cœur du sujet de cette thèse. Autrement dit, présenter l'approche proposée pour notre travail.

Chapitre IV :

***DCSic-CFSH: l'aide à la Décision
Collective dans les SIC basée sur les Cartes
Cognitives Floues et les Systèmes Multi-
Agents Holoniques***

IV.1. Introduction

Après avoir présenté quelques généralités de base sur les SICs dans le premier chapitre, avoir défini les agents et les systèmes multi-agents dans le deuxième chapitre et avoir décrit les CCFs dans le troisième chapitre, il est temps d'expliquer comment profiter d'utiliser de telles cartes causales plus précisément de type flou et de tels SMA de type holonique dans l'aide à la décision collective d'un SIC.

À l'ère de la compétitivité à l'échelle mondiale, l'exécution du processus décisionnel est considérée dans les SICs comme un mécanisme important permettant de répondre aux différentes exigences du marché. Donc, pour assurer la cohérence des différentes décisions, et assurer que la décision finale est celle optimale pour résoudre un problème distribué global, notre approche consiste à développer un mécanisme efficace qui répond à ces besoins. Dans un premier temps, nous citons les principaux concepts utilisés dans notre travail dans les trois premières sections, puis nous décrivons en détail l'approche proposée.

IV.2. Intégration des CCFs dans les SICs

Les systèmes complexes sont de plus en plus considérés comme une discipline distincte utilisée dans de nombreux domaines scientifiques tels que la physique, l'économie, la biologie, la chimie, la sociologie et l'informatique. Notre contexte de travail est le SIC, qui est un système complexe supportant le travail collaboratif entre les différents types d'entités qui peuvent être des individus, des services ou des entreprises.

Le problème de la prise de décision n'est pas évident dans un seul SI; il devient de plus en plus complexe dans les SICs. Parmi les problèmes nécessitant une prise de décision rapide, on trouve le cas des applications militaires, des urgences médicales, du diagnostic des installations industrielles, de la météorologie, de la sécurité routière, etc.

Pour assurer la cohérence des décisions et améliorer le processus de la prise de décisions dans un SIC, plusieurs systèmes d'aide à la décision sont en charge. Il existe deux types de systèmes, quantitatifs et qualitatifs. Puisque le raisonnement de l'être humain est plus qualitatif que quantitatif, nous proposons d'utiliser un système de type qualitatif appelé carte cognitive floue (CCF). Les CCFs ont prouvé leurs performances dans de nombreux domaines tels que les systèmes de contrôle, la politique, la supervision et la gestion des entreprises.

- ✓ Une carte cognitive est un mécanisme de collecte d'informations sur l'évolution des émotions suite à un événement,
- ✓ aussi bien qu'elle est facile à construire et à utiliser,
- ✓ flexible dans la performance,
- ✓ transparent aux non-spécialistes,
- ✓ traite des questions complexes liées à la gestion des connaissances, etc.

Nous utilisons les CCFs où les poids accompagnant les arcs prennent la valeur dans l'intervalle $[-1, 1]$ décrivant le type de causalité. Par exemple, s'il existe deux concepts liés au domaine de la sécurité routière (C_1 = nombre de patrouilles policière, C_2 = accident mortel), nous pouvons décrire la relation avec la CCF comme suit: si le nombre de patrouilles policière augmente, alors il y a une diminution du nombre d'accidents mortels avec 70%, au lieu de 100% (plus proche de la réalité). En outre, la CCF élimine les cas d'indétermination et d'ambivalence qui existent dans l'ancien type (les cartes cognitives simples).

Ainsi, dans notre travail nous proposons d'utiliser les CCFs dans les SICs. Cette intégration nous fournit les avantages suivants:

- ✓ Assurer la cohérence des décisions partielles et améliorer le processus de la prise de décisions ;
- ✓ La causalité qui existe dans les CCFs permet d'expliquer les décisions passées;
- ✓ Prédire les états futurs du SIC;
- ✓ Dans un SIC, les CCFs peuvent être utilisées comme un outil d'aide à la prise de décisions ;
- ✓ Elles permettent de faciliter le raisonnement sur les vues partielles (décisions) des différents systèmes composants du SIC;
- ✓ Elles permettent d'améliorer la base de connaissances des systèmes composants ;
- ✓ La fusion des différentes CCFs permet également le renforcement des relations causales en vue d'avoir une décision collective optimale.

IV.3. Modélisation d'un SIC à base d'agents holoniques

Dans un SIC, chaque SI a les caractéristiques suivantes: agir seul pour prendre des décisions, communiquer, être proactif, négocier, etc (voir chapitre I section 3.2.). Ces différentes caractéristiques peuvent être trouvées dans le concept d'agent. En plus, le système complexe (SIC dans notre contexte) qui a une structure arborescente (hiérarchique) est un système composé de sous-systèmes interconnectés, dont chacun a (lui-même) une structure arborescente, jusqu'à ce que nous atteignons le niveau le plus bas des systèmes élémentaires. Cette principale propriété, on peut la trouver dans les systèmes holoniques. Donc, nous avons utilisé les SMA et les systèmes holoniques pour la modélisation des SICs. Autrement dit, le SIC est modéliser par un SMAH. Cette modélisation nous amène à produire les propriétés suivantes les plus importantes dans la réalisation de notre approche, respectivement:

1. **Autonomie** : chaque SI dans notre système doit être perçu comme une entité autonome, ayant ses propres capacités de prise de décision, en fonction de ses connaissances pour fournir une solution partielle.
2. **Communication** : échange d'informations entre les agents holoniques pour comprendre et coordonner leurs décisions dans la résolution du problème global.
3. **Cohérence** : ce concept fait allusion au fait que le système se comporte comme une seule unité pour prendre une meilleure décision.
4. **Coopération**: c'est travailler ensemble pour trouver des solutions à un problème commun.
5. **Hiérarchie**: il est nécessaire que le système global ait une structure arborescente pour assurer la classification correcte des sous-systèmes, chacun en fonction de son niveau et de ses activités.
6. **Résolution distribuée de problèmes** : un problème peut être décomposé en sous-parties, où chacune pouvant être résolue de façon indépendante pour trouver une solution stable.
7. **Vitesse d'exécution** : qui fait allusion au concept du parallélisme, car plusieurs agents peuvent travailler en même temps pour la résolution d'un problème.

La nature des SICs est bien adaptée avec l'approche des SMA. Donc, l'utilisation des SMAH est proposée pour la modélisation et la mise en œuvre des SICs. Plus précisément, chaque agent holonique représente un SI possède une CCF laquelle va décrire son point de vue et sa décision pour résoudre un problème distribué donné.

IV.4. Architecture globale du système

L'architecture proposée qui est baptisée **DCSic-CFSH** pour l'aide à la **D**écision **C**ollective dans les **SICs** basée sur les **CCFs** et les **SMAH**, se compose de trois niveaux (voir figure IV.1). Le premier niveau est le SIC qui est le principal responsable de la résolution de tout problème distribué. Le deuxième niveau comprend un ou plusieurs sous-SIC qui composent le SIC principal. Chaque sous-SIC est un ensemble de systèmes d'information représenté au troisième niveau.

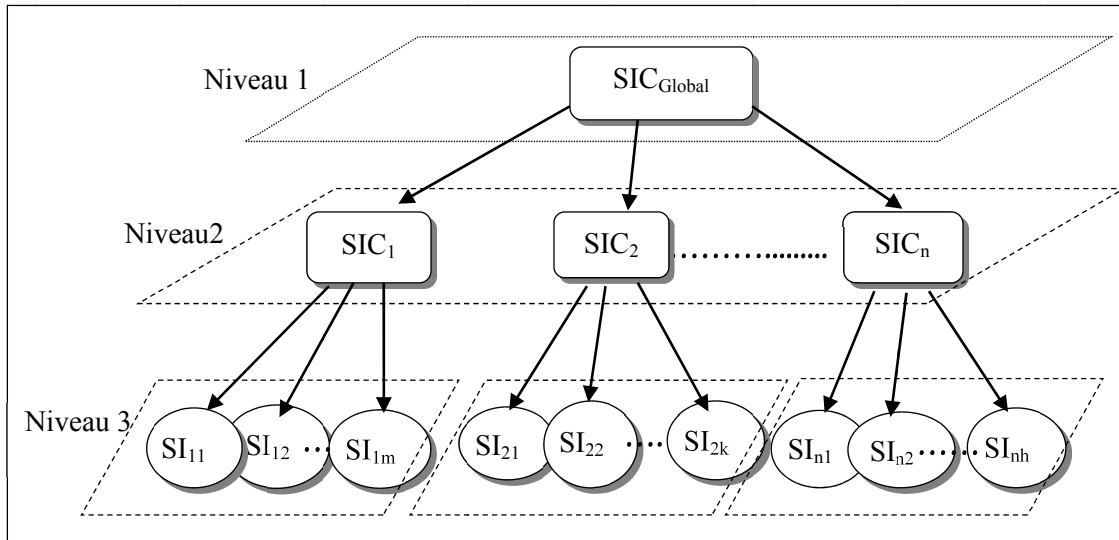


Figure IV.1. Architecture globale de DCSic-CFSH

Comme toute modélisation, l'approche multi-agent simplifiera le problème à traiter dans notre travail, elle permet en distribuant la recherche d'une solution partielle à chaque agent, de respecter et de réduire une grande partie de sa complexité globale. La figure suivante (figure IV.2) montre comment est structuré notre système en modélisant chaque sous système par un agent holonique qui a une CCF qui va décrire sa décision partielle.

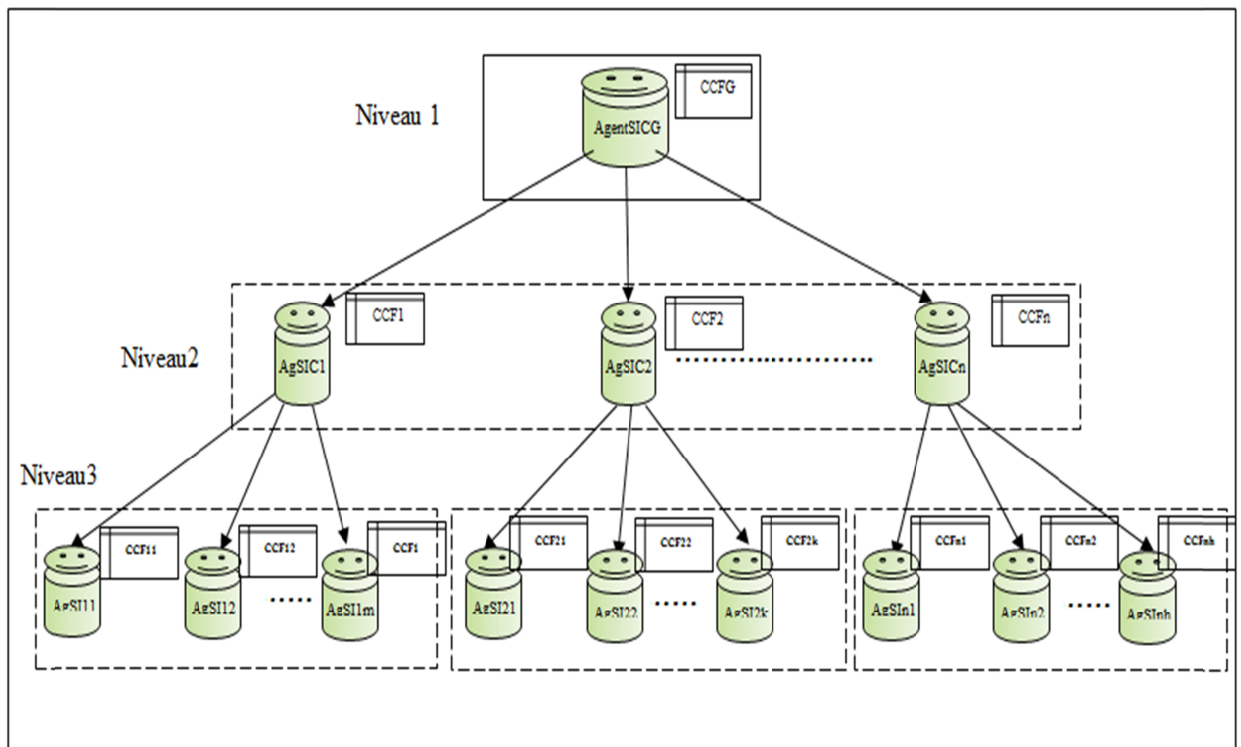


Figure IV.2. L'architecture SMAH modélisant le système initial décrit dans la figure IV.1

D'après la modélisation de notre système, il existe quatre types d'agents: agent SI, agent SIC, agent SICG et finalement agent analyseur (qui va être abordé ultérieurement). Dans ce qui suit, nous présentons la description et le rôle de chaque type d'agent.

IV.4.1. Types d'agents utilisés

- **Agent SI** : c'est un agent holonique qui représente un SI. Sa base de connaissance est sous forme d'une CCF qui est décrite par le SI correspond. Il gère l'interface entre le système et l'utilisateur (expert du domaine). Il assure un espace de travail privé pour le SI et un espace public pour le groupe d'agents (sous SIC). Il permet aussi à l'utilisateur de communiquer avec les experts du groupe en dehors du processus de prise de décision (en cas des contradictions), c'est-à-dire, il permet d'établir les communications avec les autres utilisateurs du système par l'intermédiaire de l'interface homme-machine. En coopération avec les mêmes types d'agents (agentSI), il sollicite le groupe d'agents à prendre chacun la décision partielle selon le degré d'expertise pour trouver une solution initiale au problème posé. Chaque agentSI possède un coefficient d'expertise (CE) qui reflète l'expérience de l'expert du domaine, une CCF qui va décrire la décision partielle pour résoudre le problème, une interface de perception (IP) qui est le port de communication avec l'environnement (expert du domaine dans notre cas), et une interface de communication (IC) qui constitue aussi le port de communication mais avec d'autre agentSI. La figure suivante (figure IV.3) montre l'architecture d'un agentSI.

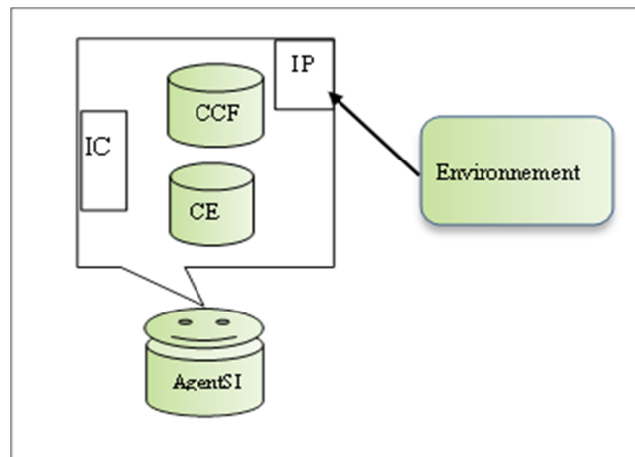


Figure IV.3. Architecture de l'agentSI

- **Agent SIC**: c'est un agent holonique qui représente un sous-SIC. Sa base de connaissances est sous forme d'une CCF résultante de la discussion de différentes décisions partielles de différents SIs composants. Chaque agentSIC possède un coefficient d'émergence (CEm) dont sa valeur est précisée par l'agent global, une CCF résultante de l'union entre les CCFs des agents SI qui va décrire aussi la décision partielle du sous SIC,

une interface de perception (IP) qui est le port de communication avec le troisième niveau du système (agents SI) pour recevoir la nouvelle CCF et aussi le coefficient d'émergence (CEm) attribué par l'agent global « agentSICG », et une interface de communication (IC) c'est aussi le port de communication mais avec d'autres agentSIC. La figure IV.4 montre l'architecture d'un agentSIC.

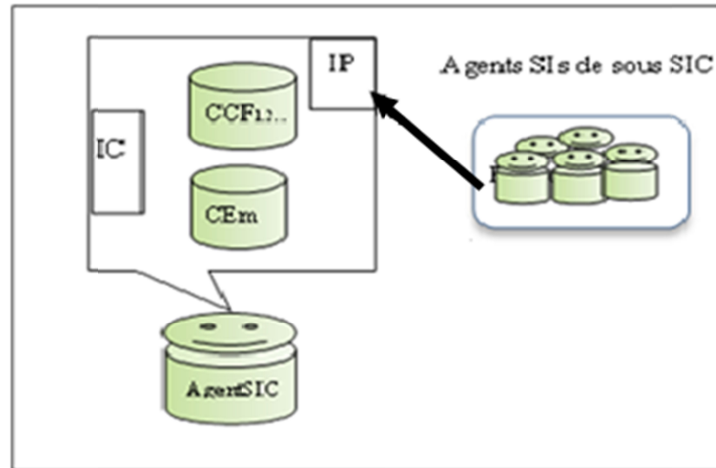


Figure IV.4. Architecture de l'agentSIC

- **Agent SICG** : c'est un agent holonique qui représente le SIC global. Sa base de connaissances est sous forme d'une CCF collective. Il est l'agent central du processus de la prise de décision. Il est l'agent superviseur du système. Son principal rôle est d'assurer le bon déroulement des différentes phases du processus de prise de décision. Il permet de démarrer le processus de décision par l'envoi d'un message qui contient le problème global aux différents agents de type agentSIC dans le deuxième niveau. L'agentSICG possède une CCF collective résultante de l'intersection entre les CCFs des agents SIC qui va décrire la décision globale de notre système, une interface de perception (IP) qui est le port de communication avec le deuxième niveau du système (agents SIC) pour recevoir la nouvelle CCF collective, et une interface de communication (IC) représentant toujours le port de communication mais avec les agents analyseurs. La figure IV.5 montre l'architecture d'un agentSICG.

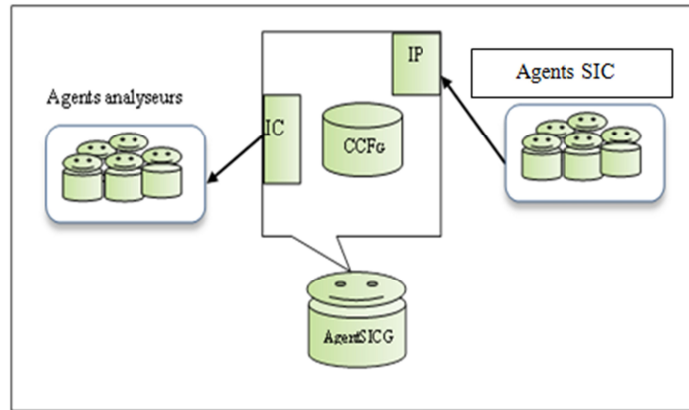


Figure IV.5. Architecture de l'agentSICG

- **Agent analyseur** : c'est un agent qui permet d'inférer et d'analyser la CCF collective résultante de la discussion de différents agents du système dont le but est d'extraire tous les comportements typiques (scénarios possibles). Autrement dit, considérer toutes les solutions possibles pour résoudre le problème posé. L'agent analyseur possède une CCF collective reçue par l'agentSICG, une interface de perception (IP) qui est le port de communication avec le premier niveau du système (agentSICG) pour recevoir la CCF collective, et un processus d'inférence qui est lancé dont le résultat d'exécution est une liste des comportements typiques qui va représenter les différentes solutions possibles. La figure suivante (figure IV.6) montre l'architecture d'un agent analyseur.

