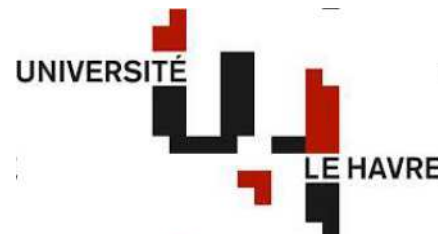




Université Abdelmalek Essaâdi
Faculté des Sciences et
Techniques, Tanger, Maroc.

Thèse en Cotutelle

Entre :



Université du Havre
UFR des Sciences et Techniques,
le Havre, France.

Spécialité
Informatique

Présentée par
M. Abdelhamid ZOUHAIR

Pour obtenir le grade de :
DOCTEUR de l'Université Abdelmalek Essaâdi
Et
DOCTEUR de l'Université du Havre

Sujet de la thèse :

Raisonnement à Partir de Cas Dynamique Multi-agents - Application à un Système de Tuteur Intelligent

Soutenue le 20 Octobre 2014 À la Faculté des Sciences et Techniques de Tanger
Devant le jury composé de :

M. FENNAN Abdelhadi	Université Abdelmalek Essaâdi, FST de Tanger	Président
M. ESSAIDI Mohamed	Université Mohammed V, ENSIAS, Rabat	Rapporteur
M. COLLET Pierre	Université de Strasbourg	Rapporteur
Mme. BOUKACHOUR Hadhoum	Université du Havre	Co-Encadrante
M. PERSON Patrick	Université du Havre	Co-Encadrant
M. EN-NAIMI El Mokhtar	Université Abdelmalek Essaâdi, FST de Tanger	Co-Encadrant
M. BERTELLE Cyrille	Université du Havre	Co-Directeur de Thèse
M. AMAMI Benaissa	Université Abdelmalek Essaâdi, FST de Tanger	Co-Directeur de Thèse

Remerciements

Avant d'exposer les résultats de mes recherches, il est nécessaire d'exprimer, avec plaisir, ma reconnaissance et mes remerciements à tous ceux qui m'ont aidé à les réaliser ou qui m'ont fait l'honneur de les juger.

Merci ...

... à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail par leur confiance, leur soutien ou leur temps, et en particulier :

J'adresse mes plus vifs remerciements à M. Abdelhadi FENNAN, Professeur à la FST de Tanger qui m'a fait l'honneur de présider le jury, je vous remercie cher professeur pour m'avoir fait l'honneur d'étudier mes travaux de thèse et pour les avoir cautionnés en qualité de rapporteur aussi.

Mes vifs remerciements s'adressent également au Professeur Mohamed ESSAAIDI Directeur de l'ENSIAS, Université Mohammed V, Rabat, au Professeur Pierre COLLET, Professeur à l'Université de Strasbourg d'avoir accepté de rapporter mon travail. Vos remarques et vos conseils, que j'ai tant appréciés, ont en effet mis en valeur mon travail de recherche et m'ont permis de me projeter vers l'avenir.

Je tiens à remercier également, mes Directeurs de Thèse : Monsieur Benaissa AMAMI, Professeur à la FST de Tanger, Monsieur Cyrille BERTELLE, Professeur à l'Université du Havre, pour leurs confiances, orientations, conseils, soutiens et leurs encouragements tout au long de la thèse. Ils n'ont pas cessés de m'aider à élaborer ce travail dans les meilleures conditions. Ils ont été toujours disponibles pour répondre aux questions que je leurs posais. Je les remercie d'avoir consacré beaucoup de leurs temps pour les nombreuses relectures de mon document. Leurs remarques m'ont permis de faire progresser ce travail.

J'adresse mes vifs remerciements aux membres du jury qui m'ont fait l'honneur de bien vouloir évaluer mon travail, et plus précisément : mes encadreurs :

J'adresse mes vifs remerciements à mon co-encadrant M. El Mokhtar EN-NAIMI, Professeur Habilité à la FST de Tanger, qui a su m'encadrer chaleureusement et de façon très constructive. Ses orientations, ses observations, ses critiques et ses conseils, m'ont été très bénéfiques. Il est toujours très disponible et ouvert à la discussion. Je le remercie d'avoir consacré beaucoup de son temps pour l'encadrement. Je le remercie particulièrement pour les nombreux week-ends de travail qu'on a partagé sensiblement.

Mes vifs remerciements s'adressent également à ma co-encadrante Mme. Hadhoum BOUKACHOUR Maître de conférences à l'Université du Havre, ainsi qu'à mon co-

encadrant M. Patrick PERSON, Maitre de conférences à l'Université du Havre. Je tiens à exprimer toute ma reconnaissance pour la pleine confiance que tous mes Cher(e)s Co-encadrant(e)s : M. EN-NAIMI, Mme. BOUKACHOUR et M. PERSON, m'ont été accordée et de m'avoir soutenu et guidé et ce d'une manière très constructive. Je leur exprime toute ma gratitude pour leurs précieux conseils, orientations, observations leurs suivis, leurs soutiens et leurs encouragements qui m'ont été très bénéfiques et surtout leur disponibilité pendant toute la période de la préparation de ma thèse. Je les remercie particulièrement pour leur confiance, leur générosité, leur sympathie et leur gentillesse.

Merci à mon adorable groupe de travail : Thierry, Jean et Florence de m'avoir accueilli et aidé tout au long de cette thèse.

Je tiens à remercier, M. Ousama MANSOUM Ingénieur en Informatique et Statistique Décisionnelles pour sa participation au développement et au test de la plateforme.

Je tiens à remercier également, M. Damien OLIVIER Co-Directeur du laboratoire LITIS, Yassine ADNAN Professeur à l'Université du Havre, Aziz ALAOUI Directeur du Laboratoire LMAH à l'université du Havre, Mohammed BOUHORMA Directeur du laboratoire LIST de la FST de Tanger, Taoufik MOURABIT Vice Doyen Chargé de la Recherche Scientifique et de la Coopération à la FST de Tanger, Frédéric GUINAND Professeur à l'université du Havre et Claude DUVALLET Maitre de conférences à l'université du Havre.

Mes vifs remerciements s'adressent à mes collègues pour leur encouragement permanent, je cite tout particulièrement : Ezzeddine BENAÏSSA, Hamdi DKHIL, Ihsen Farah, Mimoun ZEKHENINI, Mustapha ESSADAoui, Faïd TEMSMANI, Ahmed ELJAOUAHARI, Rachid AMERNIS, Khalid ECHELOCHI, Zhour GHARB et tous les autres.

J'ai le plaisir aussi de remercier chaleureusement, tous mes collègues des Laboratoires LITIS de l'Université du Havre et LIST de la FST de Tanger, pour l'aide, la gentillesse et pour leur soutien.

Dédicace



Afin d'être reconnaissant envers ceux qui m'ont appuyé et encouragé à effectuer ce
travail de recherche,

Je dédie ce travail :

À ma très chère mère, À mon très cher père qui ont été à mes cotés dans tous les
moments difficiles, par leur amour et leur encouragement, que Dieu vous protège pour

nous ;

A mes sœurs Safae, Atika et Ghezlane et frères Said, Redouan et SiMohamed ;

À tous les membres de ma famille et la famille de ma femme sans aucune exception ;

A mes meilleurs amis ;

A tous ceux qui me sont chers ;

Et

À ma chère femme pour son amour, sa patience et ses sacrifices.

Résumé

Nos travaux de thèse se situent dans le domaine des Systèmes d'Aide à la Décision (SAD) pour les situations dynamiques fondées sur les expériences passées. Plusieurs approches telles que les Réseaux de Neurones, Réseau de Petri et les Modèles de Markov Cachés ont été utilisées dans ce contexte mais elles souffrent de limites dans la gestion automatique et en temps réel des paramètres dynamiques (le concepteur doit définir au préalable les différentes situations possibles). Nous proposons une architecture multi-agents multicouche fondée sur le raisonnement à partir de cas dynamique et incrémentale (RàPCDI) capable d'étudier les situations dynamiques (reconnaissance, prédiction, et apprentissage de situations). Nous proposons une approche générique qui acquiert elle-même les connaissances du système dynamique étudié. En outre, les systèmes de Règle à Partir de Cas (RàPC) statiques souffrent de limites dans la gestion des paramètres dynamiques et ils sont incapables de détecter automatiquement l'évolution de ses paramètres ainsi que de s'adapter aux changements de la situation en cours d'évolution. En se basant sur le raisonnement à partir de cas et sur le paradigme multi-agents, nous proposons une modification du cycle statique de Règle à Partir de Cas (RàPC) dans le but d'introduire un processus dynamique de raisonnement à partir de cas fondé sur une mesure de similarité dynamique, capable d'évaluer en temps réel la similarité entre une situation dynamique en cours de progression (cas cible) et des expériences passées stockées dans la mémoire du système (des cas sources) afin de prédire la suite de la situation cible.

Nous validons l'approche proposée par la mise en œuvre d'un prototype de Tuteur Intelligent, capable d'initier et de gérer un apprentissage et d'assurer un suivi individualisé de l'apprenant. Les Systèmes Tutoriels Intelligents actuels possèdent deux limitations importantes : (1) la difficulté d'acquisition des connaissances du domaine, (2) ils sont liés aux domaines d'apprentissage (manque de généricité). Or, ces deux facteurs ont un rôle très important à jouer dans le processus permettant aux systèmes de formation et plus particulièrement aux systèmes tutoriels intelligents de mieux s'adapter aux apprenants. L'approche Incremental Dynamic Case Based Reasoning-Multi-Agent System (IDCBR-MAS) que nous proposons permet d'apporter des éléments de réponses aux difficultés liées au développement et aux limites des systèmes tutoriels intelligents. En effet, notre approche, nécessite moins de connaissances du domaine ce qui constitue un atout pour sa mise en pratique. Et en plus, nous proposons une approche générique indépendante de tout domaine d'apprentissage.

Dans notre approche IDCBR-MAS, les expériences passées sont modélisées sous la forme des traces. Elles comprennent les productions et les actions résultantes de l'interaction de l'apprenant avec la plateforme d'apprentissage. IDCBR-MAS a été analysé, conçu et modélisé suivant la méthodologie de développement de système multi-agents AUML. Ses agents ont été développés en se basant sur JAVA et la plateforme SMA/JADE.

Mots-clés : Systèmes d'Aide à la Décision, Système Dynamique, Raisonnement à Partir de Cas Dynamique, Mesure de Similarité Dynamique, Système Multi-Agents, Système Tutoriel Intelligent, Traces.

Abstract

Our thesis works are located in the field of Decision Support System (DSS) for dynamic situations based on past experiences. Several approaches have been used in this area such as neural networks, Petri Nets and Hidden Markov Model but they suffer from some limitations in automated real-time management dynamic parameters (the designer should be defined beforehand the different situations possible). We propose a multi-agent multi-layer architecture based on Incremental Dynamic Case-Based Reasoning (IDCBR) able to study dynamic situations (recognition, prediction, and learning situations). We propose a generic approach able to learn automatically from their experiences in order to acquire the knowledge automatically. In addition, the static CBR systems suffer from some limitations such as the problem of the static management for dynamic parameters of the situation and they are unable to detect automatically the changes in its parameters in order to adapt the situation during their evolution. Based on the Case-Based Reasoning and multi-agent paradigm, we propose a modification of the static CBR cycle in order to introduce a dynamic process of Case-Based Reasoning based on a dynamic similarity measure able to evaluate in real time the similarity between a dynamic situation in current Progress (target case) and past experiences stored in the memory (sources case) in order to predict the target case in the future (behavioral side).

We have been developing a generic intelligent tutoring system in order to validate the proposed approach, which can initiate learning and provide an individualized follow-up of learner. Current Intelligent Tutoring System has two important limitations: (1) the difficulty of acquiring the domain knowledge that the system will need, (2) they are related to learning areas (This is due to the lack of genericity of the ITS systems). However, these two factors have a very important role in the training process and more particularly for intelligent tutoring to better adapt and provide an individualized follow-up of learner. The Incremental Dynamic Case-Based Reasoning-Multi-Agent System (IDCBR-MAS) approach that we propose is able to provide some answers to limitations of intelligent tutoring system. Indeed, our approach requires less field knowledge. And in addition, we propose a generic approach : our approach can support any learning subject.

In our IDCBR-MAS approach, the past experiences are structured and modeled as traces. They include the interaction of the learner with the platform, which include history, chronology of interactions and productions left by the learner during his/her learning process. IDCBR-MAS was analyzed, designed and modeled with the agent unified modeling language AUML. its agents have been implemented with a JAVA, based Java Agent DEvelopment Framework JADE platform.

Key-words: Decision Support System, Dynamic Case-Based Reasoning, Dynamic Similarity Measure, Multi-Agent System, Intelligent Tutoring System, Traces.

Table des matières

Table des matières	vii
Liste des figures	xiii
Liste des tableaux	xvi
Liste des abréviations	xvii
Introduction Générale	1
1. Contexte	1
2. Problématique	1
3. Organisation du mémoire.....	3
Chapitre 1 : Systèmes d'Aide à la Décision	6
1.1 Introduction	7
1.2 Décision et système d'aide à la décision	8
1.2.1 Décision	8
1.2.2 Processus de décision	9
1.2.3 Aide à la décision	11
1.2.4 Système d'information décisionnel	12
1.2.4.1 Datawarehouse	13
1.2.4.2 Datamart.....	14
1.2.4.3 Dataweb	14
1.2.4.4 OLAP	14
1.2.4.5 Datamining.....	15
1.2.5 Les systèmes d'Aide à la Décision (SAD)	16
1.2.6 Architecture du système d'aide à la décision	17
1.3 Classification des systèmes d'aide à la décision	19
1.3.1 Classification au niveau utilisateur.....	19
1.3.2 Classification au niveau conceptuel	20
1.3.3 Classification au niveau système.....	20
1.3.4 Classification au niveau connaissance.....	21
1.4 Les systèmes à base de connaissances	22
1.4.1 Qu'est ce que la connaissance ?	22
1.4.2 Les systèmes experts	23
1.4.2.1 Méthodes de développements des systèmes experts.....	24
1.4.2.3 Limites et bilan	27
1.5 Les systèmes fondées sur des modèles.....	28
1.5.1 Réseaux de Pétri	28
1.5.2 Modèles de Markov cachés	29
1.5.3 Réseaux bayésiens	30
1.5.5 Limites et bilan	32

1.6 Les systèmes fondés sur les expériences passées : le raisonnement à partir de cas	33
1.7 Conclusion.....	34
Chapitre 2 : Raisonnement à Partir de cas	35
2.1 Motivation à l'utilisation du RàPC.....	37
2.2 Historique.....	37
2.3 Communautés en RàPC.....	38
2.4 Raisonnement à Partir de Cas.....	39
2.4.1 Principe RàPC.....	39
2.4.2 Carré d'analogie	39
2.4.3 Cycle du RàPC.....	40
2.4.3.1 Phase d'élaboration.....	41
2.4.3.2 Phase de remémoration	42
2.4.3.3 Phase d'adaptation	43
2.4.3.4 Phase de validation	44
2.4.3.5 Phase d'apprentissage	45
2.4.4 Cas / Base de cas	46
2.4.4.1 Définition du cas	46
2.4.4.2 Structure d'un cas	46
2.4.4.3 Indexation du cas	48
2.4.4.4 Base de cas dans les systèmes de RàPC.....	48
2.4.5 Connaissances.....	49
2.4.6 Domaines d'application et conditions d'utilisation du RàPC.....	50
2.4.6.1 Typologies d'applications	50
2.4.6.2 Conditions d'applications	52
2.5 Raisonnement à partir de cas pour situations statiques	54
2.5.1 La similarité entre les cas en situations statiques	54
2.5.2 L'adaptation en situations statiques.....	55
2.5.3 Système CHEF	55
2.6 Raisonnement à partir de cas pour situations dynamiques.....	56
2.6.1 Représentation dynamique des cas	57
2.6.1.1 Représentation d'un historique	57
2.6.1.2 Paramètres cible et source.....	64
2.6.1.3 Paramètres passé et futur	64
2.6.1.4 Rôle du temps	65
2.6.2 Base de cas pour les RàPCD.....	66
2.6.2.1 Organisation de la base de cas pour RàPCD.....	66
2.6.2.2 Maintenance de la base de cas	66
2.6.3 Le cycle de vie de RàPCD.....	68
2.6.4 Cycle du raisonnement à partir de cas pour les situations dynamiques.....	69
2.6.4.1 Cycle du raisonnement statique pour RàPCD.....	69

2.6.4.2 Mesures de similarité dans le cas dynamique et cycle statique	70
2.6.5 Etude comparatif des principaux systèmes RàPCD	74
2.6.5.1 REBECAS.....	74
2.6.5.2 AuRA	76
2.6.5.3 SINS	77
2.6.5.4 CASEP2	77
2.6.5.5 BROADWAY	79
2.6.5.6 SAPED	80
2.6.6 Synthèse comparative	81
2.7 Conclusion.....	84
Chapitre 3 : Raisonnement à Partir de Cas Dynamiques Multi-Agents	85
3.1 Introduction	87
3.2 Agents, systèmes multi-agents et méthodologies.....	87
3.2.1 L'intelligence artificielle distribuée.....	87
3.2.2 Agents	88
3.2.2.1 Définitions.....	88
3.2.2.2 Propriétés des agents.....	90
3.2.2.3 Typologie des agents.....	90
3.2.2.4 L'environnement des agents	93
3.2.3 Les systèmes multi-agents	94
3.2.3.1 Définitions.....	94
3.2.3.2 Raisonnement et communication dans les SMA	96
3.2.3.3 La communication dans les SMA	96
3.2.3.4 Modes de coopération	97
3.2.3.5 Organisation multi-agents	98
3.2.4 Méthodologies de développement des SMA	100
3.2.4.1 Agent-UML.....	100
3.2.4.2 AALAADIN et AGR	101
3.2.4.3 AMOMCASYS.....	102
3.2.4.4 MOISE+	102
3.2.4.5 Voyelles	102
3.2.4.6 GAIA.....	103
3.2.4.7 ROADMAP.....	103
3.2.4.8 MaSE.....	104
3.2.4.9 O-MaSE	104
3.2.4.10 PASSI.....	104
3.2.4.11 ADELFE	105
3.2.4.12 INGENIAS.....	105
3.2.5 Les plateformes de développement SMA.....	105
3.2.5.1 Zeus.....	105

3.2.5.2 MADKIT.....	106
3.2.5.3 AgentBuilder.....	107
3.2.5.4 JADE.....	107
3.2.5.5 JACK.....	108
3.2.5.6 Étude comparative des plateformes agents	108
3.3 Systèmes RàPC fondés sur SMA	109
3.3.1 Classification des systèmes RàPC fondés sur SMA	109
3.3.1.1 Raisonnement à partir de cas orienté agent.....	110
3.3.1.2 Raisonnement à partir de cas orienté multi-agents	110
3.3.2 Principaux systèmes RàPC fondés sur les SMA	111
3.3.2.1 Système FPL	111
3.3.2.2 Système CICLMAN.....	111
3.3.2.3 Système CCBR	113
3.3.2.4 Système MCBR	115
3.3.2.5 Système CBR-TEAM	116
3.3.2.6 Système ProCLAIM.....	117
3.3.2.7 Système RoboCats	118
3.3.2.8 Système AMAL	120
3.3.2.9 Système POMAESS.....	121
3.3.2.10 Système S-MAS.....	123
3.3.3 Etude comparative des RàPC multi-agents	125
3.4 Conclusion.....	128
Chapitre 4 : Système de Raisonnement à Partir de Cas Incrémental Dynamique Multi-Agents : IDCBR-MAS.....	129
4.1 Introduction	130
4.2 Architecture globale	131
4.2.1 Cycle dynamique de raisonnement à partir de cas	131
4.2.2 Principe et description du système IDCBR-MAS	134
4.2.3 Couche d'élaboration	137
4.2.3.1 L'agent de requête	137
4.2.3.2 L'agent Générateur	137
4.2.3.3 L'agent Traces-C1	138
4.2.3.4 Processus d'élaboration de la situation.....	139
4.2.4 Couche de remémoration.....	140
4.2.4.1 L'agent ILCSS	140
4.2.4.2 L'agent Analyseur.....	140
4.2.4.3 L'agent-Traces-C2	141
4.2.4.4 L'agent d'apprentissage.....	142
4.2.4.5 Processus de remémoration.....	143
4.2.5 Couche d'adaptation.....	144
4.2.5.1 L'agent évaluateur.....	144