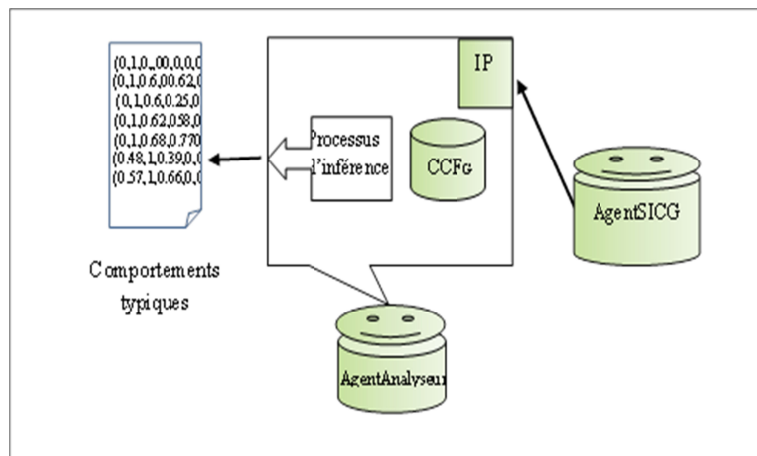


Figure IV.6. Architecture de l'agent analyseur

Dans le processus de la prise de décision collective dans notre système en vue d'une modélisation par les SMA, nous avons déduit que les agents doivent être cognitifs car ils doivent posséder des compétences et un savoir afin de réaliser la tâche qui leur a été affectée. Ainsi, les quatre types d'agents utilisés sont de type cognitif puisque :

Dans le cas des trois premiers agents (agentSI, agentSIC, agentSICG) chacun devra raisonner pour pouvoir prendre la décision afin de collaborer à la résolution partielle du problème en se fondant sur sa base de connaissances (CCF). Et pour le dernier agent (agent analyseur) qui va exécuter le processus d'analyse et d'inférence (va raisonner aussi) pour décider ou bien pour trouver les solutions possibles. Le fruit de la communication entre ces agents est d'obtenir une meilleure solution au problème posé à notre système. La figure suivante (Figure IV.7) montre la communication entre les différents agents du système DCSic-CFSH.

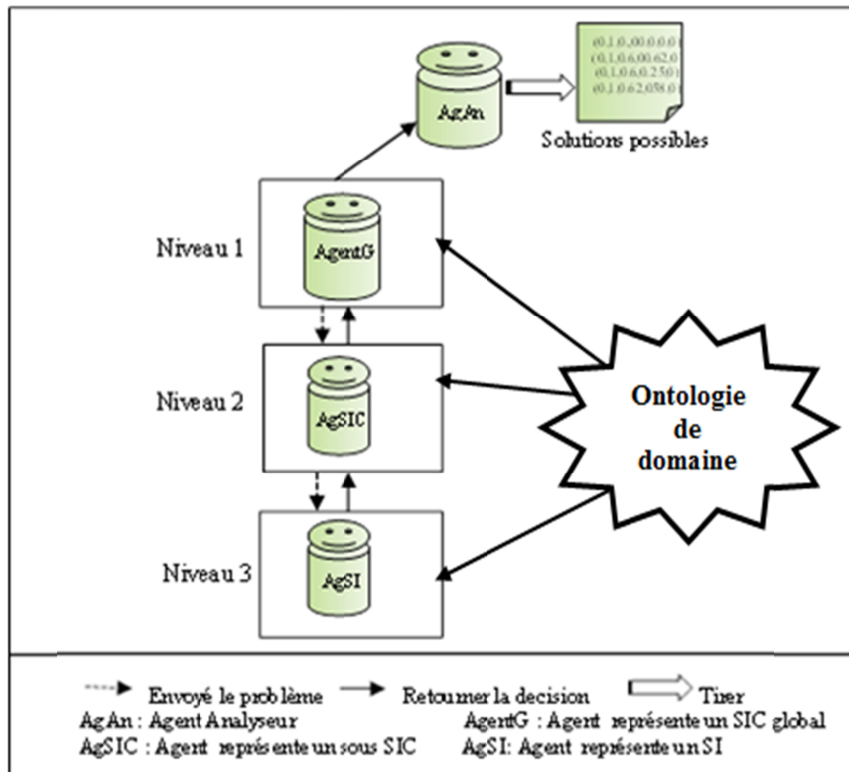


Figure IV.7. Communication entre les différents agents de DCSic-CFSH

IV.4.2. Opérations de base

La manipulation de l'information entre les CCFs des différents agents dans le SMA nécessite l'utilisation d'opérations d'union et / ou d'intersection (voir chapitre III section 3.3.3.3.c). Selon notre approche, nous avons proposé l'utilisation d'opération d'union entre les SIs et aussi l'opération d'intersection entre les sous SICs. La première opération augmente la base de connaissances des agents, ce qui permet la réduction des problèmes causés par les vues partielles de chaque agent. En d'autres termes, elle permet le partage de l'information pour promouvoir la collaboration entre les agents. Le processus d'union de deux CCFs est clarifié dans les figures (Figure.IV.8 et Figure.IV.9) qui montrent un exemple de résolution du problème de la sécurité routière. La CCF résultante est augmentée à cinq concepts et quatre relations de causalités à partir de deux CCFs de deux agents,

ce qui permet de partager plus d'information, de posséder une vue plus large sur le problème posé, et de donner une décision plus optimale et plus réaliste.

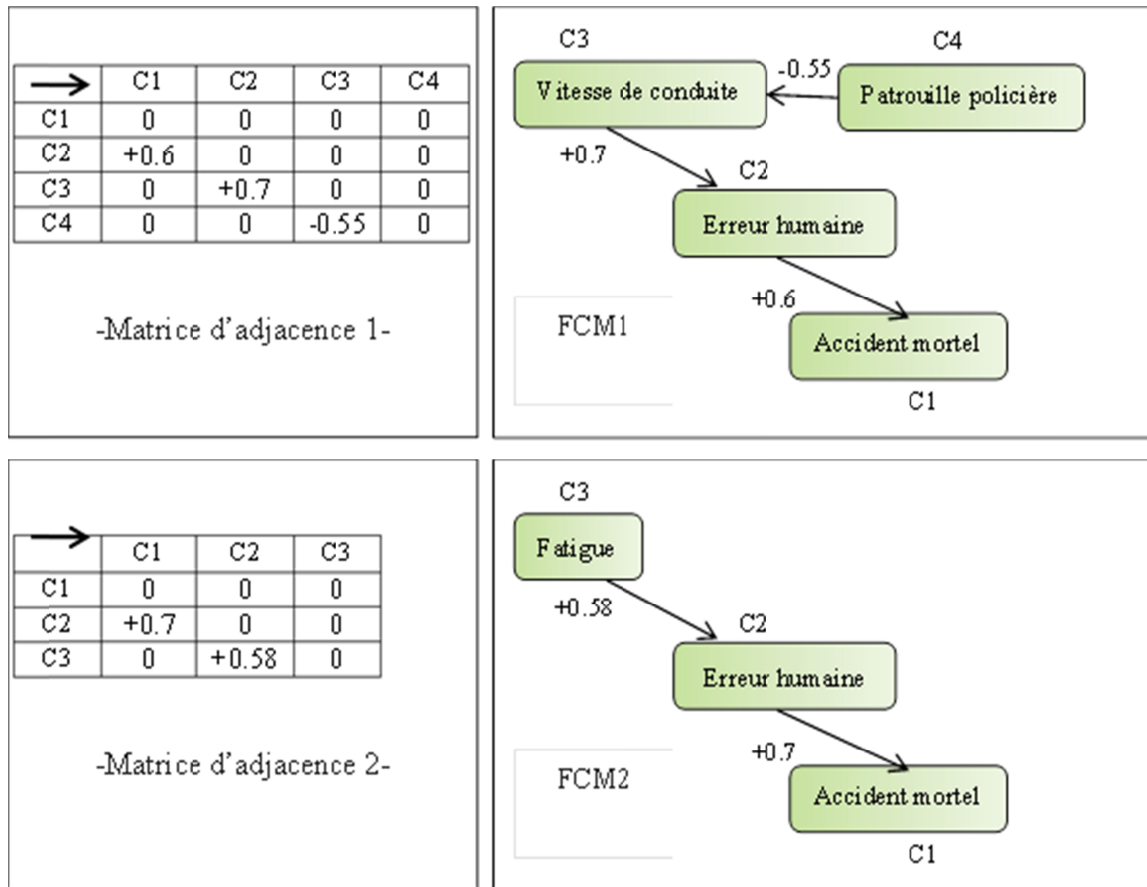


Figure IV.8. Exemple de deux CCFs de deux agents

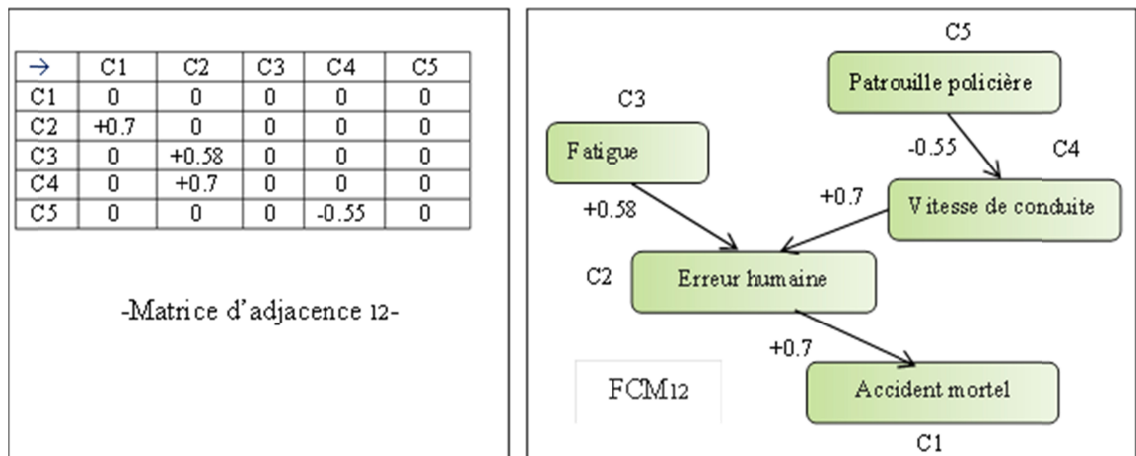


Figure IV.9. Union de deux cartes: CCF1 U CCF2 = CCF12

La deuxième opération d'intersection peut également être utilisée pour :

- ✓ renforcer les valeurs des relations causales dans la carte cognitive résultante, plus spécifiquement, trouver des accords sur les concepts et les relations de causalités par les différents agents dont le but de chercher des solutions réalisables.
- ✓ Vérifier qu'il n'y a pas l'ambivalence; autrement dit, qu'il n'y a pas des contradictions dans les décisions ou bien dans les points de vue des agents relatifs à un problème donné.
- ✓ et aussi peut être utilisée si deux agents ont été conçus pour le même but pour régler le problème de l'inflation de la CCF résultante, c-à-d, trouver une CCF la plus optimale et la plus convenue par les agents.

D'une part, Zhang [Zhang, W.R. et al, 1992] spécifie que pour obtenir la composition (union) des cartes cognitives, des paires de concepts identiques de chaque CCF sont reliées à des liaisons bidirectionnelles dont le poids est (+1). En outre, il établit que la décomposition de ces cartes peut être obtenue en changeant les poids des liaisons de 1 à 0. Kosko [Kosko, B., 1992] propose l'union entre les CCFs en combinant les matrices d'adjacence (A_1, \dots, A_n). Ceci est obtenu en additionnant les matrices augmentées des cartes (E_1, \dots, E_n) avec l'addition traditionnelle. L'utilisation de l'opération d'union entre deux cartes cognitives est possible si elles contiennent au moins un concept commun qui permet de les relier.

Pour exécuter l'opération d'union entre les CCFs dans DCSic-CFSH, nous choisissons la méthodologie de Kosko. L'union se fait de façon progressive; entre deux cartes cognitives, plus spécifiquement, entre les matrices d'adjacence de deux cartes A_1 et A_2 . Le nombre de cartes est limité à deux à la fois pour utiliser les règles d'union (la même chose avec l'opération d'intersection). Par conséquent, deux matrices E_1 et E_2 augmentées sont obtenues à partir de deux matrices d'adjacence A_1 et A_2 extraites de deux cartes (chaque matrice augmentée (E_i) avec n lignes et n colonnes, n est le nombre total de concepts). Enfin, l'opération d'union de E_1 et E_2 est appliquée. Cette dernière est effectuée à chaque couple de concepts communs entre les deux matrices, en prenant la valeur maximale entre elles [Chauvin, L., 2010]. Le processus d'union de deux CCFs est clarifié dans le deuxième chapitre.

IV.4.3. Ontologie de domaine

Selon les travaux précédents [Maziz, A., Zarour, N., 2014 ; 2015], nous avons observé que pour chaque exécution de notre processus, nous ne considérons pas la côté sémantique. Dans l'étape de la fusion des différentes CCFs (soit par union, ou par intersection), notre système ne contrôle pas les concepts sémantiquement. Plus précisément, si deux concepts ne sont pas similaires, le système les considère comme deux concepts différents, bien que ces concepts soient sémantiquement identiques. Pour ces raisons, nous proposons d'associer une ontologie de domaine à notre système.

Donc dans notre travail, le problème du nombre de concepts augmenté causé par l'absence du côté sémantique est très important. De ce fait, nous avons proposé une méthode pour appliquer l'approche ontologique dans l'intégration des CCFs (union ou intersection). L'un des objectifs principaux de l'utilisation de l'ontologie de domaine est l'efficacité pour la réutilisation, enrichir sémantiquement notre système et le transfert de nouvelles connaissances. Cette efficacité peut être mesurée en obtenant une solution optimale (sous la forme d'une CCF).

IV.5. Description détaillée du système DCSic-CFSH

Notre travail est le fruit de différents paradigmes qui sont SIC, SMAH, CCF et le concept d'ontologie. Ces paradigmes nous apportent les éléments nécessaires pour construire notre approche (DCSic-CFSH), dont l'objectif est d'améliorer le processus de la prise de décision afin d'obtenir la meilleure décision pour résoudre les problèmes distribués. Dans ce qui suit, nous présentons en détail les principales étapes de notre démarche afin d'atteindre nos objectifs.

IV.5.1. Position du problème

Le point de départ de DCSic-CFSH est de décrire et poser le problème à résoudre à notre système. Le problème est envoyé par la suite aux différents niveaux du DCSic-CFSH. Par conséquent, chaque niveau du SIC et selon ses composantes va décrire sa décision partielle sous forme de CCF; Il s'agit donc d'une communication homme-machine. Par la suite, chaque CCF qui décrit la décision partielle est associée à chaque agent holonique qui va représenter le SI correspondant.

IV.5.2. Construction de la décision collective (CCF collective)

L'étape de la construction de la décision collective est l'étape la plus importante dans DCSic-CFSH. Elle est construite elle-même par un autre ensemble d'étapes successives permettant d'accumuler les différentes décisions partielles pour obtenir une décision globale. Les sections suivantes montrent le déroulement de chaque étape.

La construction de la CCF collective est basée principalement sur les différentes CCFs proposées par les différents SIs dans le troisième niveau. Dans les trois premières sections suivantes, nous montrons comment nous construisons les différentes CCFs des différents SIs.

IV.5.2.1. La sélection de la source d'informations

Dans le troisième chapitre (section 3.3.3.3.d.1), il est montré qu'il existe deux sources d'informations : soit le document soit un groupe d'experts (par questionnaire). Dans DCSic-CFSH, nous avons choisi la deuxième source pour rester dans l'actualité, dans le réalisme de

n'importe quel domaine d'application et aussi en cas de l'émergence des contradictions, avec notre système nous contactons directement l'expert du domaine pour trouver des solutions.

IV.5.2.2. L'acquisition de la carte cognitive

Selon le domaine d'application, le SI est proposé pour déléguer à prendre une décision. Généralement, au niveau de chaque SI, il existe des experts de domaine. Donc l'étape de l'acquisition de la CCF est faite à l'aide d'un groupe d'experts. Autrement dit les différents experts de domaine:

- Premièrement, ils vont identifier les concepts pertinents pour le domaine à analyser,
- Puis, ils vont limiter le nombre de concepts en évaluant leur importance,
- Et finalement, ils vont choisir les relations de causalité avec des poids entre ces concepts.

Autrement dit, les experts de domaine vont poser initialement la décision primaire au problème posé sous forme d'une CCF.

IV.5.2.3. La conversion de la carte en forme de matrice

Pour que notre système (dans la machine) comprenne le contenu de la CCF, il faut donc la transformer en son langage (langage machine). Généralement, la forme graphique de la carte cognitive permet de faciliter sa conception et sa compréhension. Donc, pour exploiter la CCF il faut la transformer sous une forme matricielle, où tous les mécanismes proposés dans DCSic-CFSH se basent sur la représentation matricielle. De ce fait, avant de passer aux étapes suivantes, il faut tout d'abord extraire la matrice d'adjacence de chaque CCF.

L'étape suivante est l'exécution des différentes opérations. Autrement dit, appliquer la fusion des différents points de vue pour obtenir la décision globale pour résoudre un problème donné.

IV.5.2.4. Coopération entre les agents SI dans le troisième niveau

a. Coefficient d'expertise

Dans la théorie de la décision, un expert de domaine qui a une CCF, il possède aussi un coefficient d'expertise qui reflète son expérience dans le domaine d'application à analyser. Donc, dans DCSic-CFSH, nous avons modélisé chaque SI par un agent holonique, de sorte que chaque CCF est associée à l'agent correspondant. L'agentSI qui a une CCF possède aussi un coefficient d'expertise (CE) qui va préciser l'agent qui a la CCF résultante de la coopération (l'agentSI qui un CE le plus grand c'est l'agent qui possède la CCF résultante de la coopération).

b. Exécution de l'opération d'union des CCFs entre les agents SI

Dans le troisième niveau de DCSic-CFSH, les agents holoniques (représentant les SIs) doivent coopérer ou parvenir à un accord au cours de leurs activités pour résoudre le problème; la coopération est faite à l'aide de l'opération d'union.

Selon les contraintes liées à l'utilisation des opérations (union, intersection) [Kosko, B., 1992], la coopération se fait de façon incrémentielle (Figure IV.10). Chaque deux agent holoniques vont exécuter l'opération d'union entre leurs CCFs (par échange des messages), le résultat est associé à l'agent avec le plus grand coefficient d'expertise (CE). Par la suite, l'agent qui détient la CCF résultante continuera à coopérer avec les agents restants, de la même manière. Le résultat de la coopération entre les différents agents holoniques est la génération d'une nouvelle CCF qui est automatiquement associée à l'agent représentant le sous-SIC. L'ontologie du domaine est utilisée au cours de la coopération. Pour minimiser le nombre des concepts générés dans la CCF résultante et pour enrichir sémantiquement notre système.

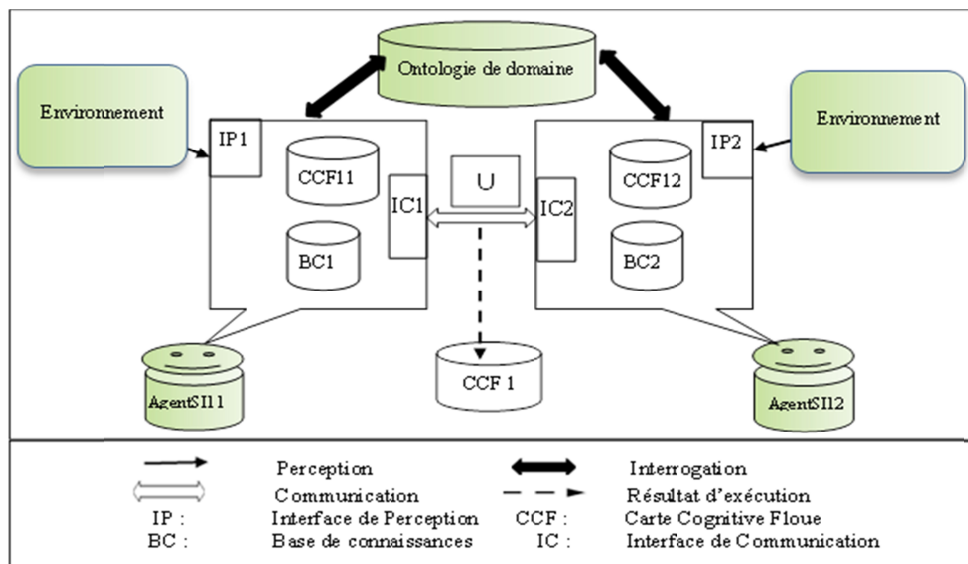


Figure IV.10. Exécution de l'opération d'union entre les agents SI (décideurs)

Dans le deuxième niveau du système, les agents SIC vont recevoir un ensemble de CCFs dont le nombre est égal à celui des sous-SIC ce qui est clarifié dans la figure IV.2.