4.2.5.2 L'agent-Traces-C3	144
4.2.5.3 L'agent d'adaptation.....	145
4.2.5.4 Processus d'adaptation	146
4.3 Cas et base de cas	147
4.3.1 Structure de cas.....	147
4.3.2 Base de cas.....	148
4.4 Mesure et tableaux de similarités	149
4.4.1 Mesure de similarité dynamique : ILCSS (Inverse Longest Common SubSequence)	149
4.4.2 Tableaux des similarités	152
4.5 Conclusion.....	152
Chapitre 5 : Implémentation de IDCBR-MAS : Système Tutoriel Intelligent fondé sur les traces de l'apprenant.....	153
5.1 Introduction	154
5.2 E-Learning et les Systèmes Tutoriels Intelligents.....	155
5.2.1 E-Learning	155
5.2.2 Les Systèmes Tutoriels Intelligents	156
5.3 Les plateformes de formation à distance.....	157
5.3.1 Définition.....	157
5.3.2 Moodle.....	159
5.4 Systèmes Tutoriels Intelligents : notre objectif.....	161
5.5 Système Tutoriel Intelligent et RàPCD	162
5.5.1 Création du cours (JADE) sous Moodle.....	163
5.5.2 Interface Moodle/ IDCBR-MAS	164
5.5.3 Trace de l'apprenant.....	166
5.5.3.1 Trait sémantique.....	166
5.5.3.2 Cas source	167
5.5.3.3 Cas cible.....	170
5.5.4 Base de cas.....	171
5.6 Développement des agents IDCBR-MAS.....	173
5.6.1 Aspects techniques	173
5.6.1.1 Motivation à l'utilisation de la plateforme JADE.....	173
5.6.1.2 Outils de contrôle et de supervision des agents dans IDCBR-MAS.....	173
5.6.1.3 Communication inter-agents dans IDCBR-MAS	174
5.6.1.4 Introspector agent JADE dans IDCBR-MAS	175
5.6.2 Les agents IDCBR-MAS appliqués au STI.....	176
5.6.2.1 Processus d'élaboration d'une situation en cours de progression	176
5.6.2.2 Processus de remémoration de la situation en cours de progression	177
5.6.2.3 Processus d'adaptation de la situation en cours de progression.....	178
5.6.2.4 Processus de prédiction d'une situation d'apprentissage en cours de progression	179

5.6.3 Cycle dynamique IDCBR-MAS appliqués au STI.....	182
5.7 Application Web	188
5.7.1 Scénario d'apprentissage	188
5.7.2 Génération de similarités entre le cas cible et cas sources	191
5.7.3 Courbe des similarités	192
5.7.4 Le cas source le plus similaire	193
5.8 Conclusion.....	194
Conclusion générale	195
Références Bibliographiques	197
Publications	215
Annexe A	218

Liste des figures

Figure 1.1 Modèle du processus de décision de Simon [Simon, 1977].....	10
Figure 1.2 Architecture du <i>Datawarehouse</i> [Goglin, 2001].....	13
Figure 1.3 Structure d'un SAD selon [Sprague et Carlson, 1982].....	18
Figure 1.4 Structure d'un SIAD basée sur la connaissance selon [Marakas, 2003].....	19
Figure 1.5 Architecture d'un système expert [Valdes, 2004].....	24
Figure 1.6 Exemple schématisé de réseau de neurones [Baptiste, 2011].....	25
Figure 1.7 Exemple de réseau de Pétri marqué [Baptiste, 2011].....	28
Figure 1.8 Réseau bayésien pour un modèle de Markov caché [Baptiste, 2011].....	29
Figure 1.9 Exemple de réseau bayésien [Baptiste, 2011].....	31
Figure 2.1 Communautés en raisonnement à partir de cas.....	38
Figure 2.2 Carré d'analogie	40
Figure 2.3 Le cycle de raisonnement à partir de cas.....	41
Figure 2.4 Exemple de structure d'un cas	47
Figure 2.5 Exemple d'un cas cible.....	47
Figure 2.6 Exemple d'un cas source	47
Figure 2.7 Niveaux hiérarchiques des applications des systèmes de RàPC selon [Althoff, 2001].....	52
Figure 2.8 Représentation d'un cas en paramètres cible et source.....	64
Figure 2.9 Représentation d'un cas en paramètres passé et futur	65
Figure 2.10 Cycle de vie d'un système RàPCD [Saxena, 2007].....	68
Figure 2.11 La représentation d'un cas dans le système REBECAS suivant une direction de propagation.....	75
Figure 2.12 Principe de recherche dans SINS.....	77
Figure 2.13 Architecture du système CASEP2.....	79
Figure 2.14 L'architecture du système d'assistance SAPED.....	81
Figure 3.1 Typologie des Agents selon [Nwana, 1996].....	93
Figure 3.2 Représentation d'un SMA [Ferber, 1995].....	95
Figure 3.3 Topologie à structure hiérarchique [Strugeon, 1995].....	98
Figure 3.4 Topologie à structure de marché [Strugeon, 1995].....	99
Figure 3.5 Topologie à structure de communauté [Strugeon, 1995].....	99
Figure 3.6 Topologie à structure de société [Strugeon, 1995].....	99
Figure 3.7 Les branchements définis dans AUML.....	101
Figure 3.8 Organisation modèle AGR.....	101

Figure 3.9 L'approche d'apprentissage individuel [Soh et Luo, 2003].....	112
Figure 3.10 L'approche d'apprentissage coopératif [Soh et Luo, 2003].....	113
Figure 3.11 Architecture du système CCBR [Ginty et Smyth, 2001].....	114
Figure 3.12 Approche RàPC utilisant une base de cas externe lorsque la base de cas local est insuffisante [Leake et Sooriamurthi, 2002].....	115
Figure 3.13 Approche du RàPC multi-base en s'appuyant sur plusieurs bases de cas [Leake et Sooriamurthi, 2002].....	116
Figure 3.14 Architecture du système ProCLAIM [Tolchinsky et al., 2012].....	118
Figure 3.15 Architecture du système RoboCats [Marling et al., 2003].....	119
Figure 3.16 Scénario de négociation dans POMAESS.....	122
Figure 3.17 L'architecture du système S-MAS [Cristian Pinzón et al., 2011].....	123
Figure 4.1 Cycle classique R^5 du RàPC inspiré de [Aamodt et Plaza, 1994].....	132
Figure 4.2 Cycle RàPC dynamique dans le système IDCBR-MAS [Zouhair et al., 2014].....	133
Figure 4.3 Architecture générale du système IDCBR-MAS.....	136
Figure 4.4 Comportement de l'agent de requête.....	137
Figure 4.5 Comportement de l'agent Générateur.....	138
Figure 4.6 Comportement de l'agent-Traces-C1.....	138
Figure 4.7 Diagramme de séquence d'élaboration de la situation.....	139
Figure 4.8 Comportement de l'agent ILCSS.....	140
Figure 4.9 Comportement de l'agent Analyseur.....	141
Figure 4.10 Comportement de l'Agent-Traces-C2.....	142
Figure 4.11 Comportement de l'agent d'apprentissage.....	142
Figure 4.12 Diagramme de séquence de remémoration de la situation.....	143
Figure 4.13 Comportement de l'agent Evalueur.....	144
Figure 4.14 Comportement de l'agent-Traces-C3.....	145
Figure 4.15 Comportement de l'agent d'adaptation.....	146
Figure 4.16 Diagramme de séquence d'adaptation de la situation.....	147
Figure 4.17 Structure globale de la base de cas en système IDCBR-MAS.....	148
Figure 5.1 : STI/E-learning.....	157
Figure 5.2 : Principaux acteurs d'une plateforme e-learning.....	158
Figure 5.3 : Page d'accueil d'un cours Moodle	160
Figure 5.4 : Page d'accueil du cours JADE importée sous Moodle, éditée par eXelearning.....	164
Figure 5.5 : Architecture Moodle-IDCBR-MAS [Zouhair et al., 2014].....	165
Figure 5.6 : Fichier datalib.php version IDCBR-MAS [Zouhair et al., 2014].....	165
Figure 5.7 : Fichier trace d'un apprenant en interaction avec Moodle et IDCBR-MAS.....	166
Figure 5.8 : Structure globale de la base de cas en système IDCBR-MAS.....	172

Figure 5.9 : Aperçu de l'interface RMA (Remote Monitoring Agent) dans notre plateforme IDCBR-MAS.....	174
Figure 5.10 : Communication inter-agent dans IDCBR-MAS [Zouhair et al., 2014].....	175
Figure 5.11 : Messages émis/reçus de l'Agent_ILCSS [Zouhair et al., 2014].....	175
Figure 5.12 : Aperçu de l'Agent_ILCSS de l'IDCBR-MAS.....	176
Figure 5.13 : Diagramme de séquence d'élaboration du cas cible	177
Figure 5.14 : Diagramme de séquence de la phase remémoration	178
Figure 5.15 : Diagramme de séquence de la phase d'adaptation.....	179
Figure 5.16 : Diagramme de séquence de prédiction d'une situation d'apprentissage.....	181
Figure 5.17 : Graphe des similarités entre cas cible et cas sources.....	186
Figure 5.18 : Graphe des scores des étapes du cycle RaPCD.....	187
Figure 5.19 : Cas d'utilisation d'un tuteur.....	188
Figure 5.20 : Cas d'utilisation d'un apprenant.....	189
Figure 5.21 : cas d'utilisation de l'administrateur.....	191
Figure 5.22 : Similarité entre la trace cible et les traces passées [Zouhair et al., 2014].....	192
Figure 5.23 : Courbe des similarités [Zouhair et al., 2014].....	193
Figure 5.24 : Choix du cas source le plus similaire [Zouhair et al., 2014].....	194

Liste des tableaux

Tableau 1.1 : Avantages et inconvénients des méthodes de développements des systèmes experts [Melfort, 2010].....	27
Tableau 2.1 : Avantages et inconvénients des différents types de représentation des cas dynamiques	63
Tableau 2.2 : Comparaison de différentes mesures de similarités entre séquences [Aich, 2007].....	71
Tableau 2.3 : Comparaison entre des systèmes de RàPC traitant des situations dynamiques.....	83
Tableau 3.1 : Agent cognitif vs Agent réactif [Ferber, 1995].....	91
Tableau 3.2 : Comparaison des plateformes SMA.....	109
Tableau 3.3 : RàPC fondé sur SMA avec cas cible statique/dynamique [zouhair et al., 2014]	110
Tableau 3.4 : exemple de cas gardien de but	120
Tableau 3.5 : Comparatif du type et du domaine	126
Tableau 3.6 : Comparatif de l'organisation des cas.....	126
Tableau 3.7 : Comparatif du Cycle RàPC pour les différents systèmes.....	127
Tableau 4.1 : Classement des systèmes RàPC.....	130
Tableau 4.2 : Similarités entre trace cible et traces passées.....	152
Tableau 5.1 : Similarités du cas cible avec les cas sources.....	185

Liste des abréviations

AA	Agent Analyseur	DSS	Decision Support System
AAD	Agent d'ADaptation	DTW	Dynamic Time Warping
ACC	Agent Communication Channel	EAD	Enseignement À Distance
ACL	Agent Communication Language	EIS	Executive Information System
ADELFE	Atelier de DEveloppement de Logiciel à Fonctionnalité Emergente	ESS	Executive Support System
AE	Agent Evalueur	EIAH	Environnements Informatiques pour l'Apprentissage Humain
AG	Agent Générateur	FIPA	Foundation for Intelligent Physical Agents
AGR	(Agent, Groupe, Role)	FPL	Federated Peer Learning
AILCSS	Agent ILCSS	GK	Guideline Knowledge
AMS	Agent Management System	IA	Intelligence Artificielle
AMOMCASYS	Adaptable MOdeling Method for Complex Administrative SYStems	IAD	l'Intelligence Artificielle Distribuée
ANTS	Autonomous Negotiating Teams	IAP	Intelligence Artificielle Parallèle
AR	Agent de Requête	ILCSS	Inverse Longest Common SubSequence
ASR	Argument Scheme Repository	JADE	Java Agent DEveloppement Framework
AS	l'Agent de Sauvegarde	JVM	Java Virtual Machine
ASM	Argument Source Manager	KPPV	K Plus Proches Voisins
ATC1	Agent-Traces-Couche 1	KQML	Knowledge Query and Manipulation Language
ATC2	Agent-Traces-Couche 2	MADKIT	Multi-Agent Developpement Kit
ATC3	Agent-Traces-Couche 3	MaSE	Multi-Agent System Engineering
AUML	Agent based Unified Modeling Language	IDCBR-MAS	Incremental Dynamic Case Based Reasoning-Multi-Agent System
API	Application Programming Interface	MCBR	MultiCase-Base Reasoning
BDI	(Belief, Desire, Intention)	MMC	Modèle de Markov Caché
CBR	Case-Based Reasoning	MOP	Memory organization packets
CBRe	Case-Based Reasoning Engine	MOPS	Memory Organization Packets
CCBR	Collaborative Case Based Reasoning	MOISE+	Model of Organization for multi-agent SystEms
CMS	Content Management System	MASK	Multi-Agent System Kernel
CNP	Contract Net Protocol	Moodle	Modular Object-Oriented Dynamic Learning Environment
DF	Director Facilitator	SMA	Systèmes Multi-Agents
DOS	Denial Of Service	SID	Systèmes d'Informations Distribués
DS	Diagramme de séquence	STI	Systèmes Tutoriels Intelligents
OLAP	On Line Analytical Processing	XML	Extensible Markup Language

O-MaSE	Organization-based Multi-Agent System Engineering	PASSI	Process for Agent Societies Specification and Implementation
POMAESS	Problem-Oriented Multi-Agent-Based E-Service System	PHP	Hypertext Preprocessor
ROADMAP	Role Oriented Analysis and Design for Multi-Agent Programming		
PSS	Planning Support System		
QCM	Questionnaire à Choix Multiples		
RàPC	Raisonnement à Partir de Cas		
RàPCD	Raisonnement à Partir de Cas Dynamiques		
RàPT	Raisonnement à Partir de Trace		
RB	Réseau Bayésien		
RBD	Réseaux Bayésiens Dynamiques		
RN	Réseaux de Neurones		
RDP	Résolution Distribuée de Problèmes		
REBEL	Roadmap Editor Built for Easy Development		
RMA	Remote Monitoring Agent		
RP	Réseau de Pétri		
SAD	Systèmes d'Aide à la Décision		
SGBD	Système de Gestion de Bases de Données		
SGBM	Système de Gestion de Bases de Modèles		

Introduction Générale

1. Contexte

Les systèmes d'aide à la décision (SAD) sont des outils informatiques dont l'objectif principal est d'assister les décideurs. Un SAD est spécifiquement développé pour aider à la résolution d'un problème de décision. C'est un système d'information interactif, flexible, et adaptable. Le SAD permet de raisonner sur une situation statique ou dynamique, de faire un diagnostic, de proposer une décision ou un plan d'action. Il peut utiliser des modèles standards ou spécifiques, pour supporter les différentes phases de la prise de décision et inclure un système à base de connaissances.

Plusieurs SAD d'anticipation des situations dynamiques ont été conçus. On distingue trois grandes approches complémentaires :

- **Système expert** : capable de générer le raisonnement d'un expert dans un domaine défini. Le système expert se base sur un certain nombre de faits et/ou de données, les analyses et renvoie des décisions.
- **Systèmes basés sur des modèles** : ce sont des outils de modélisation des systèmes d'aide à la décision dynamique : les réseaux de Pétri, les réseaux bayésiens, les réseaux de neurones et les modèles de Markov cachés.
- **Systèmes fondés sur le Raisonnement à Partir de Cas (RàPC)** : le RàPC consiste à réutiliser face à un nouveau problème, appelé cas cible, la solution d'un ancien problème similaire déjà traité et résolu [Kolodner, 1993]. L'ensemble des expériences passées (problèmes similaires), appelées les cas sources, est représenté sous forme de base de cas.

Notre travail s'insère dans la troisième approche fondée sur le partage et la réutilisation des expériences passées. L'objectif consiste à prédire la suite d'une situation en cours de progression. Il s'agit d'étudier un système dynamique, d'identifier la situation qui est en cours de progression et d'anticiper la suite. La problématique est différente de celle pour une situation statique (reconnaissance des formes par exemple). Dans le cas statique, le problème étudié est figé, établi et n'évolue pas. Or pour la prédiction de situations dynamiques, l'évolution du système est à l'origine de problèmes supplémentaires : il faut prendre en considération dans la conception de tel SAD la dimension temporelle, en plus la situation à prédire peut être incomplète, car elle est en cours de progression et pas encore achevée.

2. Problématique

Le RàPC est un système d'aide à la décision de modélisation des expériences passées et d'apprentissage incrémental de ces expériences. C'est un outil de gestion des connaissances du domaine étudié, capable de réutiliser ces connaissances dans des cas

analogues pour prendre des décisions et d'enrichir la mémoire en ajoutant de nouvelles connaissances.

Nous distinguons deux catégories d'applications RàPC : les applications statiques et dynamiques. Pour le cas statique, un problème doit être complètement décrit avant d'exécuter le cycle de raisonnement pour trouver une solution au cas cible. Or dans notre contexte, le cas cible est un tracé qui évolue avec le temps, le raisonnement à partir de cas doit donc prendre en compte cette évolution de manière incrémentale. Il s'agit donc du raisonnement à partir de cas pour situation dynamique avec certaines particularités importantes. Plusieurs systèmes RàPC ont été développés pour traiter les situations dynamiques, mais souffrent de limites dans la gestion des paramètres dynamiques, autrement dit des paramètres qui varient au cours du temps. Dans un contexte dynamique, le système doit être doté d'un mécanisme pour détecter automatiquement l'évolution de ses paramètres et s'adapter aux changements de la situation. Les systèmes RàPC existants exploitent un cycle semblable à celui adopté par les situations statiques. Ils considèrent chaque changement dans la situation en cours de traitement comme un nouveau cas cible, et par la suite ils exécutent le même cycle statique à nouveau. Or le changement concerne toujours le même cas cible, donc l'utilisation du cycle classique de raisonnement semble non adéquate, par conséquent il faut un cycle dynamique pour traiter un cas cible dynamique.

La problématique de cette thèse concerne l'application du raisonnement à partir de cas pour les situations dynamiques. Si le concept de RàPC semble applicable, il apparaît que le cycle de raisonnement à partir de cas classique n'est pas adapté aux situations dynamiques. En fait l'évolution dynamique de la situation remet en question le cycle classique du RàPC. Les verrous auxquels nous nous intéressons pour répondre à cette problématique sont le changement des ordres des étapes du cycle RàPC, processus internes à ces étapes et la prise en compte des changements dynamiques de la situation.