IV.5.2.5. Coopération entre les agents SIC dans le deuxième niveau

a. Distribution du coefficient d'émergence sur les agents qui représentent les sous SIC

Au niveau de chaque sous SIC, nous obtenons une nouvelle CCF. Par conséquent, dans le deuxième niveau de DCSic-CFSH, nous avons un ensemble d'agents holoniques, chacun possède

une CCF. Le problème identifié est que ces agents holoniques ayant des spécificités différentes. Ici, la question posée est de savoir comment maîtriser ces spécificités?

Afin de maîtriser différentes spécificités, nous proposons d'utiliser un autre paramètre nommé coefficient d'émergence (CEm). La difficulté réside dans la détermination de la valeur de ce coefficient, puisque nous considérons plusieurs facteurs. Ces derniers peuvent inclure, par exemple, si chaque sous-SIC a une perspective différente, un pourcentage de gain différent, un pourcentage de participation différent et aussi la possibilité de changer le vue pour s'abonner ou non dans le SIC global pour résoudre le problème. Donc, la difficulté est de savoir comment profiter de l'utilisation de CCFs puisque nous avons des agents avec des sous-objectifs différents; Il existe une concurrence entre eux pour atteindre l'objectif global.

La décision de la valeur du coefficient d'émergence est le rôle de l'agent qui représente le SIC global, selon la fonction d'évaluation qui dépend les différentes spécificités. Le coefficient d'émergence est distribué aux différents agents holoniques (représentant les différents sous-SIC) après la décision de sa valeur par l'agent qui représente le SIC global (Figure IV.11.).

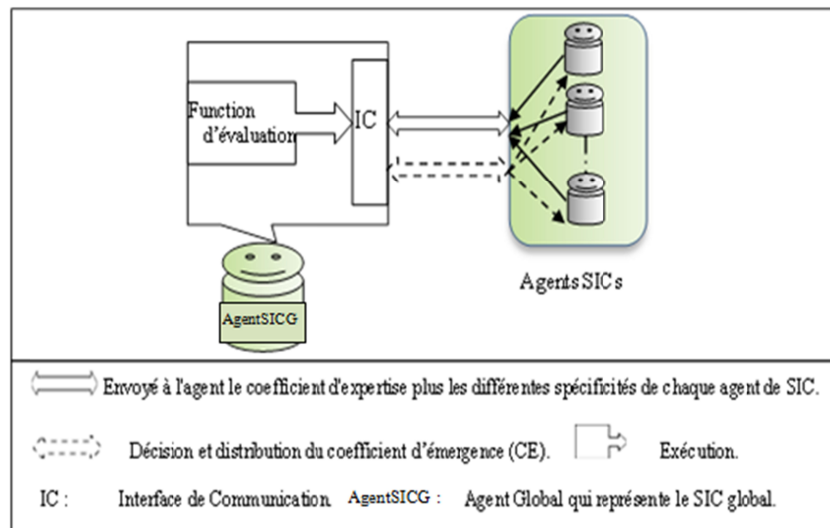


Figure IV.11. Architecture d'agent global représentant le SIC (AgentSICG)

b. Exécution de l'opération d'intersection entre les agents SIC

Une fois la distribution de la CEm est faite, les agents holoniques représentant les sous-SIC coopèrent ensemble dans le deuxième niveau. Cette coopération est réalisée en utilisant l'opération d'intersection (Figure IV.12.) qui permet de renforcer les relations causales dans la CCF collective résultante. Cela garantit que les différents agents holoniques sont dans la bonne voie pour résoudre le problème considéré. Et aussi, on utilisant une ontologie de domaine pour traiter la coté sémantique dans notre système.

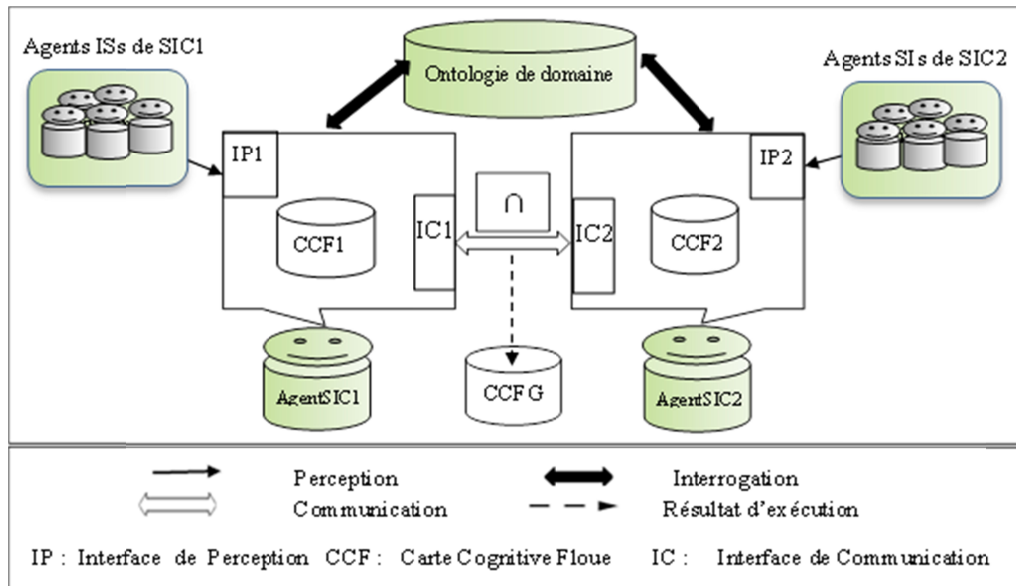


Figure IV.12. Exécution de l'opération d'intersection entre les agents SIC

Le résultat de l'exécution de l'opération l'intersection est une CCF collective qui représente le point de vue global de notre système.

Dans les différents niveaux de notre architecture, les agents holoniques coopèrent en utilisant un protocole d'interaction par rapport au domaine d'application. Donc, dans notre système, les interactions entre les différents agents de DCSic-CFSH sont basées sur un protocole d'interaction coopératif. Ce dernier est le responsable sur le traitement des requêtes dans DCSic-CFSH. Il permet aux agents d'échanger des messages structurés, de les contrôler et de faciliter leur coordination.

Le protocole d'interaction est exécuté en plusieurs étapes. Dans ce qui suit, nous présentons petit à petit le déroulement de ce protocole entre les différents agents du système en utilisant plusieurs chronogrammes, dont le but de bien clarifier et montrer la relation entre les différents types d'agents.

Dans la première étape, l'agent qui représente le SIC global va envoyer un message simple en langage naturel qui contient le problème à résoudre aux différents agentSICs. Par exemple, si on prend comme domaine d'application la sécurité routière :

- le message envoyé est comme suit : ('Comment réduire le nombre d'accidents mortels ?');
- avec un paramètre booléen nommé « **Problem** » qui a la valeur **False** dont le but est de confirmer la réception du problème à résoudre, et en retour ce paramètre va prendre la valeur **True** dans le cas où une décision partielle est trouvée au problème posé.

La figure suivante (figure VI.13) montre un chronogramme de cette première étape.

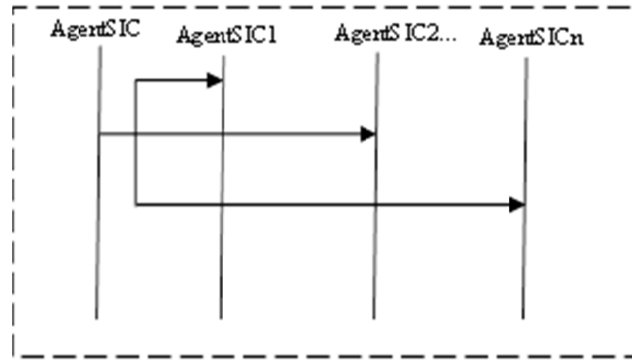


Figure IV.13. Chronogramme de démarrage du processus de la prise de décision

Dans la deuxième étape, chaque agent représentant un sous SIC va envoyer le même message aux différents agentSIs. Ce message contient le problème à traiter avec le même paramètre booléen qui contient la valeur False. Le chronogramme suivant montre cette étape au niveau d'un seul sous SIC (exemple SIC1).

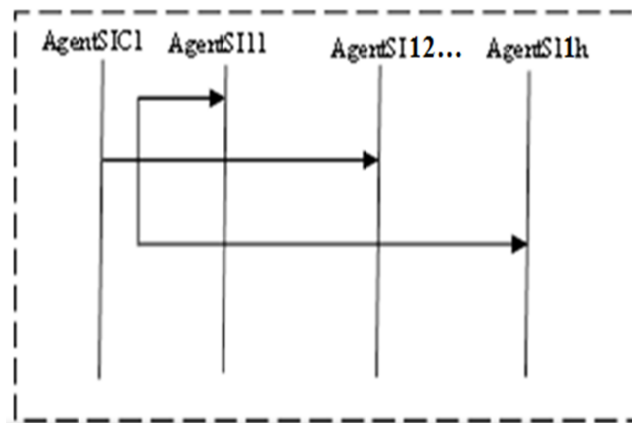


Figure IV.14. Chronogramme d'envoi l'agentSIC1 le problème aux agentSIs

Au niveau de chaque sous SIC, les différents agentSIs vont coopérer et vont répondre par un message qui contient une décision partielle (sous forme d'une matrice) au problème posé.

Donc, si nous prenons le même exemple précédant, les agents SI qui sont délégués par les experts du domaine vont recevoir le problème à résoudre (directement ou à distance). La réponse de chaque agentSI est sous forme d'une CCF (plus spécifiquement sous forme d'une matrice d'adjacence) avec le même précédant paramètre booléen **Problem** (prend la valeur **True**).

La figure suivante (figure VI.15) montre la réponse des agentSIs au premier message reçu.

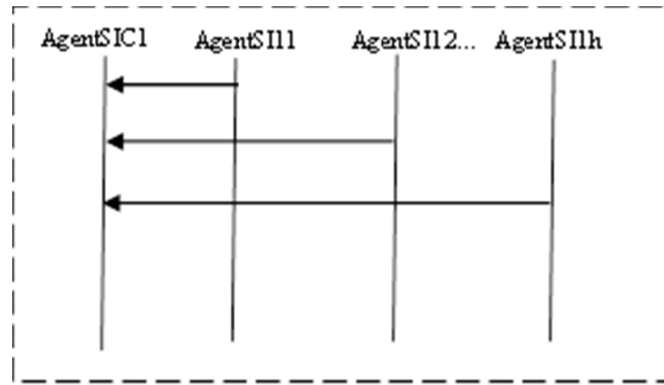


Figure IV.15. Chronogramme du retour des réponses au niveau du sous SIC1

La dernière étape est construite par la réception des différentes décisions partielles (sous forme des matrices) prises par les différents agentSICs. Une fois les messages des différents agentSICs sont reçus avec toujours le paramètre **Problem** qui a la valeur **True**, alors l'agentSIC va les fusionner pour obtenir une décision collective au problème posé. Le chronogramme suivant montre cette étape.

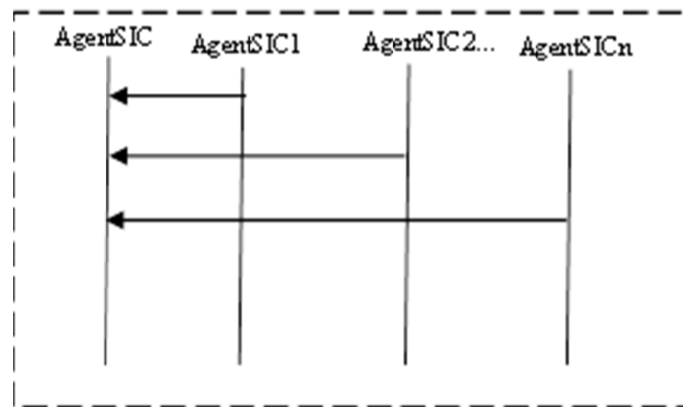


Figure IV.16. Chronogramme du retour des réponses au niveau de l'agentSIC

Dont le but de résumer le déroulement de l'échange des messages entre les différents agents du notre protocole d'interaction, la figure suivante (figure VI.17) montre ces différentes étapes.

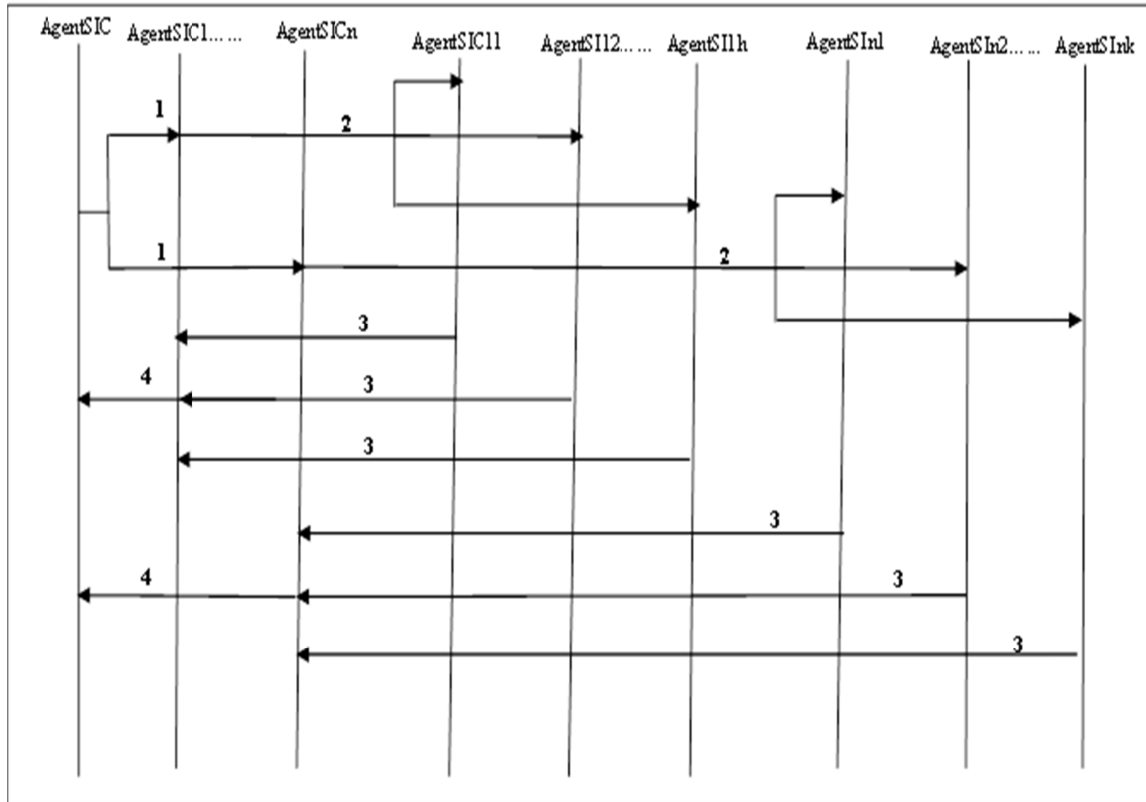


Figure IV.17. Chronogramme du protocole d'interaction

IV.5.2.6. Exécution du processus d'analyse et d'inférence sur la CCF collective du système

Le résultat de l'opération d'intersection est une nouvelle CCF globale. Une fois la carte cognitive résultante est obtenue, l'analyse est effectuée. Cela est fait par d'autres types d'agents appelés analyseurs. La communication entre l'agent global et l'agent analyseur est illustrée dans la figure IV.18.

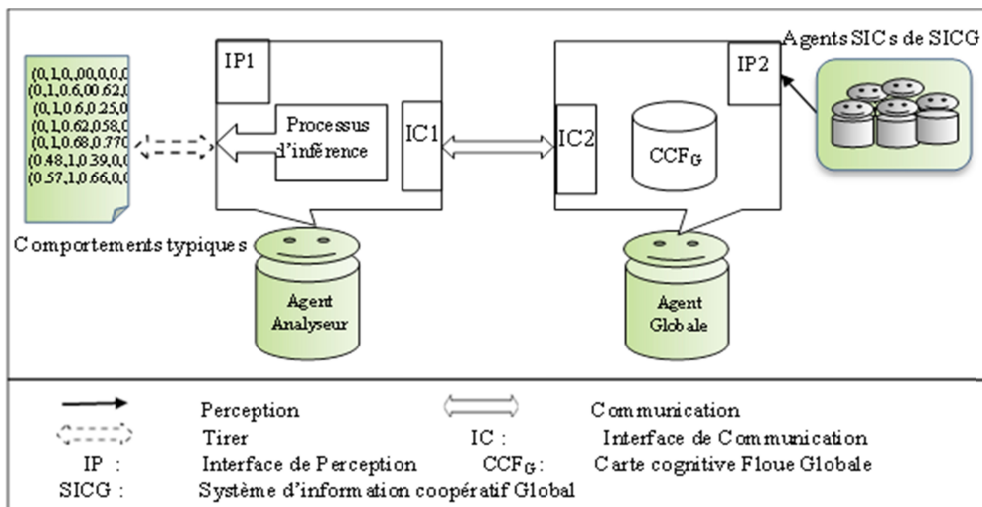


Figure IV.18. Communication entre l'agent global (SIC) et l'agent analyseur

Le travail associé à l'agent analyseur est l'exécution du processus d'analyse et d'inférence sur la CCF collective résultante. En d'autre terme, chaque agent analyseur est invité à tirer quelques comportements typiques (scénarios possibles). Le processus d'analyse et d'inférence est le même que celui proposé dans de nombreux travaux [Hengjie, S. et al, 2008, Meghabghab, G., 2001] et le même que décrit dans le troisième chapitre, à l'exception de la fonction seuil $S(x)$ [Chauvin, L. 2010, Papageorgiou, E.I., 2012] qui est décrite comme suit:

$$\text{Bivalent} \quad S(x) = \begin{cases} 0, & x \leq 0.5 \\ 1, & x > 0.5 \end{cases}$$

Le processus d'analyse et d'inférence est la partie la plus importante dans les CCFs, qui aident à la prise de décision, à la prévision des états futurs et à l'explication des actions passées. Ce processus est résumé par l'algorithme suivant:

Algorithme: Processus d'analyse et d'inférence

```

- Supposons que nous avons une carte cognitive de n concepts
  1. Extraire la matrice d'adjacence M [i, j].
  2. Préparer le vecteur d'excitation a1.
    /* Initialiser le vecteur excitateur à zéro (0). */
    Pour i = 1 à n faire
      a1 [i] ← 0;
    Fin pour
    /* Chaque valeur i représente l'état du concept i. */
  -Mettre à 1 les concepts voulus. C'est-à-dire les concepts qu'on veut voir qu'arrivera-t-il
  s'ils sont activés.
  - Ajouter a1 à la mémoire de travail MT.
    i ← 1;
  3. Tant que i <= n faire
    ai+1 = ai X M [i, j];
    Si ai+1 ∈ MT alors ai+1 est un comportement typique.
    Sinon Pour i = 1 à n faire
      Si ai+1 [i] ≥ 0,5 alors ai+1 [i] ← 1;
      Sinon ai+1 [i] ← 0;
    Fin Si
    Fin Pour
    Ajoutez ai+1 à MT
  Fin si
  i ← i + 1;
Fin Tant que

```

IV.5.2.7. Construction de la base de connaissances globale (les solutions possibles au problème global posé)

Enfin, nous pouvons construire notre base de connaissances globale par l'inférence sur tous les scénarios possibles obtenus en parallèle par les agents analyseurs, c'est-à-dire, voir qu'arrivera-t-il si n'importe quelle combinaison possible de concepts est activée. Dans cette dernière étape, un rapport détaillé est généré, contenant tous les comportements typiques qui vont représenter les solutions possibles au problème posé, avec une interprétation logique dans notre système. Cette dernière est un critère d'arrêt de DCSic-CFSH.