Le RàPC se caractérise par son pouvoir d'adaptation face aux changements de l'environnement. Toutefois, l'évolution du système RàPC nous amène à acquérir de nouvelles connaissances, ce qui complique la formalisation des cas, de la connaissance associée ainsi que son traitement. Pour ces raisons, nous posons une liste d'hypothèses :

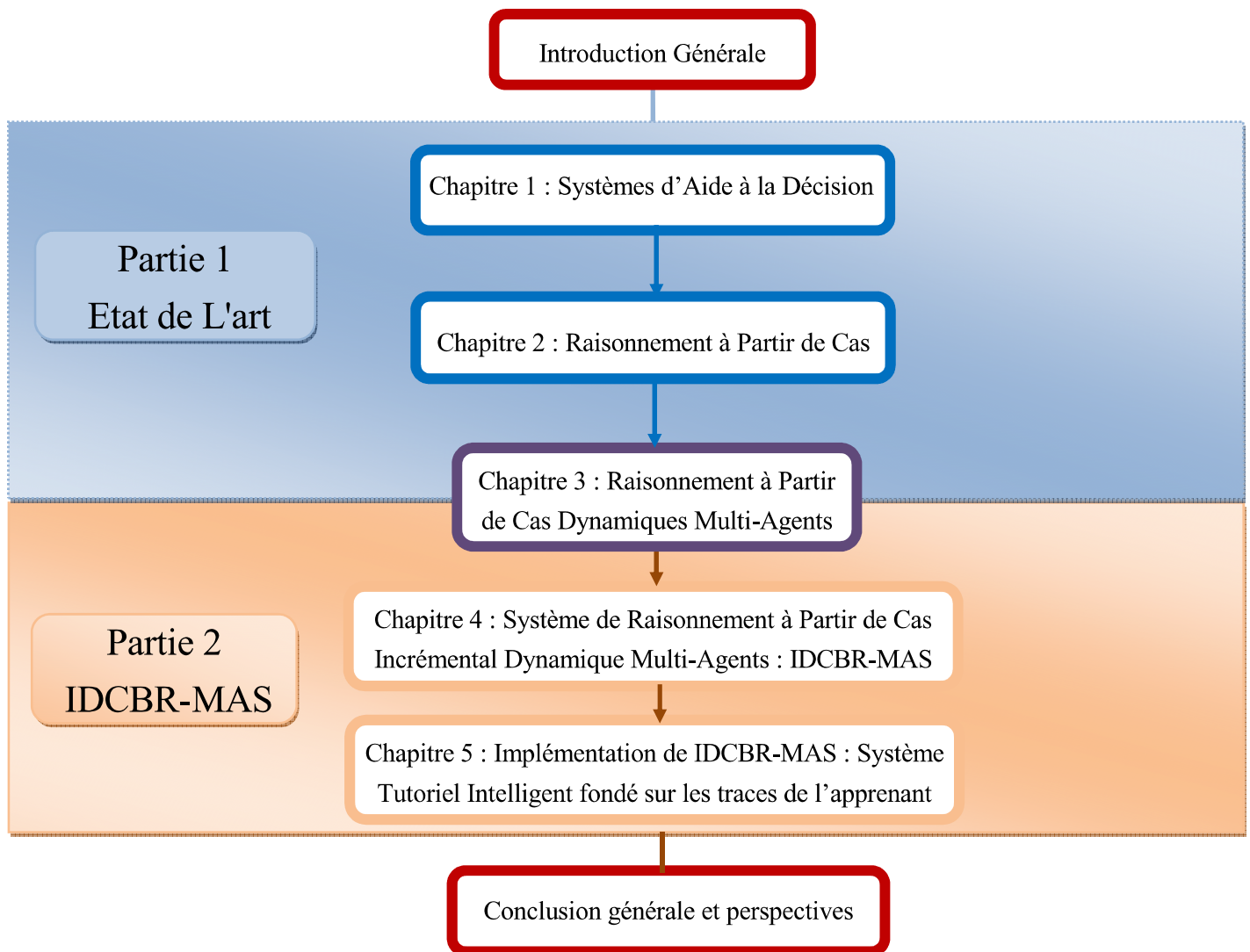
- Le système étudié est discret ou peut être rendu discret (par exemple échantillonné dans le temps de manière régulière, ou lors d'événements précis déclenchés par l'utilisateur : action, production,...) ;
- Les principaux paramètres du système sont observables ;
- Les expériences se reproduisent régulièrement et les cas retenus par le système (les expériences) peuvent être réutilisables dans le futur ;
- Les problèmes similaires peuvent avoir des solutions semblables ;
- L'application des mêmes actions (dans le cadre d'adaptation des solutions proposées) dans les mêmes conditions (contexte d'une situation) ont des conséquences semblables voir identiques ;
- Le début d'une situation dynamique conditionne la suite ;

- Les données utilisées sont supposées fiables et sans erreur ;

Les thèmes abordés dans cette thèse sont relatifs à la modélisation multi-agents du raisonnement à partir de cas pour les situations dynamiques. Dans cette thèse, nous proposons une approche multi-agents pour les systèmes RàPC dynamiques. Cette dernière est fondée sur un couplage des systèmes multi-agents et le raisonnement à partir de cas dynamique incrémental. Le paradigme multi-agents permet à notre système de : 1) surmonter des difficultés rencontrées avec l'utilisation des approches classiques, telle que la représentation temporelle complexe de la situation dynamique. 2) Prendre en compte l'évolution dynamique des valeurs de la situation en cours au fil du temps. Le raisonnement à partir de cas permet de rapprocher continûment la situation en cours de développement et les anciennes situations similaires en se basant sur un cycle RàPC dynamique : plusieurs va-et-vient entre les étapes du cycle RàPC peuvent se produire. Notre système est capable de ré-exécuter une étape antérieure du cycle RàPC suite à un changement dans le cas cible. Par exemple si le système est en train d'exécuter l'étape d'adaptation, et s'il détecte un changement dans le cas cible alors le système retourne à l'étape de remémoration en utilisant les similarités calculées précédemment et en se basant sur une mesure de similarité dynamique pour recalculer et réajuster les similarités entre la situation cible et les anciennes situations. Le SMA permet de doter notre système des capacités d'autonomie et d'adaptabilité : le passage d'une étape du cycle RàPC à une autre ou le retour à une étape précédente afin de réajuster les similarités se fait d'une manière automatique dès que le système détecte un changement dans la situation cible.

3. Organisation du mémoire

Ce mémoire de thèse est composé d'une introduction, cinq chapitres, et une conclusion générale. Les paragraphes suivants résument l'ensemble des chapitres :



L'**introduction** positionne notre problématique et évoque les objectifs qui ont motivés nos recherches. Elle traite en premier lieu le contexte de notre étude à savoir le raisonnement à partir de cas dynamique incrémental et la problématique de l'application du raisonnement à partir de cas pour les situations dynamiques.

La **première partie** concerne l'état de l'art. Nous commençons cette partie par la présentation des Systèmes d'Aide à la Décision. Par la suite, nous présentons les axes de recherche SAD : les systèmes experts, systèmes fondés sur les modèles (le réseau de Pétri, le réseau bayésien, les modèles de Markov cachés et les réseaux de neurones) et les systèmes fondés sur les expériences passées (raisonnement à partir de cas). Nous montrons que les deux premières classes ne pouvant répondre entièrement à notre problématique, et nous justifions le choix de l'approche raisonnement à partir de cas. Ensuite, nous détaillons les systèmes de raisonnement à partir de cas pour situations statiques et nous détaillons l'exemple du système CHEF. Nous continuons par la présentation des systèmes de raisonnement à partir de cas traitant de situations dynamiques, qui concernent principalement des applications de prédiction dans des

domaines variés. Nous présentons ensuite le cycle de Raisonnement à Partir de Cas pour situations dynamiques et plus particulièrement les mesures de similarités utilisées dans le cas dynamique. En outre, le troisième chapitre présente le concept d'agent et les systèmes multi-agents. La seconde partie de ce chapitre est consacrée aux systèmes RàPC fondés sur les SMA. Ensuite nous détaillons les principaux systèmes RàPC fondés sur les SMA et nous terminons cette partie de notre rapport par une étude comparative approfondie de ces systèmes.

La deuxième partie présente la solution proposée : elle décrit étape par étape notre plateforme intitulé IDCBR-MAS (*Incremental Dynamic Case Based Reasoning-Multi-Agent System*) qu'on propose pour les systèmes RàPC dynamiques. Cette dernière est fondée sur un couplage entre les systèmes multi-agents et le raisonnement à partir de cas dynamique. En premier temps, nous décrivons notre proposition de cycle RàPC dynamique en étoile au lieu de cycle RàPC linéaire classique et nous détaillons les processus internes de ses différentes étapes. Ensuite nous présentons une description de notre plateforme IDCBR-MAS à travers ses couches d'agents d'élaboration, de remémoration et d'adaptation. Nous présentons aussi l'architecture interne et le fonctionnement de la plateforme, ainsi que la structure de cas et de base de cas utilisées dans notre plateforme. La dernière partie contient l'implémentation et la validation de la solution proposée pour faciliter la compréhension de notre approche. Enfin nous présentons quelques résultats obtenus lors de la validation de notre approche.

Le rapport se termine par une **conclusion générale** qui résume les travaux réalisés et propose quelques pistes pour les travaux futurs.

Chapitre 1 : Systèmes d'Aide à la Décision

1.1 Introduction	7
1.2 Décision et système d'aide à la décision	8
1.2.1 Décision.....	8
1.2.2 Processus de décision	9
1.2.3 Aide à la décision	11
1.2.4 Système d'information décisionnel	12
1.2.4.1 Datawarehouse	13
1.2.4.2 Datamart	14
1.2.4.3 Dataweb.....	14
1.2.4.4 OLAP.....	14
1.2.4.5 Datamining	15
1.2.5 Les systèmes d'Aide à la Décision (SAD)	16
1.2.6 Architecture du système d'aide à la décision	17
1.3 Classification des systèmes d'aide à la décision	19
1.3.1 Classification au niveau utilisateur.....	19
1.3.2 Classification au niveau conceptuel	20
1.3.3 Classification au niveau système.....	20
1.3.4 Classification au niveau connaissance.....	21
1.4 Les systèmes à base de connaissances.....	22
1.4.1 Qu'est ce que la connaissance ?	22
1.4.2 Les systèmes experts	23
1.4.2.1 Méthodes de développements des systèmes experts	24
1.4.2.3 Limites et bilan	27
1.5 Les systèmes fondées sur des modèles.....	28
1.5.1 Réseaux de Pétri.....	28
1.5.2 Modèles de Markov cachés	29
1.5.3 Réseaux bayésiens.....	30
1.5.5 Limites et bilan.....	32
1.6 Les systèmes fondés sur les expériences passées : le raisonnement à partir de cas.	33
1.7 Conclusion.....	34

1.1 Introduction

La décision est un acte de choix entre plusieurs options pour résoudre un problème. Elle dépend principalement des informations disponibles et de l'environnement du problème. C'est un processus progressif, lié à d'autres processus secondaires conditionnés par l'existence de plusieurs choix alternatifs pour arriver à un but. Le processus de la prise de décision produit donc un résultat et un choix final. C'est le but fondamental de toute organisation et de toute gestion. Toute organisation dépend de la nature des décisions qui y sont prises par ses décideurs et acteurs [Adla, 2010].

De nombreuses situations de prise de décision auxquelles les décideurs doivent faire face se déroulent dans des environnements de plus en plus complexes et évoluant rapidement, ainsi, la prise de décision devient beaucoup plus difficile. L'un des problèmes importants est de déterminer les informations pertinentes pour la prise de décision [Holtzman, 1989]. Il devient donc fondamental d'utiliser des Systèmes d'Aide à la Décision, notés SAD (DSS en anglais : *Decision Support Systems*), capables d'évaluer la situation, les diverses alternatives et leurs impacts sur la situation.

Les systèmes d'aide à la décision sont donc des outils informatiques d'aide pour les décideurs, ils réunissent les ressources intellectuelles des individus avec les capacités de l'ordinateur pour améliorer la qualité des décisions. Les travaux sur les SAD sont nombreux et couvrent plusieurs domaines de recherche informatique tels que la recherche opérationnelle, l'intelligence artificielle, les systèmes de gestion de bases de données (SGBD).

Dans la première partie de ce chapitre, nous allons tenter de cerner le domaine des SAD en présentant leurs caractéristiques, les outils qu'ils utilisent, et les domaines d'applications. La seconde partie est dédiée aux classifications des SAD selon les niveaux utilisateur, conceptuel, système et connaissance. Ensuite, nous présentons trois axes de recherche SAD : systèmes à base de connaissances (les systèmes experts), systèmes fondés sur les modèles (le réseau de Pétri, le réseau bayésien, les modèles de Markov cachés et les réseaux de neurones) et enfin systèmes fondés sur les expériences (raisonnement à partir de cas). Nous montrons que les deux premières classes ne pouvant répondre entièrement à notre problématique, et nous justifions le choix de l'approche raisonnement à partir de cas. En effet notre contribution dans le domaine est la conception et la réalisation d'un système générique d'aide à la décision, fondé sur les expériences passées pour prévoir et anticiper une situation dynamique.

1.2 Décision et système d'aide à la décision

1.2.1 Décision

La décision est définie comme un acte de choix entre plusieurs alternatives [Schneider, 1994]. Pour d'autres, la décision est le résultat d'un processus global de résolution de problèmes (approches cognitives) [Simon, 1977]. Une décision est un processus de résolution de problème qui se pose à l'individu ou à l'organisation ou une action qui est prise pour faire face à une difficulté ou encore pour répondre à une modification de l'environnement [Lévine et Pomerol, 1989] [Adla, 2010]. Roy et Bouyssou [Roy, 1993] estiment que la décision est généralement présentée comme le fait d'un acteur isolé (décideur) qui exerce un choix entre plusieurs choix d'actions à un moment donné.

La théorie de la décision, nous permet de distinguer cinq catégories de problèmes décisionnels [Longueville, 2003] :

1. Description : problèmes liés à la caractérisation réelle de l'état courant de l'organisation ;
2. Investigation : problèmes liés aux relations entre deux ou plusieurs éléments de données ou événements ;
3. Explication : problèmes de structure de type une relation cause à effet ;
4. Prédiction : problèmes liés à la projection (prédiction) future basée sur des données historiques (des expériences) ;
5. Prescription : problèmes liés à la projection normative appuyée sur des données historiques.

Par ailleurs, d'après Simon [Simon, 1977], l'examen de la nature des décisions prises permet de répartir les décisions sur toute la longueur d'une échelle qui va du moins normalisée au plus normalisée.

- Une décision est bien normalisée si le processus de décision qui traite les informations entrantes dans le système est connu et explicitable.
- Une décision non normalisée est prise par une procédure peu programmée (ou non programmée), c'est une décision que l'on ne prend pas souvent et qui présente un caractère exceptionnel par sa nouveauté et son enjeu.

Une décision non normalisée est liée à un problème qui nécessite un grand effort pour en formaliser la résolution. Il faudra d'abord structurer et modéliser le problème afin de construire un espace de résolution [Simon, 1977]. Ainsi, selon le niveau de structuration des décisions, celles-ci peuvent être classées en trois types de décision [Trentesaux, 1996]:

- Décision structurée : le problème est clairement posé du point de vue technique : les données sont fiables et les modèles sont bien définis, ce qui permet au processus décisionnel d'aboutir à un résultat ou à une résolution fiable. Par exemple, l'automatisation du processus de réapprovisionnement du

stock dans une entreprise. Dans la majorité des entreprises, les décisions de réapprovisionnement des stocks sont quantifiées et automatisées. Les logiciels et outils de contrôle des stocks incluent des algorithmes décisionnels capables de gérer le processus d'approvisionnement dans un contexte de zéro rupture. L'informatisation des seuils économiques de réapprovisionnement et des stocks de sécurité constitue une décision structurée [James, 1995].

- Décision peu ou mal structurée : le problème nécessite un effort important pour être formalisé, les données sont généralement qualitatives, peu ou non fiables ou difficilement accessibles. Dans cette classe de décision, on peut préétablir certaines procédures mais pas suffisamment pour automatiser le processus décisionnel. Le décideur prend l'avantage sur la machine contrairement aux problèmes structurés. Résoudre le problème nécessite de faire appel au savoir faire du décideur. Ce dernier cherche une solution satisfaisante, mais peut être pas la meilleure (recherche heuristique) [Garlatti, 1996]. Cette idée de recherche heuristique est liée au principe de rationalité limitée de Simon [Simon, 1977]. Par exemple, allouer des ressources aux différents services d'une entreprise. Aider un décideur à établir une stratégie d'allocation de ressources dans une entreprise doit prendre en compte tous les paramètres techniques liés aux services, mais aussi le profil des employés ou les contraintes budgétaires.
- Décision non structurée : le problème traité n'est pas clairement posé et la décision prise est difficilement justifiable de manière rationnelle [Simon, 1977]. Par ailleurs, la théorie de Simon se fonde sur l'idée qu'un décideur ne possède pas la connaissance totale de la situation, d'où le terme de "rationalité limitée" [Simon, 1977] et que ces limitations dans la connaissance des faits et des hypothèses proviennent spécialement des contraintes de l'organisation qui sélectionne ou favorise tel ou tel scénario en fonction de ses intérêts. Par exemple, la décision d'une entreprise de lancer un nouveau produit qui n'a jamais été fabriqué auparavant. Nous ne considérons pas seulement l'acte final de production, mais l'ensemble complexe d'activités liées à la fabrication et commercialisation de ce nouveau produit. Plusieurs facteurs inconnus ou imprévisibles nécessitent une approche moins structurée menant à des jugements subjectifs de la part des gestionnaires de l'entreprise.

1.2.2 Processus de décision

La modélisation de la décision fait l'objet de nombreuses disciplines afin d'appréhender et de comprendre les comportements des systèmes organisationnels et humains. Il s'agit ici de s'intéresser à la manière dont les décideurs collectent et utilisent les informations, et ceci dans le but de les assister [Taggart, 1981].

La notion de décision chez beaucoup de chercheurs est juste un choix ou une activité ponctuelle à un instant t donné. Chez d'autres, la décision est une suite d'activités évoluant dans le temps et impliquant la notion de processus. La dite

décision s'inscrit d'une manière ou d'une autre dans le temps : la recherche d'information, l'identification et le choix de solutions. On retrouve ainsi la notion de processus [Adla, 2010]. Dans le domaine de la science cognitive on rencontre aussi la notion de processus pour la résolution de problèmes. Selon Scott-Morton [Scott-Morton, 1971] tout processus de décision contient les éléments suivants :

1. Un processus de recherche pour déterminer des objectifs ;
2. La formulation d'objectifs exacts ;
3. La sélection d'alternatives (stratégiques) pour atteindre ces buts ;
4. L'évaluation et test des résultats.

Dans nos travaux, nous partageons la deuxième vision de la décision dans laquelle la décision ne peut être étudiée que via la notion de processus de décision. Parmi les nombreux travaux sur les processus de décision, Simon [Simon, 1977] a proposé une vision cognitive. Elle se concrétise par un schéma de prise de décision assez générique pour pouvoir être reconnu comme un modèle canonique de décision. Dans ce dernier les phases d'information et de conception jouent un rôle très important dans le processus de décision (figure 1.1) :

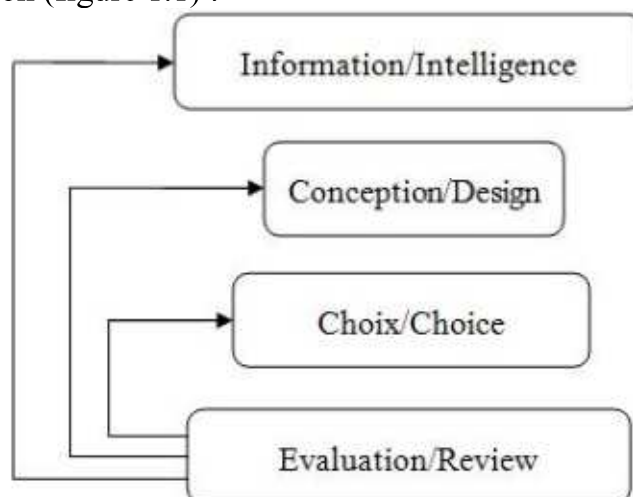


Figure 1.1 : Modèle du processus de décision de Simon [Simon, 1977]

La figure ci-dessus présente le modèle cognitif de Simon pour le processus de décision. Ce modèle distingue quatre étapes, non obligatoirement distinctes, dans le processus de décision :

1. La recherche d'information : cette étape vise à recenser l'ensemble des informations nécessaires pour la prise de décision. Le choix d'informations pertinentes est primordial pour le processus de décision car toutes les autres phases du processus s'appuient sur ces informations. Elles influencent fortement les autres étapes puisque les choix suivants en découlent.
2. La conception : cette phase comprend la conception, le développement et l'analyse des différentes suites éventuelles d'actions. Le décideur propose des solutions et les scénarios possibles, ce qui peut l'amener à rechercher des informations complémentaires.

3. Le choix : dans cette phase, le décideur choisit entre les différentes suites d'actions de la phase précédente. Il s'agit dans cette phase de définir les critères d'évaluation des différentes solutions envisageables (actions) et de mesurer les conséquences de chaque alternative.
4. L'évaluation : dans cette étape une solution sera recommandée pour le problème. Elle peut amener à la validation de la solution ou à la réactivation de l'une des trois phases précédentes. Le processus de décision peut s'intégrer dans un processus dynamique ou de nouvelles informations pertinentes peuvent influencer tel ou tel choix, voire le modifier complètement. Cette rétroaction permet de corriger d'éventuelles erreurs [Adla, 2010].

Dans la pratique, la liste des quatre phases ne s'enchaînent pas forcément de façon chronologique. Il s'agit donc d'un processus non linéaire, plusieurs allers-retours peuvent se produire. On peut par exemple être dans une situation où on est amené à générer une nouvelle alternative ou encore à rechercher de nouvelles informations ou encore l'acquisition d'informations supplémentaires. De nombreuses itérations peuvent être indispensables, ceci dépend de la complexité des problèmes de décision.

1.2.3 Aide à la décision

Selon Roy [Roy, 1993] l'aide à la décision est définie comme étant « l'activité de celui (homme d'étude) qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, cherche à obtenir des éléments de réponses aux questions que pose un intervenant (décideur) dans un processus de décision, éléments concourant à éclairer la décision et normalement à prescrire, à recommander ou simplement à favoriser un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service duquel cet intervenant se trouve placé, d'autre part ». C'est-à-dire on ne résout pas un problème, mais on aide le décideur à modéliser et à développer une représentation pertinente de la situation, afin de l'aider dans le processus de décision.

Nous signalons que les outils d'aide à la décision ne produisent pas une solution optimale pour tous les critères et tous les acteurs. Par ailleurs, ils contribuent à construire, à asseoir et à faire partager des convictions [Adla, 2010]. L'aide à la décision exploite des techniques et des méthodologies issues de plusieurs domaines comme les mathématiques appliquées (l'optimisation, les statistiques, la théorie de la décision), ou les sciences cognitives.

Roy [Roy, 1993] considère que l'aide à la décision est une science prenant appui sur trois postulats principaux :

1. Postulat de la réalité du premier ordre : les aspects de la réalité sur lesquels l'aide à la décision prend appui se ramènent à des objets de connaissances, ce sont les données (valeurs des caractéristiques des objets), valeurs jugées significatives et qui représentent un aspect de la réalité.

2. Postulat du décideur : toute décision est le fait direct d'un décideur, c'est un acteur bien déterminé, doté de pouvoirs, agissant sur le choix et le processus de la décision.
3. Postulat de l'optimum : dans toute situation dans laquelle une décision doit être prise, il existe au moins une décision *optimale*, cette décision reste neutre et indépendante vis-à-vis du processus de décision.

Dans le but de rendre aussi efficace que possible, il est nécessaire de développer des outils informatiques dédiés. Ces outils permettent de collecter, consolider, modéliser et restituer les données liées au problème traité. Ces outils peuvent être classés selon le rôle et les interventions qu'ils accordent à l'utilisateur (décideur) en cours de traitement et selon aussi l'importance des modèles de traitement par rapport aux représentations et aux données. Nous présentons dans la section suivante ces outils d'informatique décisionnelle.

1.2.4 Système d'information décisionnel

L'informatique décisionnelle ou système d'information décisionnel est un outil d'aide à la décision appuyé sur une base de données fédérant et homogénéisant les informations et données d'une organisation [Zararé, 2005]. Le système d'information décisionnel est un ensemble de données organisées d'une manière spécifique, ou encore une représentation intelligente de ces données au travers d'outils spécialisés, offrant au décideur toutes les informations nécessaires afin de prendre la bonne décision [Adla, 2010]. Ils sont basés sur des structures particulières de stockages volumineux (*Datawarehouse*, *bases OLAP*).

Ces outils d'aide à la décision permettent l'interrogation, l'incorporation de données et la restitution des résultats, à plusieurs niveaux de détails. Ils s'appuient sur les données actuelles mais aussi sur l'historique de ces données. Ceci dans le but d'analyser leur évolution temporelle et d'offrir au décideur une vision globale, pour prendre les bonnes décisions [Adla, 2010].

Deux fonctions principales sont destinées aux outils décisionnels, comme le montre la figure 1.2 ci-après :

1. Collecter et stocker : *Datawarehouse*, *Datamart*, *OLAP*, *Dataweb*.
2. Extraire et présenter : *Datamining*.

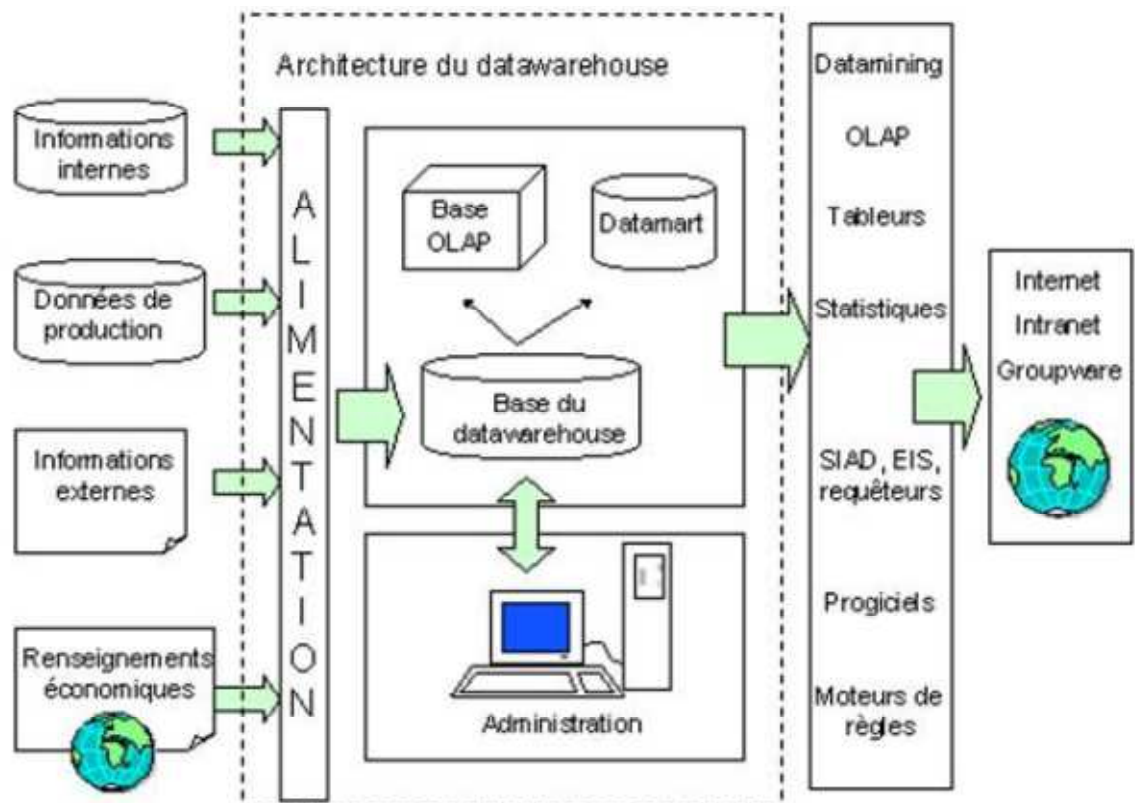


Figure 1.2: Architecture du *Datawarehouse* [Goglin, 2001]

1.2.4.1 Datawarehouse

Datawarehouse est un entrepôt de données, qui intègre des informations souvent réparties et hétérogènes (en provenance de différentes sources) et ceci dans le but de fournir une vue globale de l'information aux décideurs. Selon [Inmon, 1996], *Datawarehouse* est une collection de données thématiques, intégrées, non volatiles, historisées et organisées pour la prise de décision [Adla, 2010] :

- **Thématiques** : les données dans *Datawarehouse* sont organisées et gérées par thèmes au lieu de regrouper ces données par fonction (application). Donc on passera d'une vision verticale de l'organisation à une vision transversale beaucoup plus riche.
- **Intégrées** : l'intégration des données dans *Datawarehouse* nécessitera une forte normalisation, une bonne gestion des référentiels et de la cohérence, et une bonne maîtrise de la sémantique et des règles de gestion. L'intégration permet d'avoir une cohérence de l'information.
- **Non-volatiles** : dans le but de garder la traçabilité des informations et des décisions prises, les informations stockées au sein de *Datawarehouse* ne peuvent être supprimées. Le rafraîchissement de l'entrepôt, comprend l'ajout de nouvelles données, sans modifier ou perdre celles qui existent.

- Historisées : l'historisation des données stockées permet de suivre dans le temps leur évolution, en utilisant des techniques de prédiction, et en s'appuyant sur les évolutions passées afin de prévoir les évolutions futures.

La mise en œuvre d'un entrepôt de données se compose de plusieurs étapes :

1. L'analyse des sources de données et l'identification des besoins des utilisateurs ;
2. L'organisation des données à l'intérieur de l'entrepôt ;
3. L'interrogation des données : analyse des données et prédiction.

Ces étapes présentent plusieurs problèmes tels que, la difficulté de l'intégration des données, de manière à ce qu'elles soient de qualité pour leur stockage, la sélection des vues à matérialiser (l'organisation), le rafraîchissement de l'entrepôt, la gestion de l'ensemble de données (courantes et historisées), et enfin pour le processus d'interrogation des données, le problème du choix des outils performants pour l'accès et l'analyse de l'information [Adla, 2010].

1.2.4.2 Datamart

Un *Datamart* est un entrepôt de données particulier. C'est un sous-ensemble d'un *Datawarehouse* qui ne contient que les données d'un métier de l'entreprise, alors qu'un entrepôt de données contient toutes les données décisionnelles pour tous les métiers de l'entreprise. L'avantage de la mise en œuvre d'un tel outil *Datamart* est qu'il est plus facile dans son développement et sa mise en place par rapport à *Datawarehouse*.

1.2.4.3 Dataweb

C'est un concept qui permet d'enrichir le *Datawarehouse* avec le Web. Il permet l'accès à des bases de données distantes et publiques. L'un des avantages du *Dataweb* est de faciliter l'obtention des informations externes à l'organisme.

1.2.4.4 OLAP

OLAP (*On Line Analytical Processing* ou traitement analytique en ligne) est un outil décisionnel, orienté vers l'analyse en temps réel d'informations selon plusieurs axes, dans le but de produire des synthèses descriptives en ligne (ou vues) des données contenues dans les bases et les *Datawarehouses*. OLAP se base sur une structure de données adaptée aux extractions et aux croisements : hypercube (ou cube). Un hypercube est une architecture qui classifie les données en plusieurs axes appelés *dimensions*. Après avoir été extrait depuis un *Datawarehouse*, un hypercube permet d'extraire, ensuite de croiser des données de manière interactive suivant des critères précis. Ainsi, la technologie hypercube permet au décideur, de produire des tableaux multidimensionnels intermédiaires, afin de construire par sélection, les séries chronologiques ou les tableaux croisés [Adla, 2010].

1.2.4.5 Datamining

Le *Datamining* désigne la fouille de données, connue aussi sous l'expression d'exploration de données, ou encore l'extraction de connaissances à partir de données. Le *Datamining* utilise des techniques d'intelligence artificielle spécialement les réseaux de neurones, pour dégager des liens cachés entre les données. Le *Datamining* permet d'identifier des corrélations entre les données du système afin de faire ressortir des tendances [Adla, 2010].

L'extraction de connaissances par un expert (décideur) à pour objectif de remplir l'une des tâches ci-après :

- La classification permet d'examiner les caractéristiques d'un élément nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini.
- L'estimation est similaire à la classification à part que son résultat, permet d'obtenir une variable numérique plutôt que des catégories. Celle-ci est obtenue par une ou plusieurs fonctions combinant les données en entrée. En se basant sur les données existantes, l'estimation consiste à compléter des valeurs manquantes.
- La prédiction utilise en entrée un ensemble de données historiques pour anticiper les valeurs futures de ces variables. La plupart des applications de *Datamining* ont pour but la prédiction.
- Le regroupement par similitudes consiste à déterminer les éléments qui partagent des caractéristiques communes.
- L'analyse des clusters (*clustering*) ou segmentation permet de regrouper des objets en classes d'enregistrements ou d'objets similaires. Un cluster est une collection d'objets similaires et différents de ceux existants dans les autres clusters. Il permet de segmenter la totalité de données souvent hétérogène en des sous groupes relativement homogènes. Il permet donc de maximiser l'homogénéité à l'intérieur de chaque groupe et la minimiser entre les différents groupes.
- La description est une des tâches attendues d'un outil de datamining et consiste à décrire les données d'une base complexe.

Le concept d'informatique décisionnelle aborde l'aide à la décision d'un point de vue des données. L'aide à la décision, que peuvent fournir les SAD, est considérée non seulement d'un point de vue des données mais aussi en intégrant des modèles pour traiter ces données. Ainsi, l'objectif des systèmes d'aide à la décision est de fournir des éléments nécessaires pour recommander une décision, aider le décideur à comprendre et à agir en situation complexe. Dans la section suivante nous présentons le concept de Système d'Aide à la Décision.

1.2.5 Les systèmes d'Aide à la Décision (SAD)

Définitions

Scott-Morton a introduit pour la première fois, en 1971, l'aide à la décision qu'il a défini comme suit [Scott-Morton, 1971] : " [...] Ce travail, traitant de Décision et de Gestion, est basé sur une recherche qui combine à la fois des ordinateurs, des modèles analytiques et des terminaux utilisant des techniques de visualisation d'interactivité. Cette approche est basée sur l'analyse des décisions clés et fournit alors aux décideurs un support, une aide dans leur processus de prise de décision. Cette aide est possible dans des situations complexes et mal structurées et peut être utilisée par des décideurs en conjonction avec leur propre sentiment intuitif du problème et de sa solution."

Les Systèmes d'Aide à la Décision (SAD), désignent des systèmes informatiques qui sont chargés de fournir au décideur le maximum d'éléments dans les meilleurs délais et qui l'aideront dans sa prise de décision [Lévine et Pomerol, 1989]. Les SAD sont des outils et méthodes qui permettent de collecter, modéliser et traiter les données, en vue d'offrir une aide à la décision et de permettre aux intervenants d'avoir une vue d'ensemble de l'activité traitée [Adla, 2010].

Les SAD permettent d'évaluer la situation, les diverses alternatives possibles et leur impact. Ils offrent aussi au décideur un accès enrichi à l'information et une gestion efficace de sa complexité, afin de l'aider à prendre la meilleure décision. Un SAD est un Système de Traitement de l'Information (STI) capable d'extraire et d'offrir au décideur l'information nécessaire au processus de prise de décision. Toutefois, pour caractériser un SAD et le différencier des autres systèmes d'information, il est important de noter qu'un SAD est un résolveur de problèmes, destiné à résoudre des problèmes tout en laissant l'initiative à son utilisateur.

Keen et Scott-Morton [Keen et Scott-Morton, 1978] ont donné la définition suivante qui est l'une des plus citées : " Les systèmes d'aide à la décision font coupler les ressources intellectuelles des individus avec les capacités de l'ordinateur pour améliorer la qualité des décisions. C'est un système d'aide informatique aux décideurs qui traitent des problèmes semi-structurés ". Dans ce cadre, les SAD supposent l'utilisation d'outils informatiques pour :

1. Aider les décideurs dans leur processus de décision ;
2. Assister plutôt que remplacer le jugement des décideurs ;
3. Améliorer la qualité de la prise de décision.

Sprague et Carlson [Sprague et Carlson, 1982] ont proposé la définition suivante proche de celle proposée par Keen et Scott-Morton : " Les SAD peuvent être caractérisés comme des systèmes informatisés, interactifs, qui aident les décideurs en utilisant des données et des modèles pour résoudre des problèmes mal structurés ".

D'autres chercheurs ont remarqué que ces définitions concourent à dire que le système d'aide à la décision doit aider un décideur dans la résolution de problèmes non ou semi-structurés [Adla, 2010]. Des problèmes où :

- Les préférences, intuitions et les expériences passées sont indispensables ;
- La recherche d'une solution, la formalisation et la structuration du problème n'est pas connue à l'avance ;
- Les critères pour la prise de décision sont nombreux, et fortement dépendants de la perception de l'utilisateur ;
- La solution doit être obtenue en un temps limité.

Enfin, Turban [Turban, 1995] définit un système d'aide à la décision comme étant un système d'information interactif, adaptable et spécifiquement développé pour accompagner le décideur dans la résolution d'un problème de décision non structuré et afin d'améliorer la prise de décision. Il permet au décideur d'utiliser des données, de fournir une interface utilisateur et de développer ses propres idées ou points de vue. Il peut s'appuyer sur des modèles standards ou spécifiques, supporter les différentes phases de la prise de décision et utilise une base de connaissances.

Un système SAD est l'association d'un système informatique et d'un décideur. Le décideur, possède un méta-modèle du processus de décision, et le SAD, par sa capacité de traitement de l'information, l'accompagne dans la structuration des modèles [Lévine et Pomerol, 1989]. Le décideur contrôle le processus de décision et le SAD l'aide en effectuant les calculs et traitement sur les données [Kersten, 1999].

Le processus de décision peut être vu comme une recherche heuristique menée par le décideur ; le système présente le processus de recherche à l'aide d'indicateurs et d'informations. Le décideur continue l'exploration heuristique des actions possibles s'il n'est pas satisfait de la solution construite ou il n'a pas atteint ses buts à l'issue du traitement des données par le système [Pomerol, 1997].

Un système SAD doit être capable de gérer toutes les informations disponibles utiles à la décision suivant les perspectives d'analyse possibles, et il doit aussi proposer la trace de la logique décisionnelle qu'il a utilisé à des fins de recommandation. L'explication et recommandation permettent d'arbitrer la pertinence des déductions ou des inférences du SAD. Les SAD sont donc des outils qui permettent d'orienter le décideur vers des points qu'il ne pourrait pas toujours observer seul.