IV.6. Conclusion

Ce chapitre nous a permis de faire la liaison entre les trois premiers chapitres. c-à-d comment nous avons utilisé les CCFs dans l'aide à la décision collective d'un SIC. Nous avons vu comment notre système permet de trouver des solutions à tout problème distribué dans le contexte coopératif (SIC) à l'aide de la CCF, le SMAH et le concept d'ontologie. Ces solutions sont obtenues par la coopération de tous les agents décideurs en utilisant l'outil CCF pour faciliter et améliorer le processus de la prise de décision, SMAH pour modéliser et mettre en œuvre le système et l'ontologie pour l'enrichir sémantiquement.

Pour mettre notre approche en application, le chapitre suivant a pour vocation d'exposer la partie conception et implémentation de notre travail.

*Chapitre V : Application et
expérimentation*

V.1. Introduction

Dans ce chapitre, nous appliquons l'architecture DCSic-CFSH sur l'étude de cas de la sécurité routière. A travers cette application, nous visons à voir comment améliorer le processus de la prise de décision dans un contexte coopératif afin de trouver la meilleure décision pour résoudre le problème posé.

La description de notre cas d'étude fait l'objet de la première section. Les quatre sections suivantes décrivent l'environnement de l'application, l'application elle-même et quelques expérimentations avec une comparaison avec d'autres travaux similaires.

V.2. Exemple : Sécurité Routière

Pour illustrer notre approche, nous considérons le domaine de la sécurité routière comme un exemple d'application où le processus de la prise de décision est très important.

L'Algérie est parmi les trois premiers pays dans le monde où le pourcentage d'accidents mortels augmente chaque jour. Ce phénomène reste un grave problème de santé publique, aussi social (familles sans revenus après le décès du chauffeur), économique (perte pour les assurances), ... aux niveaux mondial, régional et national. Si des mesures sont prises dans de nombreux pays pour améliorer la sécurité routière, beaucoup reste à faire si nous voulons que le nombre de décès continue de diminuer. En Algérie, les accidents de la route ont fait 1.343 morts et 14.500 blessés durant les cinq premiers mois de 2017, selon les statistiques du Centre national de prévention et de sécurité routière (CNPSR) citées par l'agence officielle [TSA, 2017]. Donc, la sensibilisation des personnes à la crise des routes et la recherche d'une méthodologie pour minimiser le nombre d'accidents mortels est le projet du siècle. C'est l'objectif principal de l'exemple proposé.

V.3. Environnement d'application

Tout au long de notre travail, nous avons utilisé trois plateformes. Chacune a ses avantages et ces inconvénients selon les besoins de notre architecture. Premièrement, nous avons utilisé la plateforme JADE pour développer les SMA de notre architecture. Sous JADE [Maziz, A., Zarour, N., 2014], nous avons toujours besoin de créer des agents interface pour identifier la nature de la relation entre les différents agents, ce qui consomme beaucoup d'espace mémoire. Par exemple, le premier SMA de notre architecture est composé d'agents structurés d'une manière hiérarchique. Cette principale propriété, nous pouvons la trouver dans les SMAH. Cependant, il existe peu de plateformes pour le développement des SMAH, on trouve par exemple: Janus, Zeus, Magique. Cette dernière est utilisée dans l'implémentation d'une partie de notre travail de thèse qui a fait l'objet d'une communication internationale [Maziz, A., Zarour, N., 2015]. Au cours du

développement de notre application, nous proposons d'intégrer le concept d'ontologie pour traiter la coté sémantique de notre travail. Ce qui manque dans la plateforme Magique.

Donc, selon notre architecture notre application a besoin d'une plateforme qui contient les propriétés suivantes :

- ✓ Une plateforme de développement des SMAH ; permet de développer une architecture hiérarchique pour construire le premier SMA qui représente le SIC global ;
- ✓ Contient une interface Java qui permet de montrer la hiérarchie des agents composants de notre système;
- ✓ Permet d'intégrer une ontologie ou plus à l'application ;
- ✓ Permet de suivre le processus d'échange des messages entre les agents ;
- ✓ Permet d'ajouter un protocole coopératif basé sur les interactions entre les différents agents;

Nous retrouvons ces propriétés dans la plateforme AgentBuilder (Figure V.1).

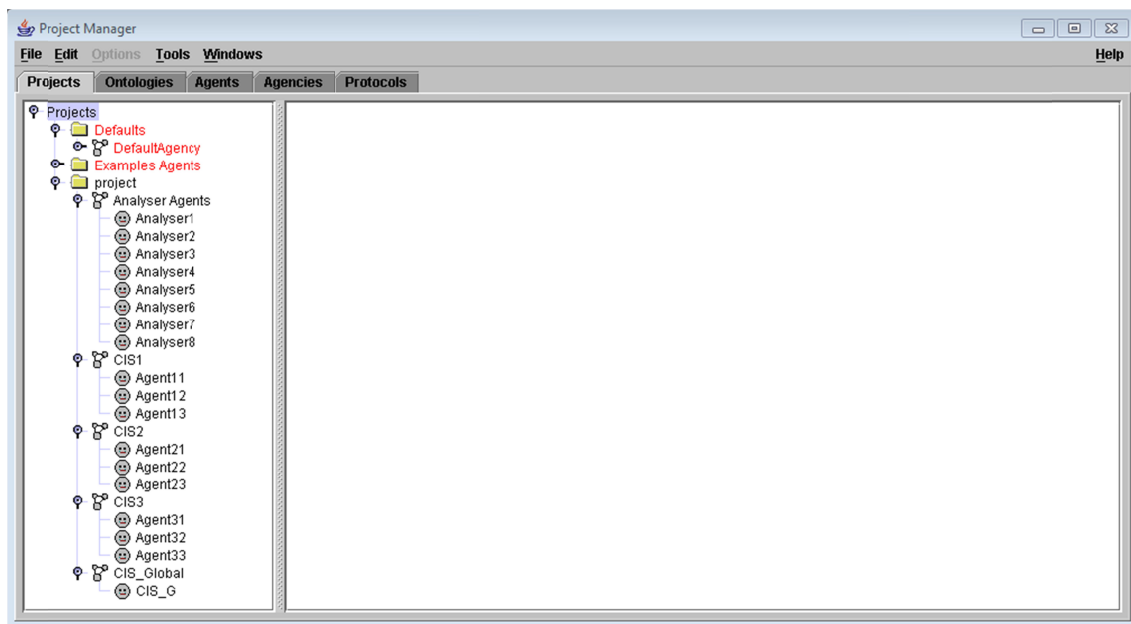


Figure V.1. Console de la plateforme AgentBuilder

AgentBuilder [AgentBuilder, 2016] est une boîte à outils logicielle intégrée qui permet aux développeurs de logiciels de développer rapidement des agents à base d'agents logiciels intelligents et des applications. Le développement d'agents est le processus de définition du comportement de l'agent. AgentBuilder fournit des outils pour spécifier les règles comportementales, les croyances initiales, les engagements, les intentions et les capacités des agents.

V.4. Application

V.4.1. Position du problème

D'après l'exemple d'application, le problème posé à notre architecture est de savoir comment sensibiliser les gens de telle façon à réduire le nombre d'accidents et particulièrement ceux mortels. Donc, le point de départ de DCSic-CFSH est de décrire et poser le problème à résoudre à notre système.

Premièrement, le Centre National de la Prévention et de la Sécurité Routière (CNPSR) de l'Algérie est associé au premier niveau qui est le SIC mondial de DCSic-CFSH. Deuxièmement, le CNPSR pose le problème à ses différents centres locaux de la prévention et de la sécurité routière (CLPSR) dans différents wilayas qui sont représentés par les SICs secondaires dans le deuxième niveau de DCSic-CFSH. Troisièmement, à tour de rôle, chaque CLPSR pose le problème aux différents services concernés (par exemple la Direction de la sécurité nationale (DSN), la météorologie et la direction des transports (DT) ...) qui sont représentés par les SIs.

L'architecture du système avec notre exemple est illustrée dans la figure suivante (Figure V.2) qui montre trois sous-SICs et trois SIs (éventuellement plus) dont le but est de voir la communication entre les agents sous la plateforme utilisée.

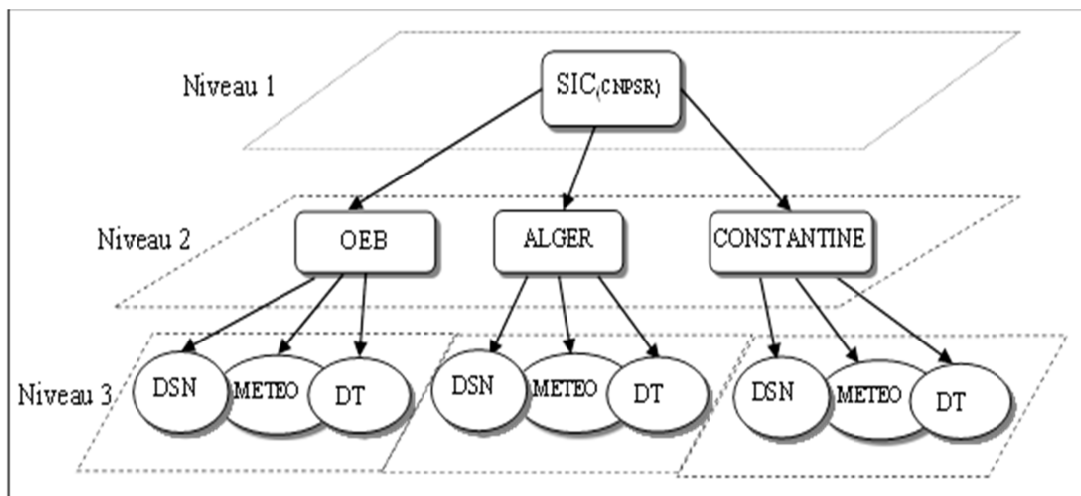


Figure V.2. Architecture proposée avec l'exemple de la sécurité routière

V.4.2. Construction de la décision collective (CCF collective)

Dans DCSic-CFSH, le système se compose en deux sous-systèmes multi-agents, le premier est un système holonique qui représente le système d'information coopératif (CNPSR), et le deuxième est un système multi-agents qui se compose d'agents analyseurs.

Nous avons utilisé trois sous-SIC (CLPSR) modélisés par trois agents holoniques et chaque sous-SIC (CLPSR) est composé de trois SIs (services concernés) modélisés par trois agents holoniques.

La décision partielle de chaque SI (par exemple la direction de la sécurité nationale (DSN), la météorologie et la direction des transports (DT) ...) est donnée par une CCF qui est présentée par un ou groupe d'experts du domaine.

V.4.2.a. Coopération entre les agentsSIs dans le troisième niveau (exécution de l'union des CCFs entre les agents SIs)

Nous avons modélisé chaque SI par un agent holonique sous AgentBuilder. Donc, chaque CLPSR est un SMAH composé de trois agents holoniques dont chacun a la CCF correspondante. Sous AgentBuilder, chaque agent holonique a un ensemble de règles qui peuvent décrire son comportement.

Ces agents coopèrent entre eux en utilisant l'opération d'union pour obtenir une nouvelle CCF commune. Et aussi une ontologie de domaine pour traiter la coté sémantique dans notre système. La CCF résultante contient une description de la décision du CLPSR correspondant. Chaque CCF résultante est associée à un nouvel agent holonique (représente CLPSR).

V.4.2.b. Coopération entre les agentsSICs dans le deuxième niveau (exécution de l'intersection des CCFs entre les agents SICs)

Une autre fois, les nouveaux agents holoniques superviseurs coopèrent ensemble en utilisant l'opération d'intersection pour l'obtention de la CCF collective du SIC global (CNPSR). L'utilisation de l'opération d'intersection permet de renforcer les valeurs des relations causales dans la CCF résultante. Sachant qu'il est utilisé aussi une ontologie de domaine pour minimiser le nombre des concepts générés dans la CCF collective.

La communication entre différents agents sous AgentBuilder est illustrée dans la Figure V.3.

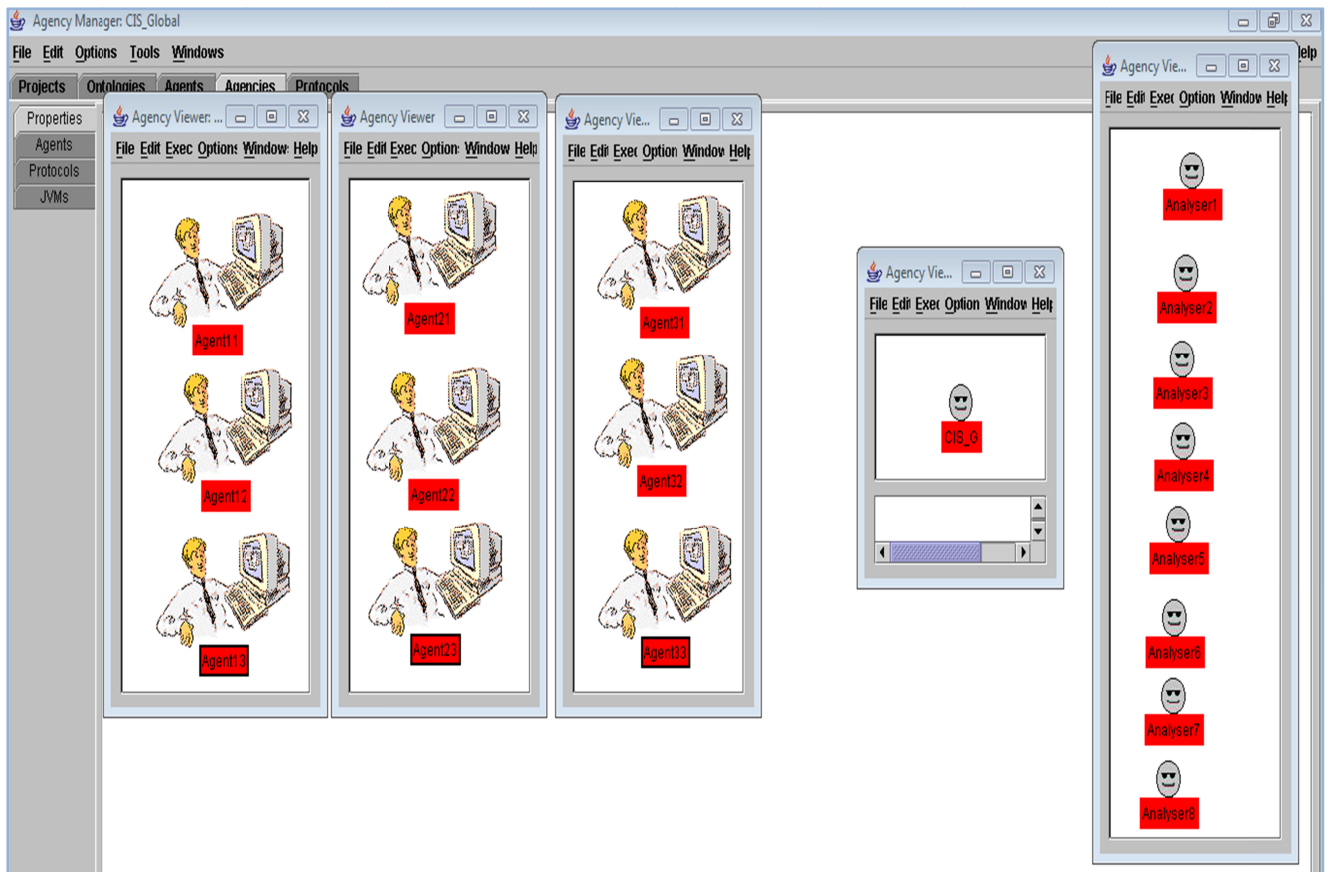


Figure V.3. Communication entre les différents agents sous AgentBuilder [Maziz, A., Zarour, N., in press]

La CCF collective résultante sous AgentBuilder est illustrée dans la figure V.4 suivante :

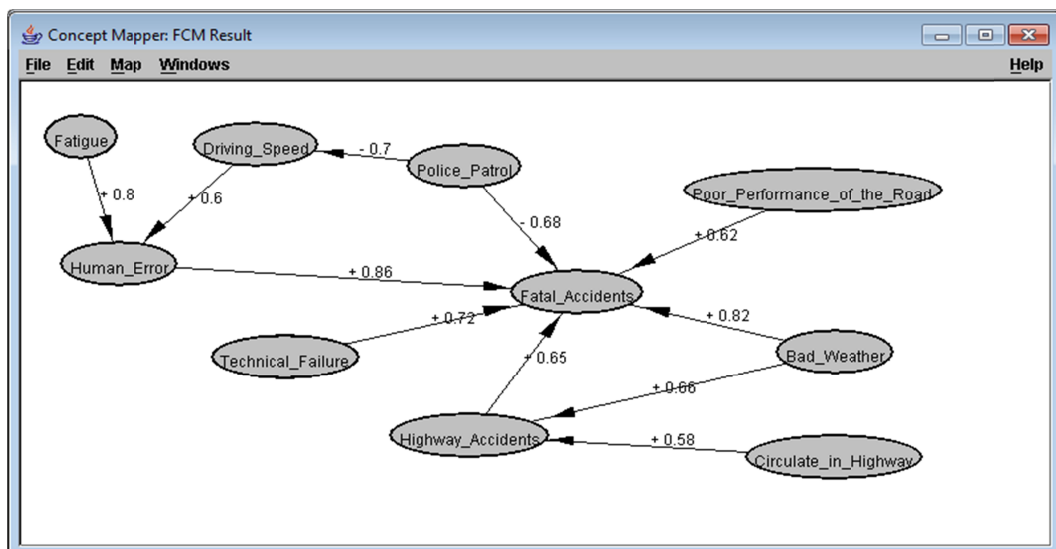


Figure V.4. Carte cognitive floue collective [Maziz, A., Zarour, N., in press]

V.4.2.c. Exécution du processus d'analyse et l'inférence sur la CCF collective

La CCF collective obtenue est associée au deuxième système multi-agent qui est composé d'un groupe d'agents analyseurs. Ces derniers exécutent en parallèle le processus d'analyse et d'inférence sur la CCF collective. Par conséquent, chaque agent analyseur va tirer un ensemble de comportements typiques (scénarios possibles) qui contribue à trouver les solutions au problème de la crise (sécurité routière). Le processus d'analyse et d'inférence est une technique dans laquelle les CCF sont utilisées pour la prise de décision, la prévision des états futurs et l'explication des actions passées.