1.2.6 Architecture du système d'aide à la décision

Plusieurs architectures pour la conception de SAD ont été proposées, il n'existe pas d'architecture standard pour un SAD. D'après Sprague et Carlson un SAD comporte trois composants : l'interface homme-machine, un Système de Gestion de Bases de Données (SGBD) et un Système de Gestion de Bases de Modèles (SGBM) [Sprague et Carlson, 1982] [Adla, 2010] (Figure 1.3). Marakas [Marakas, 2003] propose une architecture de cinq parties distinctes (Figure 1.4) : le Système de Gestion de Base de

Données (SGBD), le Système de Gestion de Base de Modèles (SGBM), le Système de Gestion de Base de Connaissances (SGBC), l'interface utilisateur et les utilisateurs :

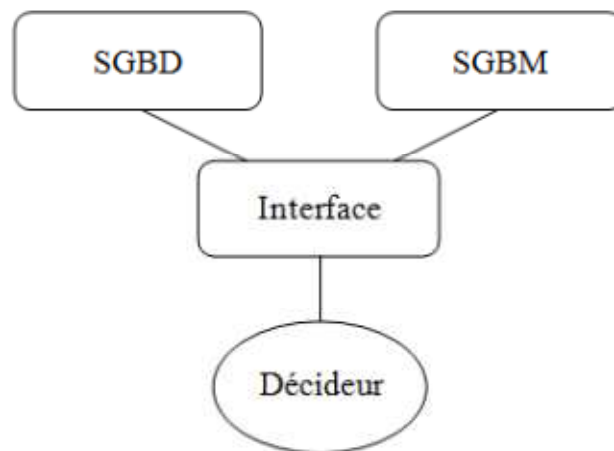


Figure 1.3 : Structure d'un SAD selon [Sprague et Carlson, 1982]

- **Système de Gestion de Bases de Données** : il stocke et organise les données pertinentes pour le système et il fournit des structures de données logiques avec lesquels l'utilisateur interagit. Les données sont les statistiques, l'estimation de l'évolution de certains paramètres environnementaux, ou autres données qui décrivent la situation courante et passée. Dans un SAD, un système de base de données doit permettre une recherche rapide de données via des requêtes et des rapports, et il doit aussi accomplir des tâches de recherche et de gestion complexe de données.
- **Système de Gestion de Base de Modèles** : il possède un rôle similaire au système gestionnaire de base de données excepté qu'il organise, trie, stocke les modèles de l'organisation. Il est chargé d'assurer l'indépendance entre les modèles qui sont utilisés dans un SAD et les applications qui les utilisent. Parmi les buts du SGBM citons la traduction des données d'un SGBD en des décisions utiles. Le module de données contient les procédures de calculs utilisées dans les différents traitements des données.
- **Interface Utilisateur** : c'est un élément clé des fonctionnalités du système global. Il permet aux utilisateurs d'interagir avec le SAD. Par l'intermédiaire des interfaces gérées par ce module, le décideur accède aux données et aux fonctions de calcul, et contrôle le SAD en utilisant ce sous-système. Le système utilise le même moyen pour lui communiquer le résultat des manipulations qu'il a effectué. Le sous-système offre la possibilité d'avoir une interaction dynamique et un échange d'informations en temps réel entre l'utilisateur et le système.
- **Système de Gestion de Base de Connaissances** : il remplit les tâches relatives à la reconnaissance de problèmes et à la génération de solutions aussi bien que

des fonctions relatives à la gestion du processus de résolution de problèmes (Figure 1.4).

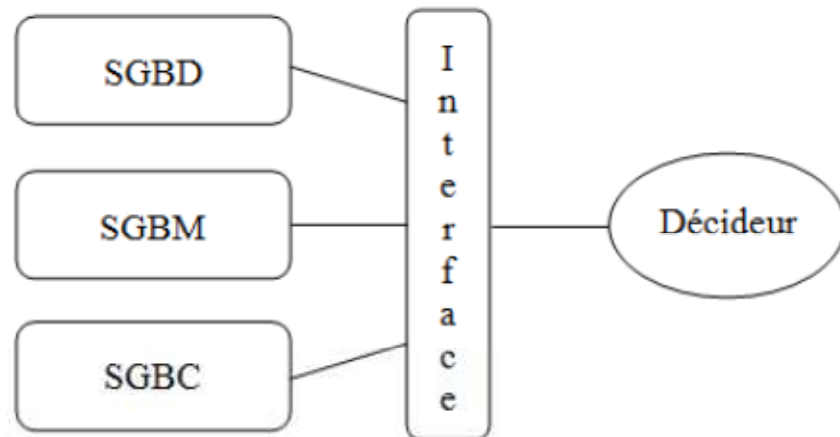


Figure 1.4 : Structure d'un SAD basé sur la connaissance selon [Marakas, 2003]

Les connaissances sont de deux catégories :

1. Les connaissances du domaine servent à définir une représentation du système qui sera utilisée pour aider le décideur dans le processus de décision. Elles couvrent la modélisation du système, les processus impliquant les experts du domaine et les cognitiens.
2. Les connaissances de résolution, de procédures et de théories, capables de traiter les connaissances du domaine afin de trouver une solution ou d'apporter des éléments d'aide à la décision.

En ce qui nous concerne, nous avons comme objectif de construire un SAD flexible et auto-adaptatif capable de traiter un problème complexe, à savoir représenter, suivre et analyser l'évolution d'une situation. L'utilisation d'outils d'intelligence artificielle [Lévine et Pomerol, 1989] a permis de renforcer le traitement de l'information en écartant les informations non pertinentes et en présentant des faits déduits à partir des données présentes dans les systèmes d'information. Grâce aux approches et aux technologies fournies par l'intelligence artificielle, il est possible actuellement de réaliser de tels systèmes dits *intelligents*, qui peuvent avoir ces caractéristiques et combler ainsi les lacunes des systèmes classiques.

1.3 Classification des systèmes d'aide à la décision

Dans la section suivante nous présentons une classification des systèmes d'aide à la décision selon les différents niveaux :

1.3.1 Classification au niveau utilisateur

Selon Hattenschwiler [Hattenschwiler, 1998] on distingue trois types de SAD :

1. SAD passif : c'est un système d'aide à la décision, qui favorise le processus décisionnel mais qui ne peut apporter au décideur des solutions ou des suggestions de décisions explicites ;
2. SAD actif : dans cette catégorie, le système peut apporter des solutions ou suggestions de décisions explicites ;
3. SAD coopératif : il permet au décideur de modifier, de compléter ou d'affiner les suggestions de décisions proposées par le système avant qu'il ne soit validé par le SAD. Ceci permet au système d'améliorer, de compléter et d'affiner les suggestions du décideur avant leur validation. Le processus entier se répète jusqu'à ce qu'une solution consolidée soit produite [Adla, 2010].

1.3.2 Classification au niveau conceptuel

Power [Power, 2002], a identifié quatre catégories génériques de systèmes d'aide à la décision :

1. SAD centré données : il permet au décideur de traiter et gérer une série temporelle de données internes à l'organisation et quelque fois de données externes ;
2. SAD orienté modèle : il donne au décideur l'accès et la manipulation d'un modèle de simulation ou d'optimisation. Un SAD orienté modèle gère des données et des paramètres pour assister les décideurs à analyser une situation, mais n'est pas forcément centré sur les données [Adla, 2010] ;
3. SAD orienté documents : ce type de système offre au décideur une expertise de résolution de problèmes, en se basant sur des outils tels que des faits, des règles, des procédures ou des structures similaires ;
4. SAD orienté communication : dans cette catégorie de système, plusieurs utilisateurs/décideurs travaillent sur une même tâche partagée.

1.3.3 Classification au niveau système

Lévine et Pomerol [Lévine et Pomerol, 1989] distinguent quatre catégories de systèmes décisionnels, dans une classification plus large des systèmes décisionnels (y compris les systèmes d'aide à la décision), suivant la tâche à réaliser [Adla, 2010] :

1. DSS (*Decision Support System*) ou SAD : ce sont des systèmes interactifs et flexibles, spécifiquement développés afin d'assister les décideurs à gérer et exploiter les données et les modèles. Les SAD sont des outils qui permettent de rechercher des solutions aux problèmes, souvent non structurés, et analysent les résultats d'éventuels changements de l'environnement du système ;
2. EIS (*Executive Information System*) : ce sont des systèmes de présentation des données sous forme de tableaux de bord, permettant aux différents niveaux de gestionnaires d'accéder aux informations qui leur sont destinées et de les présenter de façon élaborée. Ces outils sont dotés d'une interface graphique conviviale ;

3. ESS (*Executive Support System*) : ce sont des systèmes très similaires aux EIS. L'ESS a l'avantage par rapport aux EIS d'intégrer des outils de communication, d'analyse et d'intelligence. Autrement dit, l'EIS permet au décideur de se concentrer sur des données critiques afin d'obtenir une évaluation de l'organisation ;
4. PSS (*Planning Support System*) : ce sont des systèmes capables de fournir aux décideurs une assistance intelligente, en se basant sur une analyse de faisabilité des procédures ou des décisions retenues.

Retournons sur le modèle de Simon (Figure 1.1), la phase d'Information/Intelligence correspond à l'EIS et l'ESS : en effet l'EIS permet au décideur de se concentrer sur des données critiques telles que la chute des actions en bourse durant une période donnée, alors que l'ESS permet de réaliser une analyse pour une appréciation future de l'organisation. Les autres phases du modèle sont assurées selon Turban [Turban, 1998], [Adla, 2010] par le DSS. Il peut réaliser de la planification stratégique et offre par la suite la possibilité d'analyser les effets d'éventuels changements sur l'organisation, en se basant par exemple sur des simulations.

1.3.4 Classification au niveau connaissance

Un autre classement des systèmes d'aide à la décision consiste à distinguer quatre classes de systèmes, selon la nature des connaissances utilisées pour les construire [Lévine et Pomerol, 1989] [Adla, 2010] :

1. Les systèmes à base de savoir théorique lié au problème, permettant de trouver les décisions optimales ou encore satisfaisantes. Ils sont capables de simuler les résultats des décisions examinées ;
2. Les systèmes à base de savoir expert : ce sont des systèmes qui sont créés à partir des connaissances qu'utilisent les experts du domaine pour résoudre les différents problèmes ;
3. Les systèmes à base de savoir factuel : ce sont des SAD généralement construits par apprentissage par le système à partir d'un grand nombre de problèmes du domaine déjà résolus, en particulier les systèmes fondés sur les réseaux de neurones ;
4. Les systèmes et outils informatiques classiques : dans cette catégorie les SAD combinent les trois éléments suivants: (1) une base de données fournissant les informations nécessaires à la prise de décision, (2) un tableur calculant quelques éléments de décision, et enfin (3) une interface permettant de les présenter sous un format convivial.

Nous avons présenté la classification des systèmes d'aide à la décision selon les différents niveaux : utilisateur, conceptuel, système et connaissance. Nous avons montré que cette classification dépend du type des connaissances utilisées, dépend aussi de la conception des modèles qui gèrent ces connaissances, et des résultats

présentés par les outils SAD pour assister les décideurs à analyser une situation. Dans ce contexte, nous distinguons trois principaux axes de recherche pour les SAD :

1. SAD orienté connaissances ou les systèmes experts: l'objectif de cet axe est de reproduire les mécanismes cognitifs des experts, dans un domaine particulier.
2. SAD orienté statistiques et recherche opérationnelle ou encore SAD fondés sur les modèles tels que le réseau de Pétri, le réseau bayésien, les modèles de Markov cachés et les réseaux de neurones. Ce sont des outils graphiques et/ou mathématiques permettant de modéliser le comportement dynamique des systèmes.
3. SAD orienté expériences ou le raisonnement à partir de cas. C'est l'interprétation d'une nouvelle situation dans le contexte d'une ancienne situation, ou encore la reconnaissance et l'anticipation de situations dynamiques qui utilisent des expériences passées.

Dans les sections suivantes, nous allons présenter les systèmes à base de connaissances, les systèmes fondés sur les modèles mathématiques et le raisonnement à partir de cas.

1.4 Les systèmes à base de connaissances

1.4.1 Qu'est ce que la connaissance ?

Il n'y a pas, aujourd'hui, de définition unifiée du concept de connaissance mais plutôt des définitions inhérentes aux disciplines telles que la philosophie, l'épistémologie, les sciences cognitives, la sociologie, l'anthropologie. Selon Prax [Prax, 2007], la connaissance peut être définie comme la sélection, l'appropriation et l'interprétation des informations par un expert. Dans une définition plus large de la connaissance, Perrin [Perrin, 2004], distingue trois grandes notions :

- une connaissance est une notion plus ou moins complète ou précise que l'on a de quelque chose ;
- une connaissance désigne la faculté de connaître, de comprendre ;
- une connaissance est le droit de porter un jugement sur quelque chose qui est lié au niveau de connaissance que l'on a de cette chose.

Selon Davenport [Davenport, 2000], "la connaissance est un mélange inconstant d'une expérience cadrée, des valeurs, des informations contextuelles et d'un aperçu d'un expert qui fournit un cadre pour l'évaluation et l'incorporation de nouvelles expériences et d'informations. [...]. Dans les organisations, elle devient souvent intégrée non seulement dans les documents ou les dépôts, mais aussi dans les routines, les processus, les pratiques et les normes organisationnelles".

La connaissance est étroitement liée aux deux concepts : les données et l'information. Selon Vance [Vance, 1997], les données peuvent être définies en tant

que des nombres et des faits bruts, en général mesurables. Les informations sont des données traitées et contextualisées (par exemple "il fait froid aujourd'hui"). Enfin une connaissance correspond à l'appropriation et l'interprétation des informations. Exemple, Pour avoir froid, il suffit d'allumer la clim.

Dans la partie suivante, nous allons présenter les systèmes experts.

1.4.2 Les systèmes experts

Un système expert ou système à base de règles est un système d'aide à la décision. C'est un système capable de reproduire le raisonnement d'un expert dans un domaine défini. Il est la *version logicielle* de la réflexion d'un expert dans un domaine donné. Néanmoins, il ne peut pas remplacer la capacité d'analyse d'un expert, c'est un outil d'aide à la décision. Le système expert se base sur un certain nombre de faits et/ou de données, les analyse et renvoie un verdict (décision) [Duvallat, 2001]. Pour un système expert, le raisonnement suit la règle de production du type :

si la précondition est satisfaite alors exécuter une action

Parmi les cas particuliers des règles, nous trouvons les règles déductives (ou inférences) qui sont du type :

Si (prémisses) vraies Alors (conclusions) vraies.

Avec :

- Prémisses = conditions de déclenchement de la règle ;
- Conclusions = effets de l'application de la règle.

Un système expert est composé des éléments suivants Melfort [Melfort, 2010] :

- **la base de connaissances** qui contient les trois éléments indispensables :
 - des **standards d'engagement** ou connaissances de l'expert, qui représentent les informations de base et les informations de configurations du système ;
 - des **règles d'inférence** qui représentent le savoir faire avec l'ensemble des règles logiques de déduction utilisées par le moteur d'inférence ;
 - d'une **base de faits** contenant les faits observés et les faits déduits.
- **le moteur d'inférence** permet au système expert d'inférer des connaissances nouvelles à partir de la base de connaissances du système. Il conduit les raisonnements logiques et il permet de dériver des résultats à partir d'une base de faits et d'une base de connaissances. Le moteur est chargé de répondre aux requêtes utilisateur ou serveur. Nous distinguons deux modes de fonctionnement :
 - chaînage avant : déduction d'une solution à partir de faits ;
 - chaînage arrière : recherche des règles et faits ayant conduit à une solution.
- **l'interface graphique** permet à l'utilisateur d'interagir avec le système [Valdes, 2004].

La figure suivante montre l'architecture générale d'un système expert :

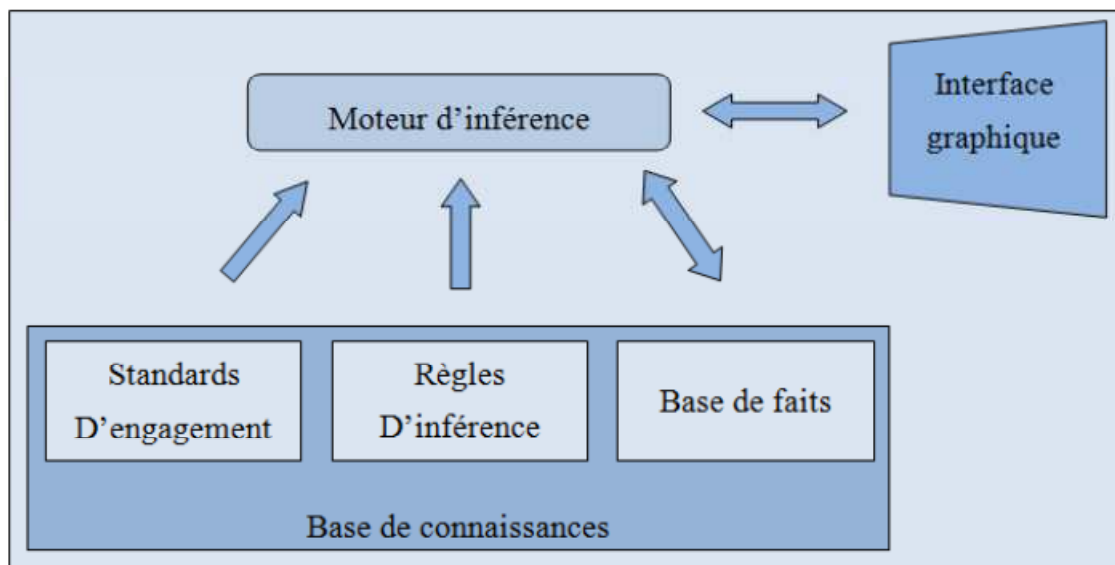


Figure 1.5 : Architecture d'un système expert [Valdes, 2004]

1.4.2.1 Méthodes de développements des systèmes experts

Différentes méthodes et techniques de calcul peuvent être utilisées dans un système expert. Elles constituent la base du moteur d'inférence et des règles de décision associées. Dans cette partie, nous présentons les principales méthodes utilisées : les algorithmes génétiques, les réseaux de neurones, la logique floue et la méthode d'agrégation multicritères (DEXi).

a) Algorithme génétique

Les algorithmes génétiques, initiés par John Holland au début des années 70, sont des algorithmes évolutionnaires d'optimisation. Ils se basent sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature : croisement, mutation, sélection (Néo-Darwinisme) [Holland, 1975]. Le but de ces algorithmes est d'obtenir une solution approchée à un problème d'optimisation dans un temps raisonnable : des solutions satisfaisantes au mieux à différents critères, souvent contradictoires. Ils sont utilisés dans le cas où il n'existe pas de méthode exacte pour trouver une solution au problème.

Les algorithmes génétiques sont capables de résoudre un problème sans aucune connaissance préalable de ce dernier. Ils peuvent être représentés par une boîte noire comportant des entrées et des sorties. Un tel algorithme ne fait que manipuler les entrées, lire les sorties, manipuler à nouveau les entrées afin d'améliorer les sorties, et ainsi de suite.

b) Réseau de neurones

Le réseau de neurones est un paradigme d'intelligence artificielle inspiré du raisonnement humain. C'est une tentative de modélisation mathématique du cerveau humain, proposée par Mac Culloch et Pitts au début des années 40 [Culloch et Pitts,

1943]. Le cerveau humain, structure extrêmement complexe, composé d'un ensemble de neurones liés par des synapses peut renfermer jusqu'à plusieurs milliards de neurones connectés les uns aux autres. Chaque neurone est réactif et n'effectue qu'un traitement très basique de l'information en fonction des stimulations qu'il reçoit (Figure 1.6).