La figure V.5 montre le résultat d'exécution du processus d'analyse et d'inférence, qui contient trois parties principales: (1) montre les différents agents qui composent notre architecture, (2) montre le temps d'exécution du processus en Milliseconde (3) montre la liste des scénarios possibles (comportements typiques) ou des solutions possibles.

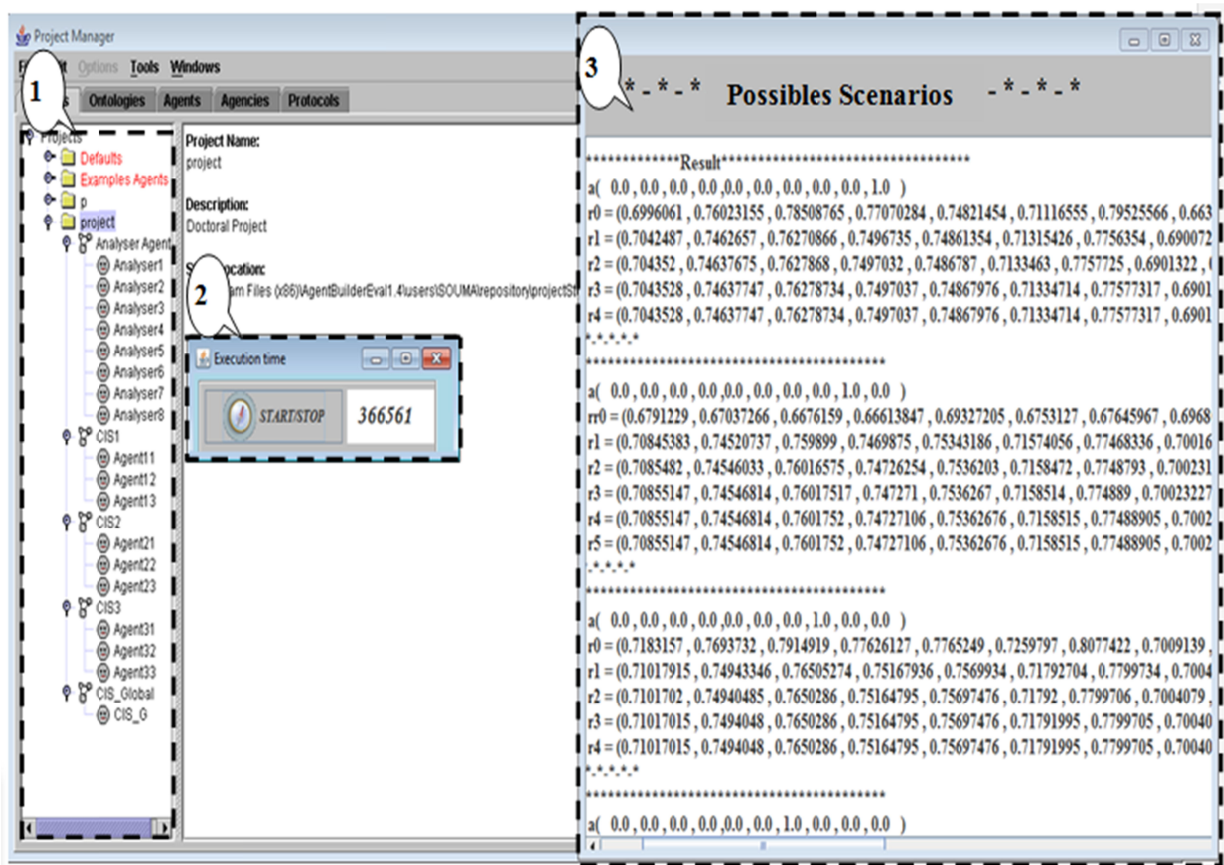


Figure V.5. Résultat d'exécution du processus d'analyse et d'inférence [Maziz, A., Zarour, N., in press]

V.4.2.d. La construction de la base de connaissance globale (les solutions possibles au problème posé)

La base de connaissance globale est construite par l'inférence sur tous les scénarios possibles, c'est-à-dire, voir qu'arrivera-t-il si n'importe quelle combinaison possible de concepts est activée.

Dans ce qui suit, nous essayons de clarifier comment inférer pour construire et développer la base de connaissance de notre système :

Selon notre exemple de la sécurité routière, la première étape est constitué d'extraire la matrice d'adjacence à partir de la CCF collective (figure V.6) avec $n=5$ (dont le but est de voir le déroulement du processus d'inférence) comme suit :

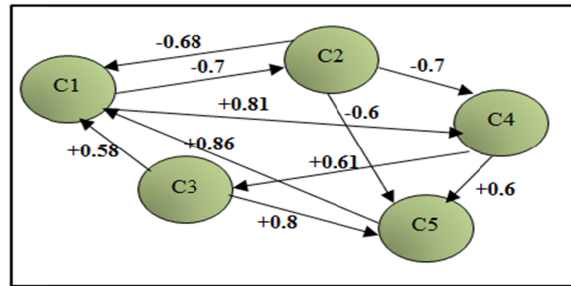


Figure V.6. Exemple d'une CCF collective

| Lien \Rightarrow | C1 | C2 | C3 | C4 | C5 |
|--------------------|-------|------|-------|-------|------|
| C1 | 0 | -0.7 | 0 | +0.81 | 0 |
| C2 | -0.68 | 0 | 0 | -0.7 | -0.6 |
| C3 | +0.58 | 0 | 0 | 0 | +0.8 |
| C4 | 0 | 0 | +0.61 | 0 | +0.6 |
| C5 | +0.86 | 0 | 0 | 0 | 0 |

- Matrice d'adjacence $M [i, j]$ -

C1= Accident-mortel ; C2= Patrouille-Policière;
 C3= Erreur-humain; C4= Vitesse-de-conduite;
 C5= Mauvais-temps.

La deuxième étape est constitué de préparer le vecteur exciteur a_1 comme suit :

- ✓ Initialiser le vecteur exciteur à zéro (0).
 / Chaque valeur i représente l'état de concept i /
 $a_1 = (0, 0, 0, 0, 0)$.
- ✓ Mettre à 1 les concepts voulus. C'est-à-dire les concepts qu'on veut voir ; qu'arrivera-t-il s'ils sont activés.
 Par exemple si nous voulons voir qu'arrivera-t-il si les concepts C1 et C4 sont activés.
 Donc, $a_1 = (1, 0, 0, 1, 0)$.
- ✓ Ajouter a_1 à la mémoire de travail MT (contient tous les vecteurs-résultants du processus d'inférence).

La troisième étape est constituée de faire le produit matriciel $\mathbf{a}_{i+1} = \mathbf{a}_i \times \mathbf{M}_{ij} \quad /i=1,2,\dots,$

La quatrième étape est constituée de vérifier si le vecteur obtenu \mathbf{a}_{i+1} appartient à la mémoire de travail alors nous avons obtenu un comportement typique. Sinon appliquer la fonction de seuil (voir chapitre IV.5.2.6) comme suit :

$$S(\mathbf{a}_{i+1}) = \begin{cases} 0, & \mathbf{a}_{i+1} \leq 0.5 \\ 1, & \mathbf{a}_{i+1} > 0.5 \end{cases}$$

Par la suite le vecteur \mathbf{a}_{i+1} est ajouté à la mémoire de travail.

L'étape suivante constitue d'incrémenter i et aller à la troisième étape. Le processus s'arrête si l'indice $i > n$.

Le résumé de ce processus est comme suit:

$\mathbf{a}_{i+1} = \mathbf{S}(\mathbf{a}_i \times \mathbf{M}_{ij}) \quad /i=1, 2, \dots, \text{Imax}$ où \mathbf{S} est la fonction de seuil. Soit on laisse l'algorithme se stabiliser à soi-même ou on le fixe par un nombre d'itérations maximal Imax .

Le résultat d'exécution de ce processus est un vecteur de dimension $1 \times n$ qui contient la liste des réponses. Chaque élément i de ce vecteur qui est différent à 0 indique que le concept i va avoir une augmentation si les concepts voulus du départ ont une augmentation et vice-versa.

Avec notre exemple, si on veut connaître qu'arrivera-t-il si les concepts « Accident-mortel » et « Vitesse-de-conduite » sont activés, on fait l'inférence comme suit :

Vecteur exciteur $\mathbf{a}_1 = (1, 0, 0, 1, 0)$.

1- $\mathbf{a}_1 \times \mathbf{M}_{ij} = (0, -0.7, +0.61, +0.81, +0.6) \Rightarrow$ Fonction de seuil $\Rightarrow (0, 0, 1, 1, 1) = \mathbf{a}_2$

2- $\mathbf{a}_2 \times \mathbf{M}_{ij} = (+1.44, 0, +0.61, 0, 1.4) \Rightarrow$ Fonction de seuil $\Rightarrow (1, 0, 1, 0, 1) = \mathbf{a}_3$

3- $\mathbf{a}_3 \times \mathbf{M}_{ij} = (+1.44, -0.7, 0, +0.81, +0.8) \Rightarrow$ Fonction de seuil $\Rightarrow (1, 0, 0, 1, 1) = \mathbf{a}_4$

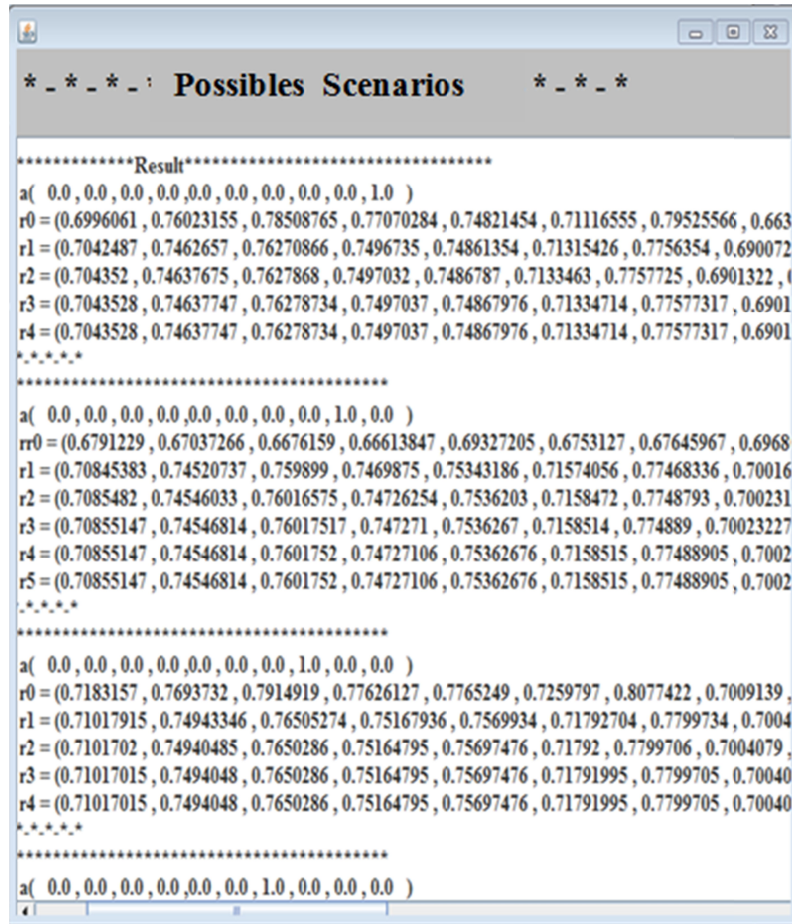
4- $\mathbf{a}_4 \times \mathbf{M}_{ij} = (+0.86, -0.7, +0.61, +0.81, +0.6) \Rightarrow$ Fonction de seuil $\Rightarrow (1, 0, 1, 1, 1) = \mathbf{a}_5$

5- $\mathbf{a}_5 \times \mathbf{M}_{ij} = (+1.44, -0.7, +0.61, +0.81, +1.04) \Rightarrow$ Fonction de seuil $\Rightarrow (1, 0, 1, 1, 1) = \mathbf{a}_6 = \mathbf{a}_5$

On peut interpréter le résultat comme suit : si les concepts « Accident-mortel » et « Vitesse de conduite » sont activés, alors les concepts « Erreur-humain » et « Mauvais-temps » auraient une augmentation. Autrement dit, plus le nombre d'accidents mortels et la vitesse de conduite augmentent, plus la probabilité que l'erreur humaine et le mauvais-temps sont les principales causes du problème de la sécurité routière.

Donc, dans cette dernière étape les agents analyseurs vont tirer tous les comportements typiques (solutions possibles) concernant le problème de la sécurité routière dans un seul rapport (nommé base de connaissances). Ce rapport (figure V.7) est une liste complète des états du système dans

tous les cas d'activation de toutes les combinaisons possible des concepts relatives au problème posé.



```
* _ * _ * _ : Possibles Scenarios * _ * _ *  
  
*****Result*****  
a( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0 )  
r0 = (0.6996061, 0.76023155, 0.78508765, 0.77070284, 0.74821454, 0.71116555, 0.79525566, 0.663  
r1 = (0.7042487, 0.7462657, 0.76270866, 0.7496735, 0.74861354, 0.71315426, 0.7756354, 0.690072  
r2 = (0.704352, 0.74637675, 0.7627868, 0.7497032, 0.7486787, 0.7133463, 0.7757725, 0.6901322, 0.6901  
r3 = (0.7043528, 0.74637747, 0.76278734, 0.7497037, 0.74867976, 0.71334714, 0.77577317, 0.6901  
r4 = (0.7043528, 0.74637747, 0.76278734, 0.7497037, 0.74867976, 0.71334714, 0.77577317, 0.6901  
*.*.*.*  
  
*****  
a( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0 )  
rr0 = (0.6791229, 0.67037266, 0.6676159, 0.66613847, 0.69327205, 0.6753127, 0.67645967, 0.6968  
r1 = (0.70845383, 0.74520737, 0.759899, 0.7469875, 0.75343186, 0.71574056, 0.77468336, 0.70016  
r2 = (0.7085482, 0.74546033, 0.76016575, 0.74726254, 0.7536203, 0.7158472, 0.7748793, 0.700231  
r3 = (0.70855147, 0.74546814, 0.76017517, 0.747271, 0.7536267, 0.7158514, 0.774889, 0.70023227  
r4 = (0.70855147, 0.74546814, 0.7601752, 0.74727106, 0.75362676, 0.7158515, 0.77488905, 0.7002  
r5 = (0.70855147, 0.74546814, 0.7601752, 0.74727106, 0.75362676, 0.7158515, 0.77488905, 0.7002  
*.*.*.*  
  
*****  
a( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0 )  
r0 = (0.7183157, 0.7693732, 0.7914919, 0.77626127, 0.7765249, 0.7259797, 0.8077422, 0.7009139,  
r1 = (0.71017915, 0.74943346, 0.76505274, 0.75167936, 0.7569934, 0.71792704, 0.7799734, 0.7004  
r2 = (0.7101702, 0.74940485, 0.7650286, 0.75164795, 0.75697476, 0.71792, 0.7799706, 0.7004079,  
r3 = (0.71017015, 0.7494048, 0.7650286, 0.75164795, 0.75697476, 0.71791995, 0.7799705, 0.70040  
r4 = (0.71017015, 0.7494048, 0.7650286, 0.75164795, 0.75697476, 0.71791995, 0.7799705, 0.70040  
*.*.*.*  
  
*****  
a( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0 )
```

Figure V.7. Exemple de la base de connaissance sous AgentBuilder [Maziz, A., Zarour, N., in press]

Le concept d'ontologie dans la plate-forme AgentBuilder est une spécification d'une conceptualisation qui permet de définir les connaissances de domaine dont l'agent a besoin pour fonctionner dans son environnement. Cela nous a permis de profiter d'utiliser l'ontologie de domaine dans notre application (voir Figure V.8.). Cette ontologie nous a permis de réduire le nombre de concepts générés par les opérations d'union et d'intersection, et aussi d'enrichir sémantiquement notre système.

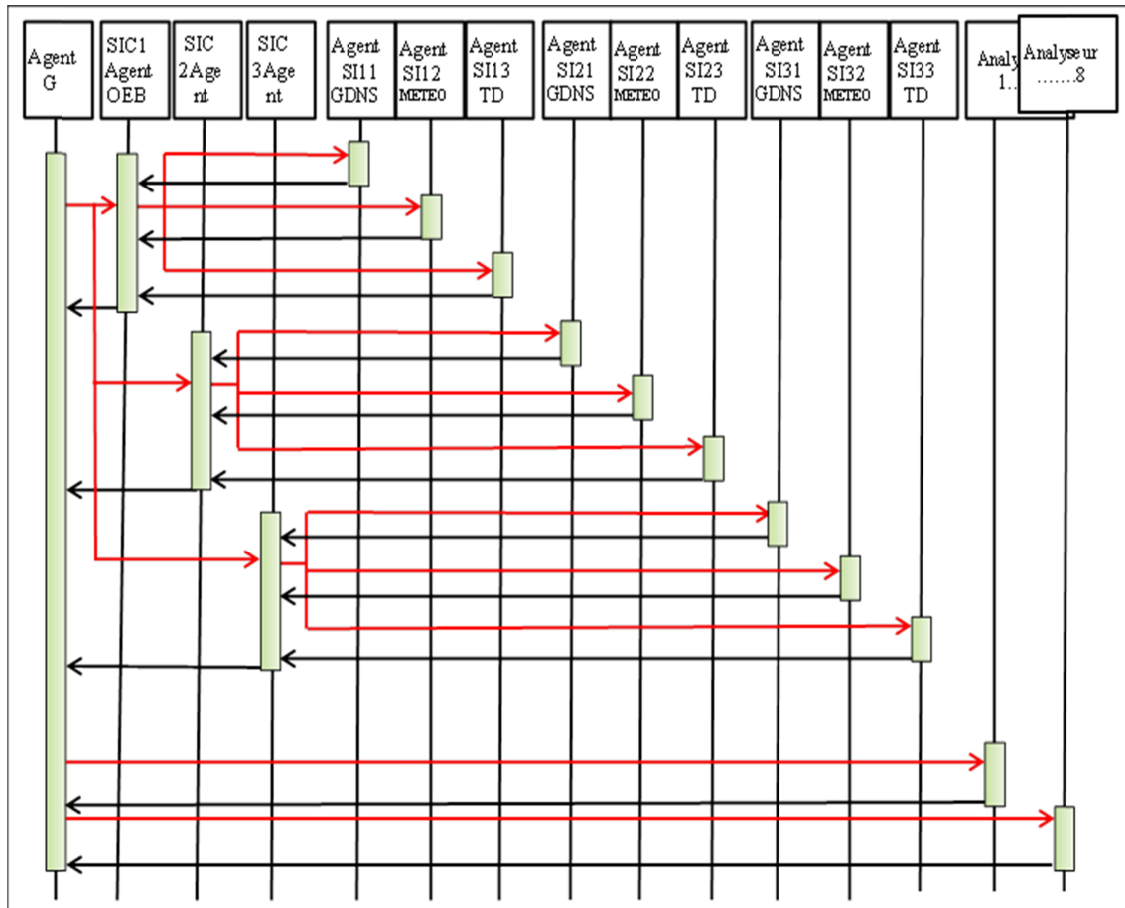


Figure V.9. Modélisation AUMML des interactions entre les différents agents

Selon la modélisation AUMML, l'agent global de la plate-forme AgentBuilder lancera trois agents holoniques (représentant des centres locaux dans différents wilayas) par un message contenant le problème global. Ces trois agents lancent également d'autres agents holoniques avec le même message. Maintenant, les agents holoniques coopèrent ensemble (chacun selon son niveau) pour trouver la bonne décision en tant que CCF. Celui-ci est retourné à l'agent mondial pour le donner au deuxième SMA pour l'analyse et l'inférence.

Le traitement des requêtes dans DCSic-CFSH est effectué selon un protocole coopératif basé sur les interactions entre les différents agents de DCSic-CFSH. Donc, notre protocole permet aux agents d'échanger des messages structurés, de les contrôler et de faciliter leur coordination. La figure V.10 montre le protocole utilisé sous AgentBuilder dans notre système sachant que chaque état i représente l'état de l'agent i .

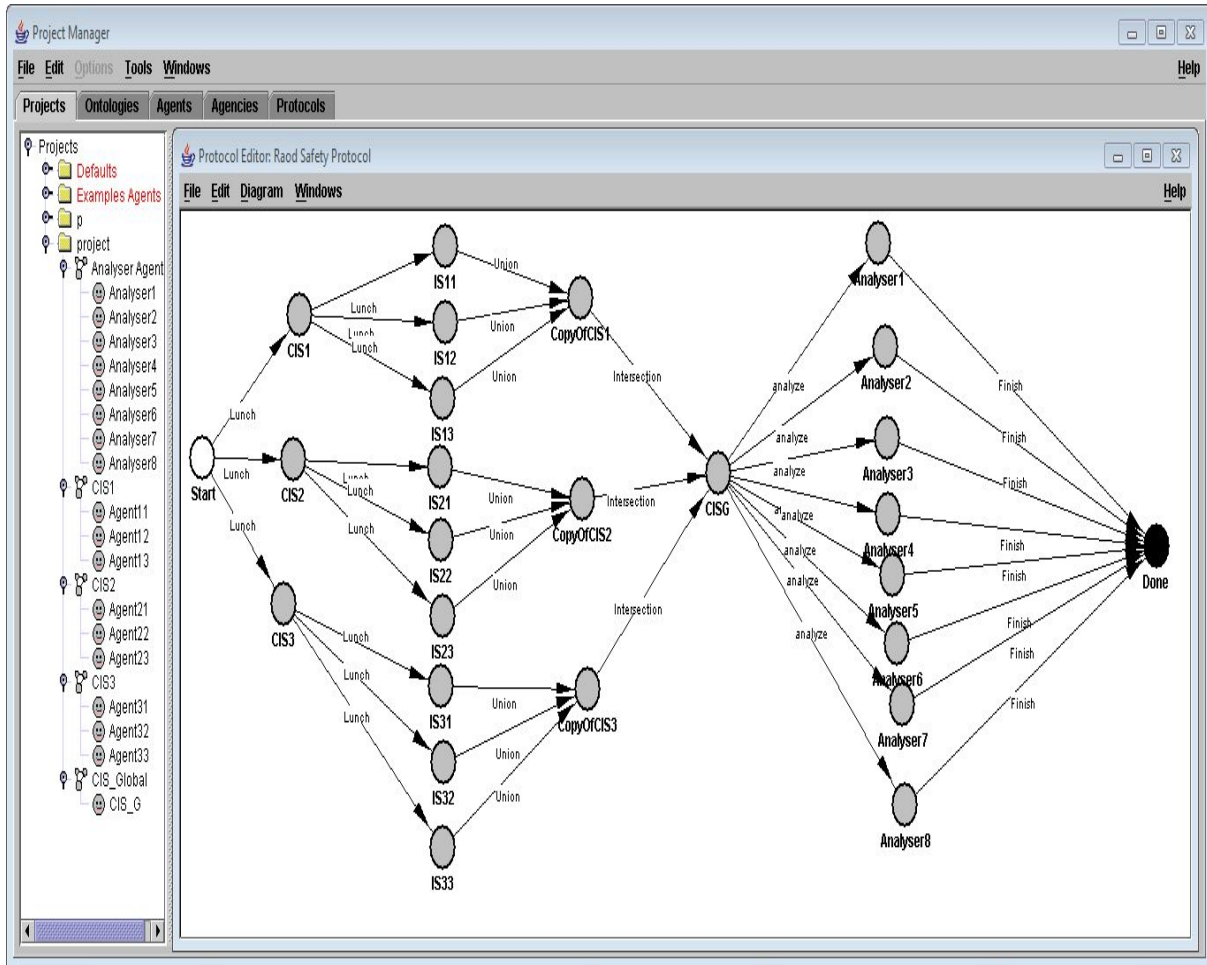


Figure V.10. Protocole de la sécurité routière [Maziz, A., Zarour, N., in press]

V.5. Expérimentation

Dans notre exemple, la CCF collective obtenue contient dix concepts donc le deuxième SMA (agents analyseurs) tirera $2^{10}-2$ comportements typiques en quelques minutes (exécution du processus d'inférence Figure V.5), ce qui présente l'avantage de réduire réellement la mémoire utilisée et le temps d'exécution.

- En termes de mémoire; Les agents de la plateforme AgentBuilder ont une structure hiérarchique, et selon APIJava, la relation entre les agents minimise leur nombre, d'où un gain considérable d'espace mémoire. En revanche, dans un autre scénario d'utilisation de la plateforme Jade dans l'implémentation de notre système [Maziz, A., Zarour, N., 2014], nous avons toujours besoin de créer des agents interface pour identifier la nature de la relation entre les agents, ce qui consomme beaucoup d'espace mémoire;
- En termes de temps d'exécution ; dans notre architecture et selon notre exemple, l'agent qui représente le SIC global lancera en parallèle les trois agents SIC. Chaque agent SIC lancera également en parallèle les agents SI. En d'autres termes, l'agent SIC recevra les décisions partielles en parallèle (en même temps) des différents agents SIC pour résoudre le

problème distribué. De plus, l'analyse du CCF globale obtenue est réalisée en parallèle par un groupe d'agents analyseurs. Par conséquent, nous gagnons beaucoup de temps en ce qui concerne l'exécution de différents agents de notre architecture.

La décision collective du SIC est obtenue par la dynamique de la CCF collective. Par exemple, si nous voulons savoir comment minimiser le nombre d'accidents mortels en Algérie, la question posée à notre système: *que se passe-t-il si le nombre d'accidents mortels augmente?* Ainsi, avec notre application, nous allons activer le concept appelé accident-mortel et observer les comportements typiques obtenus (scénarios possibles).

Tout au long de notre travail, nous avons effectué plus de milles expériences. Nous avons choisi dix expériences (chacune représente l'état du système dans le cas de l'activation de chaque concept de la CCF collective) qui apparaissent dans le diagramme suivant (Figure V.11):

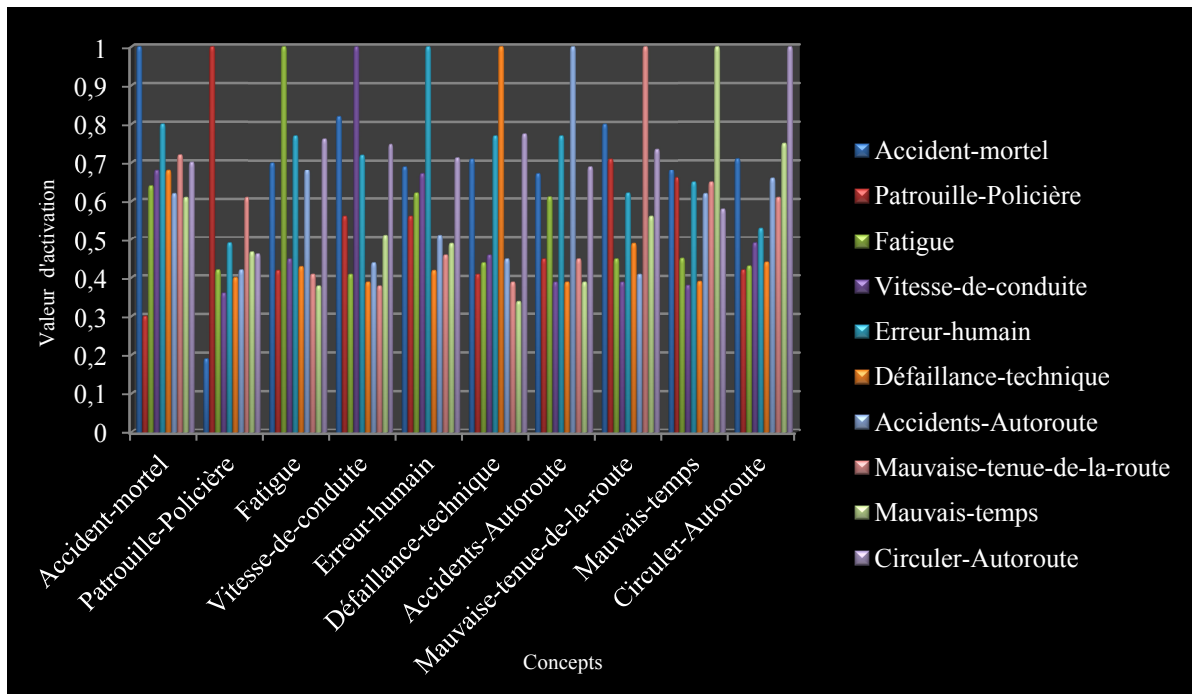


Figure V.11. Résultat de l'exécution sous forme d'histogramme

Si nous voulons surligner sur la première expérience (cas d'activer le premier concept C1= Accident-mortel). Donc, nous obtenons les valeurs suivantes :

$$(C1, C2, C3, C4, C5, C6, C7, C8, C9, C10) = (1, 0.3, 0.64, 0.68, 0.8, 0.68, 0.62, 0.72, 0.61, 0.701).$$

$$S((C1, C2, C3, C4, C5, C6, C7, C8, C9, C10)) = (1, 0, 1, 1, 1, 1, 1, 1, 1, 1).$$

Sachant que :

- | | |
|-----------------------|----------------------------|
| C1= Accident-mortel ; | C2= Patrouille-Policière; |
| C3= Fatigue; | C4= Vitesse-de-conduite; |
| C5= Erreur-humain; | C6= Défaillance-Technique; |

C7= Accidents-autoroute; C8= Mauvaise-tenue-de-la-route;
 C9= Mauvais-temps; C10= Circuler-autoroute.

Les valeurs sont interprétées comme suit : « si on active le concept « accident-mortel» alors on obtient presque une augmentation de tous les autres concepts ».

Selon les différents résultats obtenus, nous pouvons dire que les principales causes de notre problème sont l'erreur humaine, le véhicule, la météo et l'état de la route. Nous pouvons décider que pour réduire le nombre d'accidents mortels, par exemple, le responsable de la sécurité routière doit décider d'augmenter le nombre de patrouilles de la police qui peuvent réduire la vitesse de conduite et contrôler également l'état conducteur (fatigué ou alcool, ...) et l'état du véhicule. .

V.6. Validation

Il n'y a pas eu de travaux dans la littérature qui combinent entre les SIC, les CCF, les SMAH et la conception d'ontologie. Le tableau suivant (tableau 1) montre une comparaison avec deux approches les plus proches de notre travail:

| Critère Approche | Outils utilisés | Forme du problème | Paradigmes utilisés + Objectif | Résultat d'exécution |
|--|--|--------------------------|--|---|
| F. M. Avila [Avila, F.M., 2002] | Borland® JBuilder™6.0 environnement Java | Numérique | Cartes cognitives: pour gérer la prise de décision d'un seul agent. SMA: proposant le SMA sans l'utiliser. | Une solution numérique qui peut contenir l'indéterminisme. |
| B. Nacetet A. Adla [Nacet, B., Adla, A., 2014] | Jade : Une plate-forme de développement du SMA. | Textuel | SMA: utilisez les agents pour modéliser et développer un système collaboratif de prise à la décision. | Une solution choisie parmi plusieurs sans montrer leurs fonctions d'évaluation. |
| DCSic-CFSH | AgentBuilder : Une plate-forme de développement du SMAH. | Numérique | SIC: Contexte de travail. CCF: Faciliter, améliorer le processus de prise de décision en contexte coopératif. SMAH: Pour modéliser et mettre en œuvre notre | Une décision collective qui coordonnera toutes les décisions partielles afin de trouver la meilleure solution (le |

| | | | | |
|--|--|--|--|--|
| | | | système. Ontologie: enrichir sémantiquement notre système. | problème de la distribution réduit sa complexité). |
|--|--|--|--|--|

Tableau V.1. Comparaison de DCSic-CFSH avec deux autres approches

Dans la première approche, l'utilisation du SMA n'est pas réelle, mais elle gère les cartes cognitives, dont leur utilisation entraîne la création de l'indéterminisme lors de leur combinaison. Mais dans notre travail, nous avons utilisé les CCF pour éviter l'indéterminisme et aussi l'ambivalence. Dans la deuxième approche, chaque fabricant donne sa solution proposée sous une forme textuelle, ce qui rend difficile l'interprétation dans le langage machine. Il n'existe aucune méthode d'évaluation des solutions données pour choisir la meilleure. Mais dans notre système, les décisions partielles sont données sous la forme de CCF (plus spécifiquement la forme matricielle), et lors de la discussion pour l'obtention de la décision finale, la fusion de ces cartes est réalisée en fonction du coefficient d'expertise (CE) et coefficient d'émergence (CEm) de différents agents (voir le détails dans Chapitre IV.5). Cela rend la décision collective plus précise et plus réaliste.

Par conséquent, l'objectif principal de DCSic-CFSH est de trouver les meilleures solutions pour résoudre tout problème distribué donné. Ces solutions sont le résultat de la coopération de tous les agents responsables en utilisant l'outil CCF pour faciliter, améliorer le processus de prise de décision et traiter les problèmes complexes liés à la gestion des connaissances, aussi en utilisant SMAH pour modéliser et mettre en œuvre notre système, et enfin la conception d'ontologie pour l'enrichir sémantiquement.

Conclusion générale

Dans cette thèse, nous avons essayé de tirer profit de l'utilisation des CCF dans les SIC comme une alternative pour représenter les connaissances de manière qualitative, gérer, améliorer et éliminer les incohérences de la prise de décision collective inter-SIC.

La notion de CCF a existé depuis longtemps. Dans notre contexte (SIC), c'est un modèle pour analyser, prédire et comprendre comment les SIs prennent leurs décisions. Généralement, les CCF sont une représentation graphique des croyances d'une personne, d'un agent ou d'un domaine particulier. Formellement, les croyances d'un SI par rapport à un sujet sont représentées dans ce genre de cartes comme un graphe dirigé avec des concepts et des relations causales liant ces concepts entre eux. Les SIs utilisant les CCF profitent de ce modèle pour représenter d'une manière réaliste leurs vues partielles relatives à un domaine donné, dans le but de coopérer entre eux afin de prendre les meilleurs décisions.

DCSic-CFSH est le fruit de quatre paradigmes: SIC, SMAH, CCF et le concept d'ontologie. Dans notre architecture, nous avons proposé deux principaux SMA ; Le premier est un système holonique qui représente le SIC global. Les agents holoniques des différents niveaux de DCSic-CFSH lesquels interagissent les uns avec les autres en utilisant les opérations d'union ou d'intersection sur leurs propres CCF pour obtenir la CCF collective. Cette carte résultante est ensuite transmise à un deuxième SMA composé d'un groupe d'agents analyseurs pour exécuter en parallèle le processus d'analyse et d'inférence afin de parvenir à une décision collective pour résoudre le problème donné.

Ainsi, l'architecture du SMA proposée contient deux structures organisationnelles, l'une relative à chacun des SI impliqués dans la résolution collective du problème posé et l'autre du l'agent analyseur. L'implémentation du système a été réalisée sous la plateforme AgentBuidier, puisqu'elle permet de visualiser en temps réels les messages échangés au sein de SMA. Ces derniers, peuvent être archivés ce qui permet de disposer d'une mémoire structurée des échanges entre les agents holoniques, d'exécuter et de construire les différents ontologies et protocoles de communication.

Conclusion Générale

Pour mettre en œuvre DCSic-CFSH, le domaine de la sécurité routière est choisi comme un exemple d'application. Prévenir les accidents de la route nécessite l'intervention de spécialistes de différentes disciplines, ce qui nécessite une coopération et un véritable travail d'équipe de la part de toutes les autorités concernées. Notre architecture est donc utilisée pour recevoir les différents points de vue (décisions partielles) sous forme de CCF, et les intégrer pour obtenir des solutions typiques et pratiques au problème.

L'implémentation de DCSic-CFSH nous a permis de conclure que:

- ✓ L'utilisation des opérations d'union et d'intersection de Kosko entre les agents augmentent leur base de connaissances et renforcent les valeurs des relations causales dans la CCF résultante pour s'assurer qu'ils sont dans la bonne voie.
- ✓ l'intégration d'ontologie de domaine à notre architecture minimise le nombre de concepts générés dans la CCF résultante et permet de l'enrichir sémantiquement.
- ✓ Les cartes cognitives floues (CCF) offrent l'avantage de réduire le temps d'exécution, de minimiser le taux d'erreur et d'éliminer l'indéterminisme et l'ambivalence.
- ✓ L'utilisation de la plate-forme AgentBuilder offre l'avantage, à travers l'APIJava, de faciliter le développement des SMAH, la construction d'ontologie et du protocole d'interaction.
- ✓ En outre, l'exécution du processus d'analyse et d'inférence par un groupe d'agents analyseurs (pour générer en parallèle tous les comportements typiques) améliore les performances du système en minimisant le coût en termes de temps et d'espace mémoire.

Comme dans tout travail de recherche, il est naturel que les résultats conduisent à d'autres travaux futurs dans plusieurs directions, donc nous prévoyons d'étendre notre architecture pour effectuer les tâches suivantes:

- ◆ Dans le troisième niveau de notre architecture, la décision partielle de chaque SI (sous la forme d'une CCF) est entrée manuellement. En d'autres termes, chaque remplissage de la matrice est effectué directement par le SI, ce qui consomme beaucoup de temps (surtout dans le cas d'une CCF qui contient des milliers de concepts). Nous proposons d'assigner un programme initial permettant aux SI de dessiner les CCF et de remplir automatiquement les matrices associées.
- ◆ Les solutions les plus intéressantes de notre architecture sont trouvées par la dynamique de la CCF collective, autrement dit, par l'exécution des processus d'analyse et d'inférence. Les résultats sont donc sous forme numérique. Il est très difficile pour les personnes qui ne connaissent pas les CCF de les interpréter. Nous proposons

d'ajouter à notre architecture un outil permettant l'interprétation des résultats obtenus en langage naturel. Cela aidera les utilisateurs qui ne connaissent pas les CCF à comprendre les résultats et à rendre l'architecture proposée plus transparente pour les non-spécialistes.

- ◆ Intégrer le concept de négociation dans le fonctionnement de notre architecture. Par exemple, si nous avons un conflit sérieux, cela nécessite une discussion entre experts du domaine en dehors de notre architecture (directe ou distante) pour choisir la meilleure décision et éliminer les conflits et les contradictions. Par conséquent, nous avons proposé comme futur travail d'ajouter un nouveau mécanisme permettant aux agents de SIC d'être plus négociables pour trouver la meilleure solution.
- ◆ Affecter une fonction d'apprentissage à l'architecture qui ajoute un « aide-mémoire » pour identifier tout problème distribué et suggérer les solutions possibles.
- ◆ Illustrer la faisabilité de notre proposition par l'application sur des problèmes réels pour améliorer le processus décisionnel et pour trouver des solutions à des problèmes complexes par exemple dans des infrastructures telles que le Cloud Computing. Ce dernier fournit un modèle innovant qui permet aux entreprises de réduire les coûts opérationnels et d'améliorer la flexibilité et l'évolutivité.