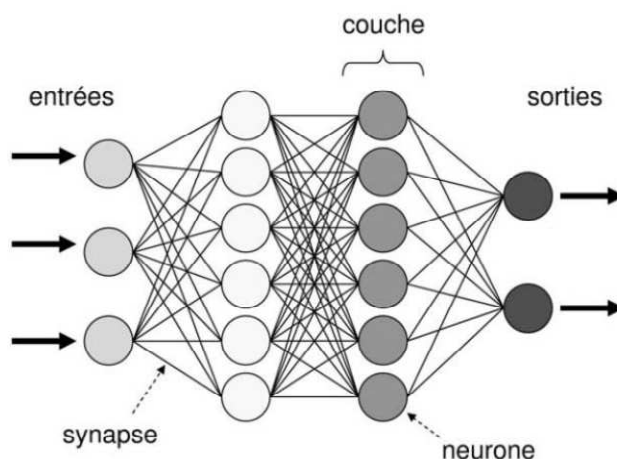


Figure 1.6 : Exemple schématisé de réseau de neurones [Baptiste, 2011]

Les réseaux de neurones sont basés sur une unité simple, le neurone *formel*, capable de réaliser quelques opérations élémentaires. Cette unité élémentaire est ensuite attachée à d'autres unités créant ainsi le réseau de neurones dont la puissance de calcul est clairement supérieure. Le réseau est constitué de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique en se basant sur les informations qu'il reçoit. Les neurones traitent exclusivement des données numériques et non pas symboliques.

Le neurone formel réalise une somme pondérée de ces entrées, puis il ajoute un seuil à cette somme et fait passer le résultat en utilisant une fonction de transfert pour obtenir sa sortie [Denis et Gilleron, 2013].

c) Logique floue

La logique floue (*fuzzy logic*, en anglais) est un outil de l'intelligence artificielle, formalisé par Zadeh en 1965. C'est une extension de la logique classique aux raisonnements approchés et permet le traitement de données imprécises ou incomplètes. La logique floue apparaît comme une logique graduelle inspirée de notre perception nuancée du monde en se basant sur une fonction d'appartenance qui prend des valeurs entre 0 et 1, [Zadeh, 1968, 1973].

La logique floue est basée sur trois phases principales : la *fuzzification* chargée de la transformation des valeurs d'entrées en grandeurs floues, l'inférence avec la base de règles, tel que si ... alors et qui prend les décisions et enfin la phase de *défuzzification* qui transforme les grandeurs floues en grandeurs classiques.

d) Méthodes d'agrégation multicritères

Un modèle multicritères a pour but la résolution des problèmes d'aide à la décision multicritères. C'est une structure hiérarchique qui représente la décomposition d'un problème en sous-problèmes, et donc plus facile à résoudre que le problème global. Il s'agit de trouver la "solution la plus adéquate", compte tenu d'un certain ensemble de critères, la dite solution pouvant prendre diverses formes (choix, affectation, classement).

Nous citons l'exemple de la méthode DEXi, qui est une interface d'évaluation multicritère de la durabilité des systèmes de production innovants. C'est un outil d'aide à la décision basé sur l'approche multicritère. Il se compose de plusieurs critères : les variables qualitatives composant les différents sous-problèmes, les échelles de valeurs de ses variables, l'arbre des attributs qui est la structure hiérarchique présentant la décomposition du problème et les règles de décision utilisés dans la méthode. Pour plus de détails, nous renvoyons le lecteur à [Bohanec, 2008].

e) Comparaison des méthodes

Nous présentons un tableau comparatif des avantages et des inconvénients de chacune de ces méthodes (l'étude a été réalisée par [Melfort, 2010]).

	Avantages	Inconvénients
algorithmes génétiques (AG)	- Efficace pour une gamme de problèmes très large - Faculté d'adaptation, réactivité, prise en compte de l'environnement - Pour des espaces de recherche important (pas de parcours exhaustif)	- Problème de la boîte noire - Choix de la fonction d'évaluation délicat - Paramètres délicats à régler (probabilité des opérateurs) - Solution trouvée n'est pas forcément la meilleure (approximation de la solution optimale) - Calcul d'un grand nombre de <i>fitness</i> pour garantir la robustesse de l'algorithme génétique
réseaux de neurones (RN)	- Représentation de n'importe quelle dépendance fonctionnelle - Grande variabilité de la variable de sortie (oui/non, valeur continue, ...) - Robuste si peu de données - Pas de recodage des données	- Problème de la boîte noire - Beaucoup d'exemples à entrer, mise en œuvre par apprentissage assez longue - Temps de modélisation long (parfois plusieurs heures)
Logique	- Pas besoins de modèle - Implémentation de	- Manque de directives précises pour la conception

floue (LF)	connaissances linguistique - Maîtrise de système à comportement complexe (non linéaire)	- Approche non systématique - Possibilité d'apparition de règles d'inférence contradictoire - Impossibilité de démonstration de la stabilité du modèle
Méthodes d'agrégation multicritères (MAM-DEXi)	- Facile à mettre en oeuvre - Règles de décision intuitives de type si ... alors - Implémentation de connaissances linguistiques	- Agrégation de trois attributs maximum - Fastidieux à construire, besoin de temps pour construire le modèle - Nécessité d'expertise conséquente pour développer l'application

Tableau 1.1 : Avantages et inconvénients des méthodes de développements des systèmes experts [Melfort, 2010].

D'après le tableau comparatif, nous pouvons signaler que les algorithmes génétiques et les réseaux de neurones apparaissent comme étant moins appropriés pour le développement des systèmes experts. En effet ils sont basés sur le principe de la boîte noire. Cette dernière ne permet pas d'ordonner correctement le savoir expert en grands domaines de sélection. La logique floue semble plus difficile à mettre en place par l'apparition de règles d'inférence contradictoire. Les réseaux de neurones sont fondés sur l'apprentissage et nécessitent un nombre d'exemples conséquents. Enfin pour la méthode d'agrégation multicritère de DEXi, elle a besoin de temps pour construire le modèle et il nécessite une expertise conséquente pour développer l'application.

1.4.2.3 Limites et bilan

Pour certains domaines, les performances des systèmes experts sont satisfaisantes. En effet les systèmes experts permettent de regrouper et d'intégrer les connaissances de plusieurs experts. Ils permettent aussi d'appliquer les règles de façon systématique et d'être utilisés par des personnes qualifiées de *non expertes* dans le domaine. Toutefois les systèmes experts connaissent encore d'importantes limites telles que :

- l'explosion rapide de la taille de la base de règles ;
- la difficulté de construction et de maintenance de la base de connaissances ;
- l'incapacité d'apprendre automatiquement ;
- la difficulté de vérification de la cohérence des règles entre elles, par conséquent, la détection des erreurs dans les règles est assez difficile ;
- la dégradation brutale des performances du système aux frontières de sa base de connaissances ;
- la prise en compte des seules données déclaratives, et non pas des données comportementales enregistrées dans le système d'information.

1.5.2 Modèles de Markov cachés

Un modèle de Markov caché (MMC) est un modèle statistique modélisant un système dynamique supposé être un processus markovien de paramètres inconnus. Les modèles de Markov cachés sont utilisés dans plusieurs domaines tels que la reconnaissance de formes ou le traitement automatique du langage naturel. Il permet de modéliser un système dont les états sont cachés et où seules les observations sont accessibles [Lawrence et Rabiner, 1990].

Un modèle de Markov caché peut être modélisé graphiquement par un réseau bayésien [Baptiste, 2011]. Il est décrit par l'ensemble des éléments suivants :

- N représente le nombre d'états du modèle ;
- M est le nombre total de symboles d'observation distincts ;
- T décrit la longueur de la séquence d'observation ;
- q_t désigne l'état dans lequel se trouve le système à l'instant t ;
- $S = \{S_1, S_2, \dots, S_N\}$ est l'ensemble des états.

La figure suivante (figure 1.8) représente un modèle de Markov caché modélisé via un réseau bayésien.

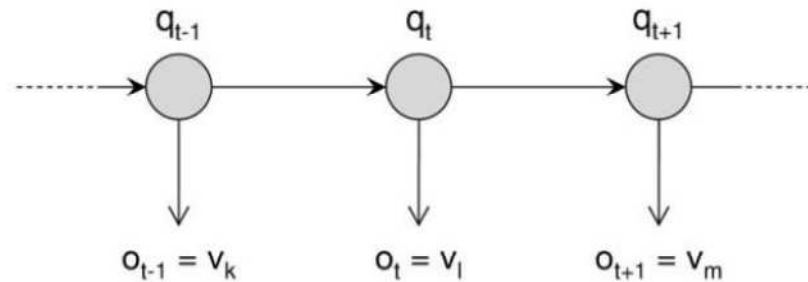


Figure 1.8 : Réseau bayésien pour un modèle de Markov caché [Baptiste, 2011]

Le réseau (figure 1.8) montre la dépendance conditionnelle états-états et états-observations d'un modèle markovien caché. Les états ne sont pas directement observables (ils sont supposés cachés). Pour v_1 , ils sont détectés lorsque l'on se trouve dans l'état q_t à l'instant t . L'observation $o_t = v_1$ réalisée à l'instant t est liée à q_t par une probabilité conditionnelle.

- $Q = \{q_1, q_2, \dots, q_T\}$ représente la séquence des états de l'observation ;
- $V = \{v_1, v_2, \dots, v_M\}$ c'est l'ensemble des symboles d'observation possibles ;
- $\pi = \{\pi_i\}$ est la distribution initiale : la probabilité d'être à l'état S_i à $t = 1$, avec $\pi_i = P[q_1 = S_i]$, $1 \leq i \leq N$;
- A désigne la matrice de transition, dans ce cas $\{a_{ij}\}$ est la probabilité d'être à l'état S_j à l'instant $t + 1$, avec S_i est l'état i à l'instant t , donc on aura : $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$, $1 \leq i \leq N$, $1 \leq j \leq N$
- B définit la matrice d'observation, dans ce cas $\{b_{jk}\}$ est la probabilité d'observer le symbole v_k lors de l'état S_j , avec b_{jk} comme suit :

$b_{jk} = P[v_k \text{ à l'instant } t | q_t = S_j], 1 \leq j \leq N, 1 \leq k \leq M;$

o_t désigne le symbole observé à l'instant t .

Les modèles de Markov cachés sont généralement utilisés pour résoudre l'un des deux types de problèmes [Sean, 1996] :

1. En partant d'un modèle λ , quelle est la probabilité que ce modèle ait généré une séquence d'observation $O = \{o_1, o_2, \dots, o_T\}$? Cela revient donc à calculer la probabilité $P[O|\lambda]$.
2. Supposons qu'un système a émis une séquence d'observation O et étant donné son modèle de Markov caché λ , alors quelle est dans ce cas la séquence d'états $Q = \{q_1, q_2, \dots, q_T\}$ correspondante ?

Pour plus de détails sur le modèle de Markov caché, nous renvoyons le lecteur à [Olivier, 2006], [Baptiste, 2011].

1.5.3 Réseaux bayésiens

Un réseau bayésien (RB) est un modèle probabiliste graphique capable d'acquérir, de capitaliser et d'exploiter des connaissances. Il est devenu un formalisme de plus en plus utilisé pour modéliser les systèmes complexes. Un RB est un modèle décrivant un ensemble de variables aléatoires sous la forme d'un graphe orienté acyclique et leurs probabilités conditionnelles. Un réseau bayésien est à la fois un modèle de représentation des connaissances, et une *machine à calculer* les probabilités conditionnelles. Un RB est adapté à la prise en compte de l'incertitude. Il sert à représenter la connaissance que l'on a d'un système, ou à découvrir cette connaissance en analysant des données par apprentissage. Le réseau bayésien permet de :

- diagnostiquer : dans ce cas on observe des effets observés et on cherche à déduire la répartition de probabilité sur les causes possibles ;
- prendre des décisions (le RB est un système d'aide à la décision) ;
- simuler le comportement du système étudié.

Un réseau bayésien est un graphe probabiliste orienté, chaque nœud comporte un fait associé à une variable aléatoire. Un arc entre deux nœuds représente l'influence d'une variable sur une autre. Le graphe est complété par les tables de probabilités conditionnelles pour chacune des variables aléatoires du réseau [Baptiste, 2011]. La figure 1.9 représente un exemple de réseau bayésien. Dans cet exemple les variables aléatoires choisies sont simples : elles peuvent prendre comme valeur *vrai* ou *faux*. L'exemple montre l'influence de la pluie sur l'arrosage automatique et celui de ces deux facteurs sur le fait que la pelouse soit mouillée.

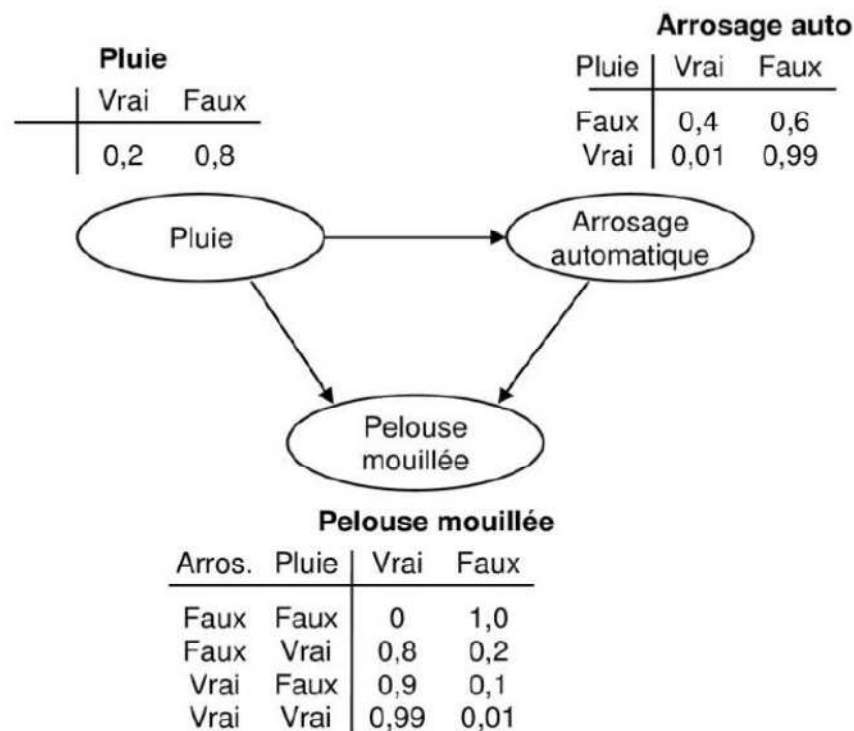


Figure 1.9 : Exemple de réseau bayésien [Baptiste, 2011]

Dans le tableau associé à pelouse mouillée, la deuxième ligne veut dire que : " *s'il n'y a pas l'arrosage automatique et qu'il pleut, la pelouse a une probabilité $p[PM|A \cap P] = 0,8$ d'être mouillée.* "

Les arcs du réseau sont orientés afin d'interpréter les probabilités conditionnelles dans le sens logique de la causalité. Toutefois, le théorème de Bayes stipule que :

$P[A|B] = (P[A] / P[B]) P[B|A]$, c'est-à-dire que la connaissance de la probabilité conditionnelle $P[A|B]$ permet de déduire $P[B|A]$. Par conséquent, nous pouvons estimer la probabilité qu'il pleuve en observant l'état de l'arrosage automatique.

Réseaux bayésiens dynamiques

Les réseaux bayésiens sont des graphes acycliques statiques qui représentent uniquement des probabilités conditionnelles. Ils ne sont pas capables de modéliser des causalités temporelles. Dean [Thomas et Keiji, 1989] propose les réseaux bayésiens dynamiques afin de prendre en compte l'aspect temporel et dynamique des situations.

Les Réseaux Bayésiens Dynamiques (RBD) permettent de représenter l'évolution des variables dans le temps. Un RBD contient une suite de tranches de temps, où chaque tranche contient des variables décrivant l'état de l'environnement correspondant à ce laps de temps [Baptiste, 2011]. Chaque tranche est en soi un réseau bayésien, donc la même structure de réseau bayésien est répliquée au fur et à mesure que le temps avance et le système évolue. Les arcs reliant les différentes tranches entre elles représentent l'aspect dynamique et temporel du système modélisé. Enfin, chaque tranche possède des nœuds d'observation qui modélisent les entrées du modèle sur

l'état courant de l'environnement. L'exemple le plus simple d'un RBD est le modèle de markov caché. Il est formé de deux nœuds, un représente l'état du système et l'autre le nœud d'observation.

Le réseau bayésien est utilisé dans plusieurs types d'applications statistiques et industrielles. Par exemple le RB a été utilisé pour la prédiction de tremblements de terre [Ferraes, 1986], pour le comportement de robots [Lazkano et al., 2007] ou pour la gestion de réseaux de capteurs [Matthew et al., 2009].

1.5.5 Limites et bilan

Les réseaux de Pétri exigent des connaissances a priori sur les situations modélisées et il est difficile de faire évoluer un réseau de Pétri afin qu'il prenne en compte de nouvelles situations, donc ils n'évoluent pas automatiquement avec les modifications progressives du système. Une autre difficulté réside dans l'apprentissage des modèles ce qui rend la participation d'un expert nécessaire. Toutefois, dans notre contexte, nous souhaitons concevoir un outil capable de s'adapter automatiquement aux évolutions du système étudié, donc les réseaux de Pétri ne sont pas adaptés pour répondre à notre problématique.

Les modèles de Markov cachés sont limités par plusieurs contraintes [Kadous et Sammut, 2002]. En effet l'apprentissage du modèle MMC est un processus très difficile et complexe. Il nécessite la définition d'un grand nombre de paramètres. Cela demande un temps de calcul très important ainsi qu'une base d'apprentissage conséquente. De plus il n'existe pas de méthode capable de déterminer à l'avance le nombre d'états et de transitions du modèle. Enfin les modèles MMC sont limités pour la reconnaissance de nouvelles séquences. Par exemple, pour le développement d'un système de reconnaissance de l'écriture, il est difficile de prévoir tous les styles et formes d'écriture d'une même lettre. Par conséquent le système dynamique auquel nous nous intéressons, anticipation d'une situation dynamique en se basant sur les expériences passées, n'est pas markovien.

Les réseaux bayésiens sont des outils de modélisation applicables à un système décrit par un ensemble de variables aléatoires. Avant de procéder à l'inférence, il est nécessaire de construire d'abord le réseau bayésien, ce qui nécessite de définir sa topologie et les probabilités conditionnelles. Autrement dit, il faut au préalable connaître les variables aléatoires, leurs valeurs possibles et leurs interdépendances.

Les réseaux de neurones sont utilisés dans plusieurs domaines et plus particulièrement pour des systèmes très peu évolutifs (voire qui n'évolue pas). Ils sont très utilisés aussi comme outil de classification, par exemple, la reconnaissance de caractères ou les prévisions météo. Le principal inconvénient des RN réside dans la phase d'apprentissage. Malgré la performance des algorithmes d'entraînement, cette phase reste très longue. De plus, la construction de la base d'apprentissage demande l'intervention d'un expert du domaine, donc les réseaux de neurones ne sont pas les mieux adaptés pour répondre à notre problématique.

Dans la section suivante, nous présentons un autre outil d'aide à la décision dynamique fondé sur les expériences passées et les connaissances : le raisonnement à partir de cas.

1.6 Les systèmes fondés sur les expériences passées : le raisonnement à partir de cas

Le Raisonnement à Partir de Cas (RàPC), où “*Case-Based Reasoning*” en anglais (CBR), est un paradigme d'intelligence artificielle capable de résoudre un problème, appelé cas cible, en s'appuyant sur la base de cas, un cas étant un problème déjà résolu (cas sources) [Schank, 1982] [Kolodner, 1988].

Le raisonnement à partir de cas, contrairement aux autres outils d'aide à la décision, ne nécessite pas une connaissance exhaustive du domaine afin de résoudre les problèmes. En effet, dans le cas du RàPC, les connaissances importantes et utiles à la résolution du problème sont contenues dans les cas. Ces connaissances sont des expériences de résolution des problèmes passées stockées en mémoire.