Bibliographie

- [Acampora, G., Loia, V., 2009]: Acampora, G., Loia, V.: A Dynamical Cognitive Multi-Agent System for Enhancing Ambient Intelligence Scenarios. IEEE Inter Confer on Fuzzy Systems, N. 5277303, pp. 770-777. (2009).
- [Adla, A., 2010] : Adla, A.: Aide à la Facilitation pour une prise de Décision Collective : Proposition d'un Modèle et d'un Outil, Thèse de Doctorat. Faculté des sciences. Université de Toulouse. France. (2010).
- [AgentBuilder, 2016] : AgentBuilder plateforme. [En ligne] <http://www.agentbuilder.com/> (consulté en mars 2016).
- [Arabani, A.B. et al., 2009]: Arabani, A.B., Ramtin, F., Rafienejad, S.N.: Applying simulated annealing algorithm for crossdocking scheduling. In: Proceedings of the World Congress on Engineering and Computer Science, San Francisco, USA, vol. 2, (2009).
- [Arens, Y. et al., 1993]: Arens, Y., Chee, C.Y., Hsu, C., Knoblock, C.A.: Retrieving and integrating data from multiple information sources, International Journal of Intelligent and Cooperative Information Systems, Vol.2, N.2, p. 127-157, (1993).
- [Atohou, B.C., 2013] : Atohou, B. C. : Architecture logique d'un système multi agents de suivi multicaméra distribué : exploitation du Modèle de Croyance Transférable, Thèse en vue de l'obtention du Doctorat en Informatique, Université du Littoral Coté d'Opale, (03 décembre 2013).
- [Avila, F.M., 2002] : Avila, F. M. : Raisonnement qualitatif dans les systèmes multi agents basé sur les cartes causales, Thèse de doctorat non publiée, Université de Laval, Canada. (Aout 2002).
- [Axelrod, R., 1976]: Axelrod, R.: Structure of Decision: the Cognitive Maps of Political Elites. NJ: Princeton University Press, (1976).
- [Baghdadi, Y., 1997]: Baghdadi, Y.: Contribution Méthodologique à la Conception des Systèmes d'Information Coopératifs. Thèse de l'université de Toulouse I, (11 Mars 1997).
- [Bayardo, R. et al., 1997]: Bayardo, r., Bohrer, W., Brince, R., Cichocki, A., Fowler, G., Helal, A., Kashyap, V., Ksiezyk, T., Martin, G., Nodine, M., Rashid, M., Rusinkiewicz., Shea, R., Unnikrishnan, C., Uniruh, A., Woelk, D. : InfoSleuth : Agent-Based Semantic Integration of Information in Open and Dynamic Environment, Proceedings of the ACM SIGMOD Intenational Conference on Management of Data, Tucson, Arizona, USA, p. 195-206, (Mai 1997).

- [Barthélemy, J.P., 1997] : Barthélemy, J.P.: Le sauvetage de l'âne - introduction aux sciences et techniques de la décision. (Document de cours), pp. 28, 39 et 64, (1997).
- [Beheshti, R. et al., 2016]: Beheshti, R., Barmaki, R., and Mozayani, N.: Negotiations in Holonic Multi-agent Systems, Springer International Publishing Switzerland 2016, pp.107-117. (2016).
- [Bolc, L., Jarke, M., 1986]: Bolc, L., Jarke, M.: Cooperative Interfaces to Information Systems. Springer-Verlag, Berlin, Heidelberg, Allemagne. (1986).
- [Boulanger, D., 1991]: Boulanger, D. : Coopération de systèmes d'information, Dossier scientifique pour l'habilitation à diriger des recherches, Université Jean Moulin, Lyon, (13 Mars 1991).
- [Boulanger, D., Dubois, G., 1997]: Boulanger, D., Dubois, G. : Objets et coopération de systèmes d'information, chapitre 10 du livre Ingénierie Objet: Concepts et techniques, InterEditions, (mai 1997).
- [Boulanger, D., Dubois, G., 2001]: Boulanger, D., Dubois, G.: Advanced Object Approach for Information System Cooperation, dans Allen Kent & James Williams Executive Editeurs, Encyclopedia of Microcomputers, New-york, Marcel Dekker Inc., vol.27, N°6, p. 15-35, (2001).
- [Brazier, F.M.T. et al., 1997]: Brazier, F.M.T., Dinin-kepliczb, M., Jennings, N.R., Trueur, J.: DESIRE: Modeling multi-agent systems, in a Compositional formal framework, International Journal of Cooperative Information Systems, 6(1), p. 64-94, (1997).
- [Briot, J.P., Demazeau, Y., 2001]: Briot, J.P., Demazeau, Y.: Introduction aux agents, In principes et architectures des systemes multi-agents, Jean-Pierre Briot et Yves Demazeau éditions, Collection IC2, Hermès Science Publications, Paris, France, (Décembre 2001).
- [Brodie, M., Ceri, S., 1992]: Brodie, M., Ceri, S. : On intelligent and cooperative information systems, International Journal of Intelligent and Cooperative Information Systems, 1(2), p. 1-35, (Septembre 1992).
- [Brooks, R.A. 1991]: Brooks R.A.: Intelligence Without Reason. Proceeding of the 12th International Joint Conference on Artificial Intelligence, IJCAI'91, Ed. by Myopoulos J.& Reiter R., Sydney, Australie, pp. 569 – 595, (Août 1991).
- [Cattell, R. et al., 1993]: Cattell, R., Barry, D., Bartels, D., Eds.: The object Database Standard: ODMG 2.0, Morgan Kaufmann, (1993).

- [Chauvin, L., 2010] : Chauvin, L. : Modèles De Cartes Cognitives Etendues Aux Notions De Contexte Et D'échelle, Thèse pour l'obtention de grade de Docteur, Faculté de science. Université d'Angers. pp.116-1170. (2010).
- [Chawathe, S. et al., 1994] : Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Uliman, J., Widom, J.: The TSIMMIS Project : Integration of Heterogeneous Information Sources, Proceedings of the 16th Meeting of the Information Proceeding Society of Japan, Tokyo, Japan, pp.7-18, (Octobre 1994).
- [Coakley, S. et al, 2012]: Coakley, S., Gheorghe, M., Holcombe, M., Chin, S., Worth, D. et Greenough, C.: Exploitation of high performance computing in the flame agent-based simulation framework. In Proceedings of the 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, HPCC '12, HPCC '12, pp. 538–545, Washington, DC, USA. IEEE Computer Society. (2012).
- [Coleman, D. et al., 1994]: Coleman, D., Arnold, P., Bodoff, S., Dollin, C., Gilchrist, H., Hayes, F., Jeremaes, P. : Object-oriented development : the fusion method, Prentice Hall, (1994).
- [Collier, N. et North, M., 2011]: Collier, N. et North, M.: RepastHPC : A platform for large-scale agentbased modeling. Wiley, (2011).
- [Couturier, V., 2004]: Couturier, V. : L'ingénierie des systèmes d'information coopératifs par réutilisation : une approche à base des patterns, Thèse présentée pour l'obtention de grade de docteur en Informatique, Université Jean Moulin Lyon 3, (Décembre 2004).
- [Cordasco, G. et al., 2011]: Cordasco, G., Chiara, R., Mancuso, A., Mazzeo, D., Scarano, V. et Spagnuolo, C.: A Framework for Distributing Agent-Based Simulations. In Euro-Par 2011 : Parallel Processing Workshops, Lecture Notes in Computer Science, Vol 7155, pp. 460–470, (2011).
- [De Michelis, G. et al., 1994]: De Michelis, G., Dubois, E., Jarke, M., Matthes, F., Mylopoulos, J., Pohl, K., Schmidt, J., Woo, C., and Yu, E.: Cooperative Information Systems. Presented at the Second International Conference on Cooperative Information Systems (CoopIS'94), Toronto, Canada, Mai 17-20, (1994).
- [De Michelis, G. et al., 1996]: De Michelis G., Dubois E., Jarke M., Matthes F., Mylopoulos J., Pohl K., Schmidt J., Woo C., and Yu E.: Cooperative Information Systems- A Manifesto. Workshop on Cooperative Information Systems, Brussels, Belgium, (Avril 1996).

- [De Michelis, G. et al, 1997]: De Michelis, G., Dubois, E., Jarke, M., Matthes, F., Mylopoulos, J., Papazoglou, M., Pohl, K., Schmidt, J., Woo, C., Yu, E. : Cooperative Information Systems: A Manifesto. Cooperative Information Systems: Trends & Directions, Mike P. Papazoglou and Gunter Schlageter (eds), Academic-Press, (1997).
- [Deloach, S. A, Wood, M., 2001]: Deloach, S. A., Wood, M.: Developing Multiagent Systems with agentTool, (ATAL'2000), Berlin, (2001).
- [Dickerson, J. A., Kosko, B., 1994]: Dickerson, J. A., Kosko, B.: Virtual Worlds As Fuzzy Cognitive Maps, Université de Southern California, Los Angeles, California, http://sipi.usc.edu/~kosko/virtual_worlds_fcm.pdf , (Accédé 2013). (1994).
- [Demazeau, Y., 1997] : Demazeau, Y. : Steps towards multi-agent oriented programming. In Jst International Workshop on Multi Agent Systems, IWMAS'97, Boston, (1997).
- [Dotoli, M. et al., 2017] : Dotoli, M., Zgaya, H., Russo, C, . : A Multi-Agent Advanced Traveler Information System for Optimal Trip Planning in a Co-Modal Framework, IEEE Transactions on Intelligent Transportation Systems, Vol. 18, N. 9, pp. 2397 – 2412, (Septembre 2017).
- [Dubois, G. et al., 2002]: Dubois, G., Boulanger, D., Couturier, V.: Coopération d'objets pour la résolution de requêtes multi bases, In INFORSID 2002 , Nantes, pp. 199-214, (1-4 Juin 2002).
- [Ferber, J., 1995] : Ferber, J.: Les systèmes multi-agents, vers une intelligence collective. IIA, Paris : InterEditions, (1995).
- [Ferber, J., 1999]: Ferber,J.: Multi-Agent Systems : an introduction to distributed artificial intelligence, Addison Wesley, London, (1999).
- [Ferber, J. et al., 2003] : Ferber, J., Gutknecht, O., Michel, F. : From agents to organizations : an organozational view of multi-agent systems, P. Giorgini, J.P. Muller et J.Odell, éditeurs : Agent-Oriented Software Engineering IV, 4th International Workshop, AOSE 2003, Melbourne, Australia, (Juillet 2003).
- [FIPA, 2002]: « FIPA ACL Message Structure Specification », <http://www.fipa.org/specs/fipa00061/index.html> , (2002).
- [Fleyeh, H., Davami, E., 2013]: Fleyeh, H., Davami, E.: Multiclass adaboost based on an ensemble of binary adaboosts. Am. J. Intell. Syst. **3**(2), pp.57–70, (2013).
- [Frendi, M ., 2013]: Frendi, M .: Modélisation des systèmes d'information, Thèse présentée pour l'obtention de grade de magister en système d'information, Université d'Oran, Oran, Algérie, (2013).

- [Gadallah, A. M. et al., 2010]: Gadallah, A. M., Hesham, A., and Hefny, H. A.: A Novel Multi agent System Based on Dynamic Fuzzy Cognitive Map Approach. Intelligent Systems Design and Applications (ISDA), (2010).
- [Gasser, L., 1990]: Gasser, L. : Social conception of knowledge and action, Technical report ACT-IA-355-90, MCC, (Octobre 1990).
- [Garcia-Molina, H. et al., 1997]: GarciaMolina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Uliman, J., Vassalos, V., Widom, J.: The TSIMMIS approach to mediation : Data models and Languages, In Journal of Intelligent Information Systems, Vol.8, p. 117-132, (1997).
- [Geen, S. et al., 1997]: Geen, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F., Evans, R.: Software Agents: A review. Rapport technique, Trinity College Dublin, (1997).
- [Genesereth, M.R., 1997]: Genesereth, M.R.: An Agent-based Framework for Interoperability, In Software Agents, J.M. Bradshaw (editor), AAAI Press/The MIT Press, Menlo Park, California, (1997).
- [Giabbanelli, P.J. et al., 2017]: Giabbanelli, P.J., Gray, S. A., Aminpour, P.: Combining fuzzy cognitive maps with agent-based modeling: Frameworks and pitfalls of a powerful hybrid modeling approach to understand human-environment interactions, Environmental Modelling & Software, Vol. 95, pp. 320-325, (September 2017)
- [Gleizes, M.P. et al., 2008] : Gleizes, M.P., Bernon, C., Migeon, F., Picard, G. : Méthodes de développement de systèmes multi-agents, Génie logiciel, N°86, (2008).
- [Goh, C., 1997]: Goh, C.: Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems, Thèse pour obtenir le grade de docteur, MIT Sloan School of Management, USA, (1997).
- [Goh, C. et al., 1994]: Goh, C., Madnick, S., Siegel, M. : Context interchange : overcoming the challenges of large-scale interoperable database systems in a dynamic environment, In Proceedings of the Third International Conference on Information and Knowledge Management, ACM Press, Gaithersburg, Maryland, USA, p.337-346, (Décembre 1994).
- [Gouiza, F., 2016] : Gouiza, F.: Modélisation et évaluation des performances de la chaîne de transport intermodal de porte à porte : le cas du corridor de la Vallée de Seine, Thèse en vue de l'obtention du Doctorat en Génie informatique, automatique et traitement de signal, Université du Havre, Université de Toulouse III - Paul Sabatier, (8 mars 2016).

- [Graja, Z., 2015] : Graja, Z.: Vérification formelle des systèmes multi-agents auto-adaptatifs, Thèse en vue de l'obtention du Doctorat en Informatique, Université de Toulouse III, (15 Septembre 2015).
- [Gutknecht, O., 2001] : Gutknecht, O. : Proposition d'un modèle organisationnel générique de systèmes multi-agents examen de ces conséquences formelles, Implémentatoires et méthodologiques, Thèse de doctorat, Université de Montpellier II, France, (2001).
- [Hengjie, S. et al., 2008]: Hengjie, S., Chunyan, M., Zhiqi, S., and Yuan, M.: Fuzzy Cognitive Map Learning Based on Multi-Objective PSO. International Journal of Computational Cognition, Vol. 6, No.3. (Septembre 2008).
- [Herin, D. et al., 2001]: Herin, D., Espinasse, B., Andonoff, E., Hannachi, C.: Des systèmes d'information coopératifs aux agents informationnels, Chapitre 8 du livre « Ingénierie des Systèmes d'information » Hermès, (2001).
- [Hicks, J., 1993]: Hicks, J.: Management Information Systems :Auser perspective, Wes Publishing Company, St. Paul, (1993).
- [Holtzman, S., 1989]: Holtzman, S.: Intelligent decision systems reading, Massachusetts, Addison-wesley, (1989).
- [Howard, R., Matheson, J., 1984]: Howard, R. Matheson, J. : Influence Diagrams, Readings on the Principles and Applications of Decision Analysis, pp. 721-762, (1984).
- [Iakovidis, D.K., 2011]: Iakovidis, D.K.: Intuitionistic Fuzzy Cognitive Maps for Medical Decision Making, IEEE Transactions On Information Technology In Biomedicine, Vol. 15, N. 1. (Janvier 2011).
- [Iglesias, C.A. et al., 1998]: Iglesias, C.A., Garijo, M., Gonzalez, J.C., Velasco, J.R. : Analysis and design of multiagent systems using MAS-CommonKADS, In M. P. Singh, A. S. Rao, et M. Wooldridge (Eds.), Proceedings of the 4th International Workshop on Intelligent Agents, Agent Theories, Architectures and Languages, LNAI 1365, pp.313-326, (1998).
- [Jade, 2014] : Jade plateforme. [En ligne] <http://jade.tilab.com/>(consulté en 2014).
- [Jarras, I., Chaib-draa, B., 2002] : Jarras, I., Chaib-draa, B.: Aperçu sur les systèmes multi-agents, Série scientifique, Centre interuniversitaire de recherche en analyse des organisations, Montréal, Canada, (2002).
- [Jérémy, B., 2014] : Jérémy, B. : Apprentissage du contrôle de systèmes complexes par l'auto-organisation coopérative d'un système multi-agent : application à la calibration de

- moteurs à combustion, Thèse en vue de l'obtention du Doctorat en Informatique, Université de Toulouse III, (28 mars 2014).
- [Jennings, N. et al., 1998]: Jennings, N., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems* 1, pp.7–38, (1998).
- [Jing, L. et al., 2016] : Jing, L., Yaxiong, C., Chen, Z.: A Dynamic Multiagent Genetic Algorithm for Gene Regulatory Network Reconstruction Based on Fuzzy Cognitive Maps, *IEEE Transactions on Fuzzy Systems* , Vol. 24, N.2, pp. 419 – 431, (Avril 2016).
- [Jouanot, K., 2000]: Jouanot, K. : un modele sémantique pour l'interopérabilité de systèmes d'information, In *INFORSID 2000*, Lyon, p.347-364, (16-19 Mai 2000).
- [Jouanot, F., 2001]: Jouanot, F. : DILEMMA : vers une coopération de systèmes d'information basée sur la médiation sémantique et la fusion d'objets, Thèse de doctorat, Université de Bourgogne, France, (Novembre 2001).
- [Jung, J., Geun-Sik, J., 2000]: Jung, J., Geun-Sik, J.: CRMMaps: Managing Customer Relationships Based On Ontological Fuzzy Cognitive Maps, *Intelligent Ecommerce systems laboratory, School of Computer Science and Engineering Inha University, Incheon Korea.*(2000).
- [Kamoun, M.A. et al., 2005]: Kamoun , M. A., Uster, G., and Hammadi, S.: An Agent-Based Cooperative Information System for Multi-modal Travelers Route Computation, *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, pp.1162 – 1167. (2005).
- [Kenfack_Dongmo, C.V., 2010] : Kenfack Dongmo, C.V. : Un Système multi-agents comme outil d'intermédiation pour les Communautés de Pratiques. Thèse de doctorat en Informatique, Université Jean Moulin Lyon 3, (4 Juin 2010).
- [Kessi, K. et al., 2015]: Kessi, K., Alimazighi, Z., Oussalah, M.: Requirement Meta Model of a Cooperative Information System Oriented Viewpoints. *Procedia Computer Science*. Vol. 64, pp. 474–482. (2015).
- [Kim, W. et al., 1994]: Kim, W., Choi, I., Gala, S., Scheevel, M.: On resolving schematic heterogeneity in multidatabase systems, In *Modern database systems : the object model, interoperability, and beyond*, Addison-Wesley, pp. 649-663, (1994).
- [Kim, W., Seo, J., 1991]: Kim, W., Seo, J.: Classifying schematic and data heterogeneity multidatabase systems, *Journal of the ACM*, vol. 24, p.12-18, (1991).