Les systèmes de raisonnement à partir de cas sont des systèmes évolutifs. En effet, les systèmes RàPC acquièrent de nouvelles connaissances ou encore les expériences passées au fur et à mesure qu'ils résolvent les problèmes.

Le raisonnement à partir de cas est utilisé dans de multiples domaines d'applications tels que l'enseignement, la médecine, les disciplines juridiques, la sécurité routière, la maintenance ou l'analyse financière. Le RàPC est utilisé également pour plusieurs types de problèmes : la planification, la classification, le diagnostic, la gestion des connaissances, l'aide à la décision qui représente justement notre centre d'intérêt. Nous étudions de manière approfondie le raisonnement à partir de cas dans le chapitre suivant.

D'autres méthodes, dérivées du RàPC intègrent davantage les expériences. Nous pouvons citer par exemple le Raisonnement à partir d'exemples, le Raisonnement à Partir de Trace et le Raisonnement par analogie.

- **Raisonnement à Partir d'Exemples :** dans ce type de raisonnement, les concepts sont organisés sous forme de classe d'exemples. La résolution d'un problème, c'est-à-dire d'un nouveau exemple non classifié, consiste à la recherche de la classe qui correspond le mieux à l'exemple. La classe des solutions représentent donc la solution pour l'exemple traité. Dans cette catégorie de raisonnement, les solutions ne sont pas modifiées [Benjamin et Matthieu, 2004].
- **Raisonnement à Partir de Trace :** c'est l'un des cas particuliers des RàPC, où l'une des variantes du RàPC, qui intègrent davantage l'idée d'expérience. Le raisonnement à Partir de Trace (RàPT) vise à capturer la connaissance à partir de l'expérience, en se basant sur l'expérience humaine, définie par des

situations temporelles. Il est représenté par des enregistrements temporels appelés *traces*. Ces enregistrements permettent de modéliser l'expérience et d'être réutilisés en s'appuyant sur le processus de raisonnement à partir de cas [Mille et al., 2006].

- **Raisonnement par analogie** : il est caractérisé par le fait de faire des analogies entre des cas différents sans tenir compte du domaine auquel appartiennent le problème cible et les expériences. C'est un raisonnement orienté vers le problème de la représentation des cas et les moyens de faire ces analogies par association d'idées, corrélations et transferts. Pour plus de détails, nous renvoyons le lecteur à [Coulon et al., 1990].

1.7 Conclusion

Nous souhaitons concevoir un outil d'aide à la décision capable d'anticiper une situation dynamique en se basant sur les expériences passées, capable aussi de s'adapter automatiquement aux évolutions du système étudié. Dans ce chapitre, nous avons présenté les systèmes d'aide à la décision couramment utilisés pour la modélisation et le suivi de situations dynamiques. Ces outils jouent un rôle primordial et décisif dans la prise de décision. En effet ils permettent au décideur de mieux gérer la masse et la complexité des données et informations. Pour chacun de ces systèmes, nous avons discuté de l'applicabilité à notre problématique. Nous en tirons les conclusions suivantes.

Les systèmes experts connaissent d'importantes limites telles que la difficulté de la maintenance de la base de connaissances, la difficulté aussi de la vérification de la cohérence des règles entre elles, par conséquent, la détection des erreurs dans les règles est assez difficile. Pour les systèmes d'aide à la décision à base de modèles, nous avons montré que ces méthodes ne sont pas les plus adaptées pour répondre à notre problématique. En effet l'apprentissage automatique de ces modèles est complexe et long (RP, RN, RDB, MMC) et l'intervention d'un expert est souvent sollicité pour participer à la conception des modèles.

Parmi les méthodes à base d'expérience, le raisonnement à partir de cas, semble le plus apte à répondre à notre problématique. En effet, contrairement aux autres systèmes et outils de résolution des problèmes en SAD, le RàPC ne nécessite pas une connaissance exhaustive du domaine afin de résoudre les problèmes. Les systèmes de raisonnement à partir de cas sont des systèmes évolutifs. Ils acquièrent de nouvelles connaissances au fur et à mesure qu'ils résolvent les problèmes.

Nous consacrons le chapitre suivant au Raisonnement à Partir de Cas.

Chapitre 2 : Raisonnement à Partir de cas

2.1 Motivation à l'utilisation du RàPC.....	37
2.2 Historique.....	37
2.3 Communautés en RàPC.....	38
2.4 Raisonnement à Partir de Cas.....	39
2.4.1 Principe RàPC	39
2.4.3.1 Phase d'élaboration	41
2.4.3.2 Phase de remémoration.....	42
2.4.3.3 Phase d'adaptation.....	43
2.4.3.4 Phase de validation.....	44
2.4.3.5 Phase d'apprentissage.....	45
2.4.4 Cas / Base de cas	46
2.4.4.1 Définition du cas.....	46
2.4.4.2 Structure d'un cas	46
2.4.4.3 Indexation du cas.....	48
2.4.4.4 Base de cas dans les systèmes de RàPC	48
2.4.5 Connaissances	49
2.4.6 Domaines d'application et conditions d'utilisation du RàPC.....	50
2.4.6.1 Typologies d'applications	50
2.4.6.2 Conditions d'applications.....	52
2.5 Raisonnement à partir de cas pour situations statiques	54
2.5.1 La similarité entre les cas en situations statiques	54
2.5.2 L'adaptation en situations statiques.....	55
2.5.3 Système CHEF	55
2.6 Raisonnement à partir de cas pour situations dynamiques	56
2.6.1 Représentation dynamique des cas.....	57
2.6.1.1 Représentation d'un historique.....	57
2.6.1.2 Paramètres cible et source	64
2.6.1.3 Paramètres passé et futur	64
2.6.1.4 Rôle du temps.....	65
2.6.2 Base de cas pour les RàPCD	66
2.6.2.1 Organisation de la base de cas pour RàPCD	66

2.6.2.2 Maintenance de la base de cas.....	66
2.6.3 Le cycle de vie de RàPCD.....	68
2.6.4 Cycle du raisonnement à partir de cas pour les situations dynamiques.....	69
2.6.4.1 Cycle du raisonnement statique pour RàPCD	69
2.6.4.2 Mesures de similarité dans le cas dynamique et cycle statique	70
2.6.5 Etude comparatif des principaux systèmes RàPCD	74
2.6.5.1 REBECAS	74
2.6.5.2 AuRA.....	76
2.6.5.3 SINS	77
2.6.5.4 CASEP2.....	77
2.6.5.5 BROADWAY	79
2.6.5.6 SAPED	80
2.6.6 Synthèse comparative.....	81
2.7 Conclusion.....	84

2.1 Motivation à l'utilisation du RàPC

Le RàPC est une approche d'intelligence artificielle qui permet de résoudre de nouveaux problèmes en se basant sur les expériences passées ou encore sur des problèmes résolus disponibles en mémoire [Schank, 1982] [Kolodner, 1988]. L'outil RàPC a pour objectif de capitaliser les expériences passées sous forme de connaissances dans sa mémoire [Haouchine, 2009]. Ce choix du RàPC pour notre travail est justifié par les points suivants :

- Le RàPC est un outil de gestion des connaissances du domaine étudié, capable de réutiliser ces connaissances dans des cas analogues pour prendre des décisions et d'enrichir la mémoire en ajoutant de nouvelles connaissances d'une manière dynamique ;
- Le RàPC est une approche qui permet de raisonner dans un domaine à connaissance réduite (sur un ensemble restreint de données et de connaissances du domaine), au départ, les connaissances du domaine sont a priori insuffisantes pour créer une base de connaissance complète. Toutefois, le RàPC est capable de raisonner avec cette base réduite [Main et al., 2000] ;
- L'approche RàPC est capable de raisonner avec des données imprécises ou incomplètes (la description du problème à résoudre est incomplète ou n'est pas exactement similaire aux problèmes résolus disponibles en mémoire) ;
- Le système que nous voulons mettre en place dispose de peu d'expériences et puisque l'approche RàPC peut être développée avec un ensemble limité de cas, alors l'acquisition incrémentale de nouvelles expériences est possible durant le fonctionnement du système. Cette acquisition permet l'ajout de nouveau cas jugé pertinent dans la base de connaissance du système, et d'améliorer par la suite les résultats fournis par ce dernier.

2.2 Historique

Le paradigme RàPC trouve ses origines dans des travaux en sciences cognitives sur la mémoire humaine à la fin des années 70, plus particulièrement sur la théorie de mémoire dynamique, d'abord par les travaux de Marvin Minsky [Minsky, 1974] sur la notion de *frame*, puis dans les travaux de Roger Schank [Schank, 1982] sur le modèle de mémoire dynamique. Ces travaux permettent de dépasser les limites rencontrées par les systèmes experts, ainsi que les limites dues à la connaissance exclusivement sous forme de règles [Ivana, 2006].

En 1974 Minsky [Minsky, 1974], dans le but de représenter les connaissances, a développé le concept de *frame*. C'est une structure de données fonctionnelles utilisée pour subdiviser la connaissance en sous-structures représentant des situations stéréotypées [Minsky, 1988]. Elle est constituée à partir d'expériences précédentes pour comprendre de nouvelles situations et pour agir. Les frames sont reliés entre eux pour former une idée complète. Elle contient de l'information sur la manière d'utiliser

le frame, sur ce qu'on peut en attendre suite à son utilisation, et ce qui peut être fait si ces attentes ne sont pas satisfaites. Parallèlement à l'évolution des modèles de représentation des connaissances, les techniques de raisonnement évoluent. Schank [Schank, 1982] se base sur les travaux de Minsky pour proposer au début des années 80 l'idée du raisonnement à partir de cas. L'auteur propose le concept de "mémoire dynamique", qui se base sur l'utilisation de scripts pour expliquer des situations, et l'utilisation d'expériences dans la construction de ses scripts. La mémoire peut être vue comme un réseau dense d'expériences. Selon Schank on ne raisonne pas à partir de règles générales, de connaissances abstraites, mais directement à partir de cas et en utilisant des analogies. Plusieurs chercheurs ont suivi ses travaux et la communauté s'est orientée vers le développement des systèmes à base de RàPC et leurs applications dans divers domaines.

La recherche dans le domaine du RàPC se poursuit dans les années 80, d'abord aux Etats-Unis avec les conférences DARPA initiées en 1988 par Kolodner [Kolodner, 1988], avant de s'imposer au niveau européen avec la première conférence en 1993 à Kaiserslautern [Richter et al., 1993], puis au niveau international, avec la première conférence internationale à Lisbonne en 1995 [Veloso et al., 1995]. En 1993, Kolodner [Kolodner, 1993] a développé l'un des premiers systèmes utilisant le RàPC nommé CYRUS. Ce système retrace la vie politique de l'ancien secrétaire d'état américain Cyrus Vance. Il fonctionne selon un processus de questions/réponses. Il est doté de connaissances sur les différents voyages de Cyrus Vance. Kolodner a développée ce système en se basant sur le modèle de mémoire dynamique de Schank Ce modèle a ensuite servi de base pour plusieurs systèmes, comme CHEF [Hammond, 1986], CASEY [Koton, 1988], PROTOS [Bareiss et al., 1993], ICARUS [Varma, 1999], CASSIOPEE [Bergmann et al., 2003], CREEK [Aamodt, 2004].

2.3 Communautés en RàPC

La méthode raisonnement à partir de cas se situe dans le domaine de l'intelligence artificielle et dans certains cas particuliers dans le domaine de l'ingénierie des connaissances selon le type de méthode développée (figure 2.1) [Haouchine, 2009]. L'ingénierie des connaissances est l'intersection de deux communautés de recherche : l'intelligence artificielle et les sciences cognitives.

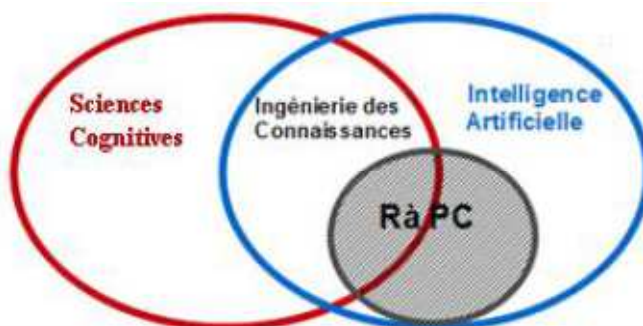


Figure 2.1 : Communautés en raisonnement à partir de cas

L'intelligence artificielle est souvent abrégée par le sigle IA, est une discipline de recherche permettant d'élaborer des systèmes intelligents (programmes informatiques capable de produire un comportement intelligent, qui est pour l'instant, accomplie de façon satisfaisante par des êtres humains car il demande des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, une forte organisation de la mémoire et le raisonnement critique, et enfin ces programmes doivent être capable d'éprouver une impression d'une réelle conscience de soi). L'IA est la recherche de moyens susceptibles de doter les systèmes informatiques de capacités intellectuelles comparables à celles des êtres humains [CNTRL, 1979].

Les sciences cognitives correspondent à l'ensemble des disciplines scientifiques consacrées à l'étude et la compréhension des mécanismes de la pensée humaine, animale ou artificielle, et de manière générale tout système cognitif, c'est-à-dire tout système complexe de traitement de l'information capable d'acquérir, conserver, utiliser et transmettre des connaissances [Wikipédia, 2009].

L'ingénierie des connaissances est l'étude des techniques, des concepts et méthodes permettant de formaliser, de modéliser et d'acquérir les connaissances pour des systèmes réalisant ou aidant les humains à accomplir des tâches [Charlet et al., 2000].

2.4 Raisonnement à Partir de Cas

2.4.1 Principe RàPC

Le raisonnement à partir de cas est une approche qui s'appuie sur les expériences passées, appelées **cas sources**, pour résoudre un nouveau problème appelé **cas cible**. Les cas sources sont stockés dans la base de cas ou mémoire du système. Un nouveau problème est résolu en cherchant des cas similaires dans la mémoire, et en l'adaptant à la nouvelle situation. Ce principe de résolution peut être réalisé par un processus, décrit par un ensemble d'étapes séquentielles (élaboration, remémoration, adaptation, révision, apprentissage), que l'on représente par un cycle [Aamodt et Plaza, 1994] [Mille, 2006].

2.4.2 Carré d'analogie

L'approche RàPC permet de résoudre un nouveau problème à partir de problèmes déjà résolus, par analogie. Le RàPC peut donc être représenté par le carré d'analogie (figure 2.2) [Mille et al., 1996]. La solution recherchée (solution cible) est au problème cible ce que la solution source est au problème source. Ce carré d'analogie explique :

- d'une part le lien entre la description d'un cas et sa solution.
- d'autre part, les liens entre la description et la solution du cas source (base de cas) et du cas cible (problème à résoudre). Dans ce cas on parle alors de l'adaptation de solution source au problème cible en fonction de la similarité et

de l'adaptation des descripteurs des cas sources similaires de la base de cas au cas cible.

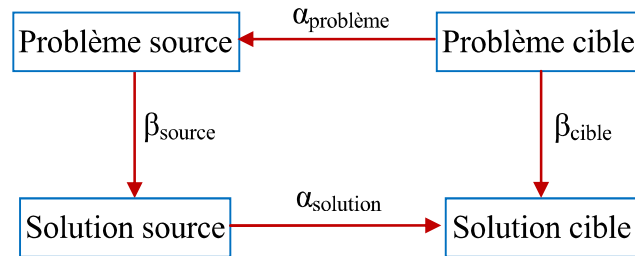


Figure 2.2 : Carré d'analogie

α : similarité entre deux problèmes (cas), β : trace du raisonnement menant à la solution

Un cas source est sélectionné à partir des valeurs de descripteurs du problème cible en se basant sur la mesure de similarité α . Ensuite nous avons les relations de dépendance β entre les valeurs de descripteurs de problème et les valeurs de descripteurs de la solution, qui permettent de mettre en évidence les descripteurs de la solution afin de l'adapter. Les descripteurs de la solution dépendent des descripteurs de problèmes sources qui sont différents des descripteurs de problèmes cibles [Haouchine, 2009].

En fonction de ces dépendances et des écarts α constatés à modifier et à corriger, l'adaptation permet de fournir une solution cible candidate qui pourra être examinée par rapport à sa conformité aux dépendances particulières qui pourraient exister entre problème et solution cible.

2.4.3 Cycle du RàPC

Le RàPC est capable de réutiliser les connaissances des expériences précédentes dans des nouvelles situations, en s'appuyant sur un cycle de raisonnement composé de cinq phases. Aamodt et Plaza [Aamodt et Plaza, 1994] sont les premiers auteurs à avoir décrit le cycle du RàPC. Selon ces auteurs le RàPC se compose de quatre phases : la première phase est la remémoration ou la recherche du cas similaire, ensuite l'adaptation ou la réutilisation du cas retrouvé, ensuite la validation ou la révision du cas sélectionné et enfin la phase de mémorisation ou encore l'apprentissage. Pour certain auteur le cycle du RàPC est en trois phases seulement à savoir la remémoration, l'adaptation et la mémorisation [Fuchs et al., 2006]. D'autres auteurs en RàPC ajoutent une phase préliminaire d'élaboration au début du cycle [Mille, 2006]. Dans notre thèse nous exploitons le cycle de RàPC composé de cinq phases (figure 2.3).

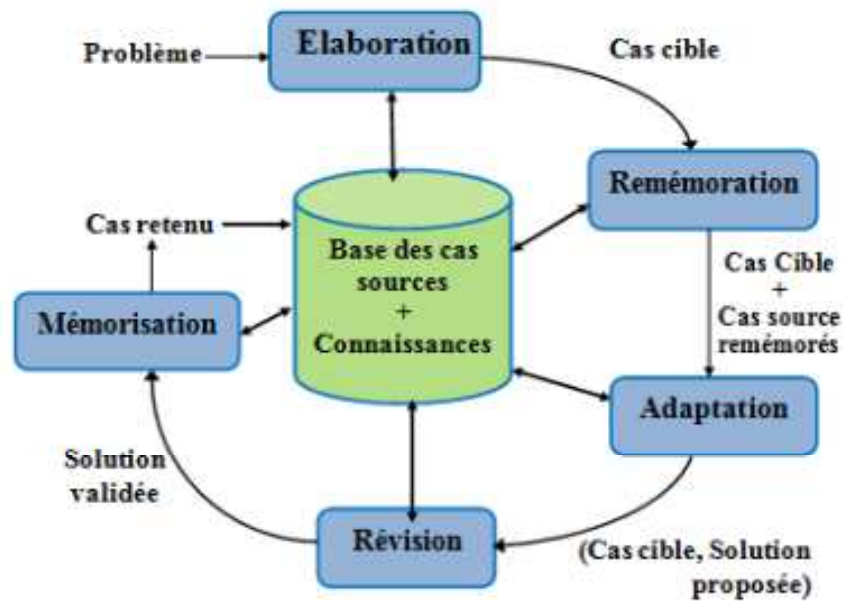


Figure 2.3 : Le cycle de raisonnement à partir de cas

2.4.3.1 Phase d'élaboration

Souvent, lorsqu'un problème est posé, les connaissances à son sujet peuvent être de nature très différentes. Une partie de ces connaissances peuvent être totalement étrangères, alors que d'autres, peuvent être absentes ou mal posées et pourtant ces connaissances sont indispensables pour permettre d'envisager une résolution [Haouchine, 2009]. Cette phase du raisonnement a pour rôle de mettre en place différents mécanismes pour passer d'un problème souvent mal exprimé à un problème correctement défini.

L'élaboration du cas cible consiste à compléter et/ou filtrer la description d'un problème en se basant sur les connaissances du domaine. Ces opérations permettent d'obtenir une description complète du cas cible à partir d'une description éventuellement incomplète. Ces opérations permettent aussi de pondérer les descripteurs du problème cible en fonction des dépendances identifiées entre les descripteurs du problème cible et les descripteurs de la solution recherchée. Dans cette étape on peut utiliser l'ontologie du domaine afin de compléter les descripteurs importants. Fuchs et al. [Fuchs et al., 2006] ont proposé une formalisation de la phase d'élaboration suivante : une étape qui consiste, à partir de l'entrée du système de RàPC, à construire le problème cible.