- [Kinny, D. et al, 1996] : Kinny, D., Georgeff, M., Rao, A., A methodology and modelling technique for systems of BDI agents, W. Van de VELDE et J. W. PERRAM, éditeurs : Agents Breaking Away : Proceedings of the Seventh European Workshop on Modeling Autonomous Agents in Multiagent World, Vol.1038 de Lecture Notes in Artificial Intelligence (LNAI), pp. 51-71. Springer-Verlag, (1996).
- [Koestler, A., 1968]: Koestler, A.: The ghost in the machine. Psychiatr. Commun. **10**(2), p.45 (1968).
- [Kosko, B., 1986]: Kosko, B.: Fuzzy Cognitive Maps, International Journal of Man Machine Studies, Vol 24, N. 1, pp.65-75. (Janvier 1986).
- [Kosko, B., 1992]: Kosko, B.: Neural Networks And Fuzzy Systems: A Dynamical Systems Approach ToMachine Intelligence, University of Southern California, <https://fr.scribd.com/doc/163341141/Neural-Networks-and-Fuzzy-Systems-A-DynamicalSystems-Approach-to-Machine-Intelligence>. (Consulté en Juin 2013), (1992).
- [Laroche, H., Nioche, J.P. 1994]: Laroche, H.,Nioche, J.P. : L'approche cognitive de la stratégie d'entreprise. RFG, pp. 64-78, (1994).
- [Lee, N. et al., 2011]: Lee, N., Bae, J.K., Koo, C.: An agent-based cognitive mapping system for sales opportunity analysis, Expert Systems with Applications, Vol. 38, N. 6, pp.7016–7028. (Juin 2011).
- [Lee, K.C. et al., 2013]: Lee, K.C., Lee, H., Lee, N., Lim, J.: An agent-based fuzzy cognitive map approach tothe strategic marketing planning for industrial firms. Industrial Marketing Management, Vol. 42, pp.552–563. (2013).
- [Lévine, P., Pomerol, J., 1989] : Lévine, P. et Pomerol, J. : Systèmes interactifs d'aide à la décision et systemes experts, Editions Hermés, (1989).
- [Libourel, Th. et al., 2003]: Libourel Th. Et al.: Médiation via les méta-données, Synthèse de l'action Spécifique 97 du department STIC du CNRS, RTP 9, sous l'animation de Th.Libourel, <http://www.lirmm.fr/%7Elibourel/MM/RA-AS.doc>, (2003).
- [Liu, L., Lu, C., 1995]: Liu, L., Lu, C.: The distributed interoperable object model and its application to large -scale interoperable database systems, International Conference on Information and Knowledge Management, (1995).
- [Liu, L., Pu, C., 1995]: Liu, L., Pu, C.: The Diom approach to large-scale interoperable information systems, Technical report, TR95-16, Department of Computing Science, University of Alberta, Edmonton, Albert, (Mars 1995).

- [Ludascher, B. et al., 1999]: Ludascher, B., Papakonstantinou, Y., Velikhov, P.: A Framework for Navigation-Driven Lazy Mediators, Proceedings of WebDB'99, pp. 85-90, (Juin 1999).
- [Madkit, 2018]: Madkit plateforme. [En ligne] <http://www.madkit.org/>. (consulté en Décembre 2018).
- [Magique, 2015]: Magique plateforme. [En ligne] <http://www2.lifl.fr/SMAC/projects/magique> (consulté en Janvier 2015).
- [Mathieu, L., 2017]: Mathieu, L.: La gestion de la connaissance des équipes de développement logiciel, Thèse de Doctorat. Département d'ingénierie informatique et d'ingénierie logicielle. École polytechnique de Montréal, Université de Montréal. (2017).
- [Maziz, A., Zarour, N., 2014]: Maziz, A., Zarour, N.: A Fuzzy Cognitive Maps-based Collective Decision in Cooperative Information Systems. Papier présenté aux 4^{ème} Journées Doctorales en Informatique, Guelma, Algérie. (3-4 Decembre 2014).
- [Maziz, A., Zarour, N., 2015]: Maziz, A., Zarour, N.: Improving the process of decision making in the CIS: an approach based on fuzzy cognitive maps and holonic systems'. Papier présenté au 8th National Seminar in Computer Science. Biskra, Algérie. (20-21 Janvier 2015).
- [Maziz, A., Zarour, N., in press]: Maziz, A., Zarour, N.: A novel architecture based on fuzzy cognitive maps and holonic systems for decision making in a cooperative context, Int. J. Information and Decision Sciences, Vol. X, No. Y, pp.xxx–xxx. (in press).
- [Mazyad, H., 2013]: Mazyade, H.: Une approche multi-agents à architecture P2P pour l'apprentissage collaboratif, Thèse pour l'obtention le grade de docteur en Informatique, Université du Littoral Côte d'opale, (2013).
- [Meghabghab, G., 2001]: Meghabghab, G.: Fuzzy cognitive state maps VS. Markovian modeling of user's web behavior, IEEE International Conference on Systems, Man and Cybernetics, Tuscon, AZ, pp.1167-1172.(2001).
- [Mei, S. et al., 2014]: Mei, S., Zhu, Y., Qiu, X., Zhou, X., Zu, Z., Boukhanovsky, A.V., and Sloat, P. M. A.: Individual Decision Making Can Drive Epidemics: A Fuzzy Cognitive Map Study. IEEE Transactions On Fuzzy Systems, Vol. 22. (Avril 2014).
- [Mena, E., 1999]: Mena, E.: OBSERVER : An approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies, Thèse pour obtenir le grade de docteur, Université de Zaragoza, Spain, (Février 1999).

- [Miao, Y. et al, 2001] : Miao, Y. Liu, Z.Q. Siew, C.K. , Miao, C.Y. : Dynamical cognitive network - an extension of fuzzy cognitive map, *Fuzzy Systems, IEEE Transactions on*, Vol. 9, , pp 760-770. (2001).
- [Michael, H., 1997]: Michael, H.: Generations ahead: Michael Huhns on cooperative information systems. *IEEE EXPERT: Intelligent Systems & Their Applications*. (1997).
- [Mingchua, Z. et al., 2012]: Mingchua, Z. Haixia, Z., Ruijuan, Z. and Qingtao, W.: A Novel Multi-ANs Cooperative Decision-making Mechanism for Cognitive Network, *Fourth International Symposium on Information Science and Engineering, IEEE*, pp. 189-192. (2012)
- [Mora et al., 2014]: Mora, M., Phillips-Wren, G., Marx-Gomez, J., Wang, F., and Gelman, O.: The role of decision-making support systems in IT service management processes, *Intelligent Decision Technologies*, Vol.8. (Janvier 2014).
- [Mullet, E. et al, 1996] : Mullet, E., Barthélemy, J.P., Duponchelle, L., Sastre, M. T. Muñoz, & Neto, F.: Décision, choix, jugement, orientation. *L'orientation scolaire et professionnelle*, pp. 169–192, (1996).
- [Munan, L., Fen, H., 2016] : Munan, L., Fen, H. : Formal describing the organizations in the pervasive healthcare information system: Multi-agent system perspective, *Advanced Robotics and Mechatronics (ICARM), International Conference on Macau, China, IEEE*, (27 October 2016).
- [Nachet, B., 2014] : Nachet, B. : Modèle multi-agents pour la conception de systèmes d'aide à la décision collective, Thèse pour l'obtention le grade de docteur en Informatique, Université d'Oran, Algérie. (2014).
- [Nachet, B., Adla, A., 2014] : Nachet, B., and Adla, A.: An agent-based distributed collaborative decision support system. *Intelligent Decision Technologies Journal*. Vol.8. pp.15-34. (Janvier 2014).
- [Nalchigar, S. et al., 2011]: Nalchigar, S., Nasserzadeh, S. M. R., Akhgar, B.: Simulating strategic information systems planning process using fuzzy cognitive map. *International Journal of Business Information Systems*, Vol. 8, N.3, pp. 286-306, (Septembre 2011).
- [Obernesser, C., 2003] : Obernesser, C. : Cartes cognitives pour la modélisation comportementale, Mémoire de DEA Sciences de cognition, Ecole doctorale des sciences de l'homme, Université Victor Segalan, Bordeaux 2, (23 juin 2003).
- [Ouahrani, L., Alimazighi, Z., 2009] : Leila, O., Zaia, A.: An Agent Based Model for Uniform Access to Resources in Cooperative Information Systems, *IEEE International Conference on Computational Intelligence, Modeling and Simulation, CSSim '09*, pp.126 – 131, (2009).

- [Ouzzani, M., Bouguettaya, B., 2004] : Ouzzani, M., Bouguettaya, B.: Query Processing and Optimization on the Web, *Distibuted and Parallel Databases Journal*, 15(3), Kluwer Academic Publishers, pp. 187-218, (Mai 2004).
- [Papageorgiou, E.I., 2012]: Papageorgiou, E.I.: Learning Algorithms for Fuzzy Cognitive Maps—A Review Study, *IEEE Transactions On Systems, Main, and Cybernetics-Part C*, Vol. 42, N.2, (Mars 2012).
- [Papazoglou, M.P. et al, 1992]: Papazoglou, M.P., Laufmann, S., Sellis, T.K.: An organizational framework for cooperating intelligent information systems, *International Journal of Cooperative Information Systems*, Vol. 1, N. 1, (Mars 1992).
- [Parenthoen M. et al., 2001] : Parenthoen, M., Tisseau, J., Reignier, P., et Dory, F. : Perception des Agents et Acteurs Artificiels dans les Mondes Virtuels : Les Cartes Cognitives Floues au Travail, in VRIC'01 proceedings *International Journal of Design and Innovation Research*, France. (2001).
- [Perusich, K., 1996]: Perusich, K.: Fuzzy cognitive maps for policy analysis. In *Proceedings in International Symposium on Technology and Society Technical Expertise and Public Decisions*, Vol. 10, pp. 369–373. (1996).
- [Picard, G., Gleizes, M.P., 2004] : Picard, G., Gleizes, M.P. : The ADELFE methodology- designing adaptive cooperative multi-agent systems, volume *Methodologies and Software Engineering for Agent Systems*, Chapitre 08, Kluwer, pp. 157-176, (2004).
- [Picard, G., 2004] : Picard, G. : Méthodologie de développement de multi-agents adaptatifs et conception de logiciels à fonctionnalité émergence, Thèse de doctorat, Université Paul Sabatier de Toulouse III, France, (2004).
- [Rao, A., Georgeff, M., 1995] : Rao, A., Georgeff, M.P. : BDI Agents: From Theory to Practice . In: Lesser, V. (Ed.): *Proc. of the 1st Int. Conference in MultiAgent Systems (ICMAS'95)*, IEEE Computer Society Press, San Francisco, pp. 312-319. (1995).
- [Rodin, V. et al., 2009]: Rodin, V., Querrec, G., Ballet, P., Bataille, F.R., Desmeulles, G., Abgrall, J.F., and Tisseau, J.: Multi-Agents System to model cell signalling by using Fuzzy Cognitive Maps.Application to computer simulation of Multiple Myeloma.2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering, pp.236-241. (2009).
- [Rolland, C. et al., 1988]: Rolland, C., Foucault O., Benci G. : Conception des systèmes d'information : la méthode REMONA, Eyrolles, (1988).

- [Roy, B., 2000] : Roy, B. : Réflexions sur le thème : quête de l'optimum et aide à la décision, Cahier du Lamsade N° 167, Université Paris-Dauphine, France, (2000).
- [Rusell, S., Norving, P., 1995]: Artificial Intelligence : a modern approach. Prentice Hall, USA. (1995).
- [Séguran, M., 2003] : Séguran, M. : Résolution Des Conflits Sémantiques Dans Les Systèmes D'information Coopératifs : Proposition D'un Modèle D'interaction Entre Agents, Jean Moulin University, Lyon3. (2003).
- [Scheuermann, P. et al, 1990]: Scheuermann, P., Elmagarmid, A., Garcia-Molina, H., Manola, F., McLeod, D., Rosenthal, A., Templeton, M. : Report on the workshop on heterogeneous database systems held at north west on university, ACM Sigmod Record, Vol. 19, pp. 23-31, (Décembre 1990).
- [Sciore, E. et al., 1994]: Sciore, E., Siegel, M., Rosenthal, A.: Using semantic values to facilitate interoperability among heterogeneous information systems, ACM Transactions on Database Systems, Vol. 19, p. 254-290, (Juin 1994).
- [Schneider, D., 1996] : Schneider, D. : Modélisation de la démarche du décideur politique dans la perspective de l'intelligence artificielle. Thèse présentée pour l'obtention du grade du Docteur, Faculté des sciences économiques et sociales, Université de Genève. (1996).
- [Sheth, A., 1999]: Sheth, A.: Changing focus on interoperability in information systems : from system, syntax, structure to semantics, In Interoperating Geographic Information Systems, M. F. Goodchild, M.J. Egenhofer, B.R. Fegeas and C.A. Koffman Editors, Kluwer Academic Publishers, (1999)
- [Sheth, A., Kashyap, V. , 1992]: Sheth, A., Kashyap, V. : So far (Schematically) yet so near (Semantically), in proceedings of the IFIP DS-5, Conference on Semantic of Interoperable Database Systems, Lorne, Australia, (Novembre 1992).
- [Sheth, A., Larson, J., 1990]: Sheth, A., Larson, J.: Federated database systems for managing distributed, heterogeneous and autonomous database, ACM computing Survey, Vol.22, N.3, p. 183-190, (1990).
- [Simon, H.A., 1957] : Simon, H.A.: A behavioural model of rational choice. In H.A. SIMON, Ed., Models of man: social and rational; mathematical essays on rational human behavior in a social setting, pp. 241-260. J. Wiley, New York, (1957).
- [Simeon, J., 1999]: Simeon, J.: Intégration de sources de données hétérogènes – ou comment marier simplicité et efficacité, Thèse de doctorat, Université Paris XI, (18 Janvier 1999).

- [Simon, H.A., 1977]: Simon, H.A.: The new science of management decision. Prentice Hall, New Jersey, Etats Unis. (1977).
- [Souici, I., 2016] : Souici, I. : Modélisation en Aide à la Décision, Support de court, Université de Jijel, Algérie, (Avril 20016).
- [Sousa, J.L.R. et al., 2015] : Sousa, J.L.R., _Banco Jr, T., and T._Amorim , B.S.R.: Modeling Information Systems Socio materiality and its Impact on Organizational Cooperative Environment Development. *Procedia Computer Science*. Vol. 64. pp.1140–1148. (2015).
- [Stula, M. et al., 2010]: Stula, M., Stipanicev, D., Bodrozcic, L.: Intelligent Modeling With Agent Based Fuzzy Cognitive Map, *International Journal of Intelligent Systems*. Vol 25, N. 10. (Octobre 2010).
- [Styblinski, M.A., Meyer, B. D., 1988]: Styblinski, M. A., Meyer B. D.: Fuzzy Cognitive Maps, Signal Flow Graphs and Qualitative Circuit Analysis, *Proceedings of the 2nd IEEE International Conference on Neural Networks (ICNN-87)*, pp.549-556, (Juillet 1988).
- [Stylios, C., Groumpos, P., 1998]: Stylios, C., Groumpos, P.: The Challenge Of Modeling Supervisory Systems Using Fuzzy Cognitive Maps. *Journal of Intelligent Manufacturing* [enligne], https://www.researchgate.net/publication/2362041_The_Challenge_of_Modelling_Supervisory_Systems_Using_Fuzzy_Cognitive_Maps, (Consulté en 2012). (1998).
- [Sycara, K. et al, 2010]: Sycara, K., Norman, T.J., Giampapa, J.A., Kollingbaum, M.J., Burnett, C., Masato, D. McCallum, M. and Strub, M. H.: Agent support for policy-driven collaborative mission planning. *Comput. J.*, p. 53 :528–540, (Juin 2010).
- [Taber, R., 1991]: Taber, R.: Knowledge processing with fuzzy cognitive maps. *Expert Systems with Applications*, Vol 2, pp. 83-87. (1991).
- [Tardieu, H. et al, 1994]: Tardieu, H., Rochfeld, A. et Coletti, R. : La Méthode Merise. Les Editions d'Organisation, Paris, (Juillet 1994).
- [Tolman, E., 1948]: Tolman, E.: Cognitive Maps in rats and men, *Psychological Review*, <http://psychclassics.yorku.ca/Tolman/Maps/maps.htm>, vol.55, N. 4, pp.189-208, (1948).
- [Tomá, N., 2015] : Tomá, N. : Optimization of Decision-Making in Artificial Life Model Based on Fuzzy Cognitive Maps, *Intelligent Environments (IE)*, 2015 International Conference , Prague, Czech Republic, IEEE, (13 Aout 2015).

- [Tomasic, A. et al, 1995]: Tomasic, A., Raschid, L., Valduriez, P.: Scaling heterogeneous databases and the design of Disco, Technical report, N° 2704, INRIA, Recquencourt, France, (1995).
- [TSA, 2017] : Tout sur l'Algérie (TSA). [en ligne] <https://www.tsa-algerie.com/accidents-de-la-route-alger-en-tete/> (Accédé le 16 Juillet 2017).
- [Uglev, V., 2014]: Uglev, V.: Implementation of Decision-making Methods in Intelligent Automated Educational System Focused on Complete Individualization in Learning, AASRI Procedia, Vol. 6, pp. 66 – 72. (2014).
- [Verharen, E., 1997]: Verharen, E. : A Language-Action Perspective on the Design of Cooperative Information Systems, Thèse présentée pour l'obtention de grade de docteur en Informatique, Université de Tilburg, (1997).
- [Vernadet, F., Azena, P., 1992] : Vernadet, F., Azena, P. : Prototypage de systèmes d'agents communicants, Journée SMA PRC-GRD, Intelligence artificielle, Nancy, (Décembre 1992).
- [Wang, A.I. et al., 1999]: Wang, A.I., Conradi, R., Liu, C.: A multiagent Architecture for CSE. Software Engineering and Knowledge Engineering (SEKE) Kaiserslautern, Germany. (1999).
- [Wiederhold, G., 1992]: Wiederhold, G. : Mediators in the Architecture of Future Information Systems, IEEE Computer, Vol.25, N°3, p. 38-49, (Mars 1992).
- [Wiederhold, G., Genesereth, M., 1995] : Wiederhold, G., Genesereth, M.: The Basis for Mediation, Proceedings of the third International Conference on Cooperative Information Systems, pp. 140-157, (Mai 1995).
- [Wooldridge, M.J., 2000]: Wooldridge, M.J.: The computational complexity of agent design problems, Dans : Fourth International Conference on Multi Agent Systems , icma'2000, Boston, Massachusetts (USA), (10-12 Juillet 2000).
- [Wooldridge, M., 2002]: Wooldridge, M.: An Introduction to Multi-Agent Systems. JohnWiley & Sons. (2002).
- [Zacklad, M., 1999]: Zacklad, M. : Une définition renouvelée du Système d'Information entre Système de Connaissances Métier et Système Informatique, (1999).
- [Zambonelli, F. et al., 2001]: Zambonelli, F., Jennings, N., Wooldridge, M. : Organisational Abstractions for the Analysis and Design of MultiAgent Systems. In: Ciancarini P.,

Wooldridge, M. (eds.): AgentOriented Software Engineering, LNCS 1957, Springer-Verlag, pp. 235 – 251. (2001).

[Zarour, K., 2007]: Zarour, K. : Vers un système d'information coopératif pour la prise en charge des soins à domicile : une architecture basée agent, Thèse présentée pour l'obtention de grade de magister en Informatique, Université Mentouri de Constantine, Constantine, Algérie. (2007).

[Zarour, K. et al., 2010] : Zarour, K., Zarour, N., and Khalfi, S.: Inter-Agent Interaction In Medical Information System: Case Homecare. Journal of Theoretical & Applied Information Technology. Vol. 11, N. 2, pp.130-142. (2010).

[Zarour, N., 2004]: Zarour, N.: Contribution à la modélisation de la coopération des systèmes d'information distribués et hétérogènes : le système DAArACHE- Application aux entreprises de production. Thèse présentée pour l'obtention de grade de docteur d'état en Informatique, Université Mentouri de Constantine, Constantine, Algérie, (Septembre 2004).

[Zarour, N. et al., 2005] : Zarour, N., Seinturier, L., Boufaïda, M. et Estraillier, P. : Supporting Virtual Enterprise Systems Using Agent Coordination, Int. Journal of Knowledge and Information Systems (IJKAIS), Springer Verlag Publishing. Vol. 8, N.3, pp. 330-349, (September 2005).

[Zhang, W.R. et al., 1992]: Zhang, W.R., Chen, S.S., Wang W., and King, R.: A Cognitive Map-Based Approach To The Coordination Of Distributed Cooperative Agents. IEEE Transactions on systems, man, and cybernetics.Vol 22, N.1, pp.103–113, (1992).