Cette phase qui est le point de départ du cycle de RàPC a une grande importance, car elle permet de faciliter la recherche d'un cas similaire au problème posé et de l'orienter vers une solution adaptable. Donc une bonne élaboration du cas cible facilite l'exécution et l'évaluation des différentes phases du cycle du raisonnement.

2.4.3.2 Phase de remémoration

La phase de remémoration consiste à rechercher dans la base de cas le ou les cas sources les plus similaires au cas cible. Cette recherche se base sur les descripteurs de la partie problème du cas cible à résoudre et des cas sources stockés dans la mémoire. Cette étape permet d'obtenir la bonne solution en procédant comme suit : l'identification des caractéristiques pertinentes du problème, ensuite la remémoration et la sélection des meilleurs cas parmi les cas sources [Haouchine, 2009]. La réalisation de ces tâches est essentiellement dépendante de la structure de cas, de leur indexation et de l'organisation de la base de cas.

La recherche des cas sources similaires se base sur la mesure de similarité. Le choix d'une mesure de similarité est déterminant pour l'efficacité du RàPC. Dans la sous-section suivante, nous présentons les mesures de similarité.

Mesures de similarité

Les mesures de similarité peuvent être locales ou globales:

Similarités locales : dans ce cas les mesures sont établies au niveau des caractéristiques du cas. Ces mesures dépendent du type des descripteurs (numérique, symbolique, taxonomique) et des rangs des valeurs des descripteurs [Haouchine, 2009]. Généralement le calcul des similarités locales est basé sur la notion de la distance :

- pour les valeurs de descripteurs numériques : $sim_i(d_i^c, d_i^s) = 1 - \frac{d_i^c - d_i^s}{range}$
- pour les valeurs de descripteurs symboliques : $sim_i(d_i^c, d_i^s) = \begin{cases} 1 & \text{si } d_i^c = d_i^s \\ 0 & \text{si } d_i^c \neq d_i^s \end{cases}$
- pour les taxonomiques $sim_i(d_i^c, d_i^s) = \frac{h(\text{commonnode}(d_i^c, d_i^s))}{\min(h(d_i^c), h(d_i^s))}$

Où d_i^c, d_i^s : sont des valeurs des descripteurs respectivement du cas cible et cas sources, *range* est la valeur absolue de la différence entre la borne supérieure et la borne inférieure de l'ensemble des valeurs, *h* est le nombre de niveaux dans l'hérarchie (l'arbre taxonomique).

Pour le calcul de la similarité locale, une distinction est faite entre : la similarité de surface qui ne prend en compte que les attributs des cas et la similarité structurelle qui s'intéresse aux relations entre les attributs des objets.

Similarités globales dans ce cas le système RàPC calcule la similarité globale entre le cas cible et les cas sources de la base de cas, par agrégation de similarités locales [Haouchine, 2009]. Il n'existe pas une mesure de similarité parfaitement appropriée à tous les domaines, donc différentes mesures sont établies telles que [Ivana, 2006] :

- *Weighted block-city* : $sim(d^c, d^s) = \sum_{i=0}^n p_i sim_i(d_i^c, d_i^s)$
- la mesure euclidienne : $sim(d^c, d^s) = \sqrt{\frac{1}{n} \sum_{i=0}^n ((sim_i(d_i^c, d_i^s))^2}$
- la mesure de *Minkowski* : $sim(d^c, d^s) = \sqrt[r]{\frac{1}{n} \sum_{i=0}^n ((sim_i(d_i^c, d_i^s))^r}$

Où n est le nombre d'attributs, P_i est le poids de l'attribut i , qui est évalué en fonction de l'importance de l'attribut, sim_i est la similarité locale calculée dans le paragraphe précédent pour l'attribut i .

Pour la distance de *Minkowski* et si $r = 2$ alors on retrouve la distance *euclidienne*. Il existe d'autres mesures de similarité qui tiennent compte des historiques, des séquences, ou du coût de l'adaptation.

Algorithmes de recherche des cas similaires

Après le calcul de la mesure de similarité par le système RàPC, la remémoration ou la recherche des cas dans la base de cas se poursuit par un algorithme de recherche. Il existe plusieurs algorithmes qui permettent de rechercher des cas appropriés, par exemple :

- L'algorithme des K plus proches voisins (KPPV) : cette méthode utilise deux paramètres : la fonction de similarité pour comparer le cas cible aux cas sources stockés dans la base de cas, et le nombre K de cas sources les plus proches (similaires) du cas cible. C'est la méthode la plus utilisée [Fadi, 2009].
- L'algorithme basé sur les approches inductives : cet algorithme permet de déterminer les propriétés pertinentes dans la discrimination des cas. Il permet aussi de générer une structure arborescente des décisions pour organiser les cas dans la base des cas. On peut citer l'exemple de l'approche de KD-arbres basée sur les arbres de décision. Un autre exemple est celui des algorithmes d'apprentissage automatique [Haouchine, 2009].
- La recherche basée sur la structure (*template retrieval*) : c'est une méthode qui est similaire aux requêtes SQL et dans laquelle l'algorithme recherche des cas correspondant à certains paramètres.
- Les algorithmes appliqués à l'historique des séquences.
- L'algorithme de chemin de similarité.

2.4.3.3 Phase d'adaptation

L'adaptation consiste à proposer une solution au cas cible, dans le nouveau contexte propre au cas cible, par la réutilisation de la solution du cas source remémoré dans la phase précédente. Selon [Fuchs et al., 1999] l'adaptation est un plan dont l'état initial est la solution de départ et l'état final est la solution adaptée.

D'après [Lieber et al., 2004], la phase d'adaptation consiste à effectuer un raisonnement par analogie : *sachant que la solution du cas cible est à la solution du cas source ce que le cas cible est au cas source, connaissant le cas source et sa solution ainsi que le cas cible, que vaut la solution du cas cible ?*, ceci en s'appuyant sur la solution du cas source le plus similaire.

L'adaptation peut se faire soit via une intervention humaine (expert) soit d'une manière automatique à l'aide d'algorithmes, de règles, de formules, etc.

Il existe plusieurs types d'adaptation automatique dont les principaux sont les suivantes [Wilke et Bergmann, 1998] [Ivana, 2006] :

- **L'adaptation générative** : dans ce cas on suppose que nous avons toutes les connaissances pour résoudre le problème, en particulier on dispose de la trace du raisonnement ayant mené à la solution du cas source remémoré. Dans ce type d'adaptation on rejoue le raisonnement du cas retrouvé dans le nouveau contexte (cas cible).
- **L'adaptation transformationnelle** : dans ce type d'adaptation des éléments de la solution du cas remémoré sont modifiés, supprimés, ajoutés, selon des écarts de contexte observés entre le cas source et le cas cible. Cette transformation se fait grâce à un ensemble de règles d'adaptation.
 - **L'adaptation par substitution** comprend des changements des valeurs d'attributs.
 - **L'adaptation structurelle** permet de changer la structure de la solution et permet aussi d'organiser des éléments de la solution par l'ajout ou la suppression de certaines caractéristiques.
- **L'adaptation compositionnelle** utilise deux ou plusieurs cas similaires retrouvés dans la phase de remémoration afin d'effectuer l'adaptation en composant les différentes solutions proposées.
- **L'adaptation hiérarchique** où les cas sont structurés à plusieurs niveaux d'abstraction.

Cette phase est considérée comme la phase la plus complexe et la plus délicate à mettre en œuvre.

2.4.3.4 Phase de validation

L'étape de validation consiste à évaluer la solution proposée à l'issue de la phase d'adaptation. L'évaluation peut se faire via plusieurs actions [Mille, 1999] :

- tester la solution proposée par simulateur, système expert classique, etc. ;
- évaluer la solution proposée dans le monde réel ;
- analyser la base de cas en utilisant l'ensemble des caractéristiques des cas (les deux parties problème et solution) afin de vérifier que les cas similaires retrouvés ont donné entière satisfaction.

Après l'évaluation et si la solution est jugée insatisfaisante alors elle va être corrigée. Ces corrections peuvent être réalisées par :

- **L'utilisateur**, qui peut donner sa propre évaluation et ses corrections par rapport à la solution proposée par le système RàPC. On peut citer l'exemple des travaux de [Karoui et al., 2006] ;
- **L'expert** qui représente l'expertise du domaine étudié. On peut citer l'exemple des travaux de [Cordier et al., 2007] ;
- **La correction automatique** qui se base sur le principe d'auto-évaluation en utilisant la base de cas. Par exemple les travaux de [Malek et Kanawati, 2004].

La phase de révision permet de continuer l'élaboration de la solution cible si besoin. Après la validation du cas, les explications de révision deviennent une source importante d'apprentissage pour faire évoluer les connaissances du système RàPC. Toutefois, il n'existe pas encore de méthode standard pour la phase de révision dans le RàPC.

2.4.3.5 Phase d'apprentissage

La phase d'apprentissage ou de mémorisation permet d'incorporer ce qui est utile à retenir dans la base de cas. Elle permet de synthétiser les nouvelles connaissances dans la base de cas. Cette phase permet aussi d'optimiser la base de cas afin que les nouveaux cas soient réutilisés ultérieurement. La phase de mémorisation permet [Mille, 2006]:

- d'augmenter l'expérience du système et d'enrichir la base de connaissances ;
- de modifier les connaissances du système en cas de révision.

Il peut être utile de sauvegarder la trace du cycle en entier avec le détail des corrections faites. Cette trace peut être utilisée comme un modèle de correction du nouveau cas en phase de révision [Haouchine, 2009].

L'organisation de la base de cas est liée au nombre de cas qu'il contient et de leur structuration à l'intérieur de celle-ci. Par conséquent, la base de cas évolue dans le temps et la taille de la base augmente au fur et à mesure de l'ajout de nouveaux cas. Ceci peut perturber le fonctionnement et l'organisation de la base et diminuer efficacité et performance. Pour y remédier, nous parlerons de la maintenance de la base de cas [Ivana, 2006].

La maintenance de la base de cas est le processus d'affinement de la base de cas d'un système de RàPC. Elle permet de mettre en place des stratégies pour réviser la structuration et / ou le contenu de la base de cas pour faciliter le raisonnement futur [Leake et Wilson, 1998]. On distingue deux critères d'évaluation de la maintenance de base de cas :

1. La performance du système RàPC calculée par le temps nécessaire pour proposer une solution à un nouveau problème ;
2. L'efficacité du système mesurée par le nombre de problèmes différents résolus correctement par le système.

Il existe plusieurs approches pour la maintenance de la base de cas, toutefois la plupart de ces méthodes s'orientent vers l'optimisation du contenu de la base de cas. Ces méthodes permettent de réduire la taille de la base donc d'augmenter la performance du système. Cette diminution peut être réalisée par des suppressions de cas en catégorisant les cas dans la base de cas selon leurs compétences [Haouchine et al., 2006] [Leake et Wilson, 2000] ainsi que par des ajouts de cas [Smyth et McKenna, 1999], [Yang et Zhu, 2001] en créant une nouvelle base de cas réduite. Il existe d'autres stratégies de maintenance qui consistent à partitionner la base de cas sous forme d'un ensemble de petites bases où la recherche de cas est moins coûteuse [Yang et Wu, 2000]. Enfin il existe des méthodes qui s'appuient uniquement sur la stratégie des mesures de performance et qui sont généralement basées sur le coût d'adaptation [Leake et Wilson, 2000]. Dans la section suivante, nous présentons en détail les structures de cas et base de cas.

2.4.4 Cas / Base de cas

2.4.4.1 Définition du cas

Un cas décrit un épisode de résolution de problème, c'est une expérience représentée par une connaissance. Un cas peut être vu comme étant la description informatique d'un épisode de résolution de problème [Fuchs et al., 2006].

2.4.4.2 Structure d'un cas

Un cas en RàPC est généralement composé de deux parties : la partie problème dans laquelle on trouve la description du problème à résoudre ou les objectifs à atteindre et la partie solution, qui contient la description de la solution apportée par le raisonnement. Dans certain cas la partie solution contient en plus de la solution sa justification, son évaluation ainsi que les étapes qui ont mené à cette solution.

On distingue deux types de cas : *cas cible* et *cas source*. Le cas cible est un couple (problème, solution) dont lequel la partie solution n'est pas renseignée. Le cas source est un couple composé d'une partie problème et une partie solution. Les deux parties qui composent le cas source sont bien renseignées. Donc, c'est un cas dont on va s'inspirer pour résoudre un nouveau problème (cas cible). On signale que le cas source peut contenir une autre partie. Cette partie contient la justification, l'évaluation ou encore les étapes du raisonnement qui ont amené à la solution.

Il existe de nombreux travaux sur la représentation des cas. Cette représentation dépend de la nature du problème à résoudre. Les approches traditionnelles généralement utilisent la représentation de cas structurée en liste de descripteurs (qui peuvent être des objets complexes). Un cas, son problème et sa solution sont donc décrits par un ensemble de descripteurs [Haouchine, 2009]. Un descripteur d est défini par une paire $d = (a : \text{attributs}, v : \text{valeurs})$ où v est la valeur qui est associée à l'attribut a dans ce cas. Nous empruntons le formalisme de [Lieber, 2007] pour définir

les cas. Un cas cible est le couple $(cible, Sol(cible))$, où $Sol(cible)$ est inconnue et pour laquelle on cherche une solution et un cas source qui est représenté par un couple $(source, Sol(source))$. Suivant ce vocabulaire, les cas cible et source peuvent être définis de la manière suivante :

- Source = $\{d_1^s \dots d_n^s\}$ où d_i^s est un descripteur du problème source.
- $Sol(source) = \{D_1^s \dots D_m^s\}$ où D_i^s est un descripteur de la solution source.
- Cible = $\{d_1^c \dots d_n^c\}$ où d_i^c est un descripteur du problème cible.
- $Sol(cible) = \{D_1^c \dots D_m^c\}$ où D_i^c est un descripteur du problème cible à retourner par le cycle RàPC.

Exemple de cas :

Exemple d'un cas : condition de vente d'un appartement.

Partie problème d_1^s = Surface de l'appartement (un réel) d_2^s = Le nombre des pièces (un réel) d_3^s = Le nombre de balcons (un réel) d_4^s = L'existence d'un parking (symbole)
Partie solution D_1^s = Prix de vente de l'appartement (un réel)

Figure 2.4 : Exemple de structure d'un cas

Dans cet exemple, la structure du cas contient quatre descripteurs en partie problème et un descripteur en partie solution.

Cas cible Partie problème Surface = 60 Le nombre des pièces = 2 Le nombre de balcons = 1 L'existence d'un parking = 'Non'
Partie solution Prix de vente = ?

Figure 2.5 : Exemple d'un cas cible

Cas source i Partie problème Surface = 65 Le nombre des pièces = 2 Le nombre de balcons = 1 L'existence d'un parking = 'Oui'
Partie solution Prix de vente = 65000 euros/DHs

Figure 2.6 : Exemple d'un cas source

La représentation utilisée dans cet exemple est une représentation de type vecteur, sous forme d'une liste de couple (attribut, valeur). Les attributs peuvent être de différents types : numérique, symbolique, ou booléen.

2.4.4.3 Indexation du cas

Les cas adoptés par le système RàPC sont organisés dans une mémoire appelée base de cas. Pour une organisation performante ainsi qu'une recherche rapide et efficace du cas le plus approprié au problème posé, il faut les indexer. Lors de la recherche de ces cas sources similaires au cas cible à résoudre, c'est la partie problème qui va être sollicitée. Cette partie problème peut être décrite par un ensemble de caractéristiques pertinentes appelées indices [Ivana, 2006]. Ces indices aident à déterminer le contexte et la situation dans laquelle les cas vont être recherchés et retrouvés facilement. Les cas mémorisés sont proposés au problème rencontré.

Il existe plusieurs méthodes d'indexation : manuelles ou automatiques. Pour le cas des méthodes manuelles, il faut déterminer particulièrement l'objectif d'utilisation des cas et les circonstances dans lesquelles les cas seront utiles [Haouchine, 2009]. Néanmoins, les méthodes d'indexation sont de plus en plus automatisées. Nous distinguons trois principales méthodes :

- **Indexation basée sur la similarité** : elle crée les indices d'un cas en se basant sur les propriétés communes des cas semblables.
- **Indexation basée sur l'explication** : elle précise les caractéristiques adéquates pour chaque cas. Cette méthode procède par une analyse des cas pour trouver les attributs prédictifs qui peuvent être utilisés comme des index.
- **Méthodes basées sur l'apprentissage inductif**, dans ce cas les attributs pouvant jouer le rôle d'index sont déterminés par les algorithmes inductifs basés sur les arbres de décision.

Après l'ajout d'un nouveau cas à la mémoire, le problème qui se pose est de savoir quels indices générer et manipuler pour ce cas. Selon [Kolodner, 1996], le choix des indices dépend du domaine d'application. Kolodner ajoute que ces indices doivent vérifier certaines propriétés :

- prédictifs afin de jouer un rôle déterminant au niveau du choix de la solution pour un nouveau problème ;
- suffisamment abstraits de telles façons qu'un cas puisse être utilisé plusieurs fois pour la résolution de plusieurs problèmes ;
- suffisamment concrets afin que le cas soit reconnu rapidement pour la résolution d'un nouveau problème.

2.4.4.4 Base de cas dans les systèmes de RàPC

Les cas sources sont stockés dans une base appelée base de cas (la mémoire). Le bon fonctionnement d'un système RàPC est fortement lié à l'organisation de sa mémoire. C'est un élément important dans l'indexation et l'organisation des cas. Cette organisation dépend de critères bien déterminés permettant de retrouver facilement et efficacement les cas sources [Fuchs, 1997].

En fonction de la représentation du cas, on distingue deux types d'organisations principales de la base de cas :

- base de cas plate dans laquelle les cas sont organisés de façon linéaire (vecteur, tableau, graphe...). Les cas sont stockés dans une liste séquentielle. Cette organisation est sans doute la plus simple et la plus utilisée.
- base de cas hiérarchique où les cas sont structurés suivant une organisation hiérarchique.

Il existe également des bases de cas hybrides, qui sont construites en utilisant les deux types d'organisations combinées. Par exemple, nous pouvons trouver ce type de combinaison dans les travaux de Malek [Malek, 2000].

Lors de la structuration d'une base de cas, les trois points suivants doivent être considérés [Main et al., 2000] :

1. La représentation et la structure des cas ;
2. Le modèle de la mémoire utilisée pour organiser les cas ;
3. Le choix des indices qui sont utilisés pour identifier chaque cas.

En effet le type d'organisation de la mémoire, qui représente le cœur du système RàPC, influencera les étapes du RàPC et plus particulièrement les étapes de recherche et d'apprentissage qui exploitent directement la mémoire.

La base de cas joue un rôle primordial dans le fonctionnement du système RàPC et constitue leur principale source de connaissances. En effet le système RàPC a pour but de capitaliser l'expertise sous forme de connaissances dans sa base de cas. Il a pour but aussi de réutiliser des connaissances similaires pour prendre une décision et d'enrichir la mémoire en ajoutant de nouvelles connaissances (les cas sources). Dans la section suivante, nous présentons en détails les connaissances dans le système RàPC.

2.4.5 Connaissances

Les différentes connaissances utilisées dans un système de raisonnement à partir de cas sont regroupées en quatre catégories ou quatre containers de connaissances distincts à savoir [Richter, 1998] :

1. **Le container de vocabulaire** : c'est le vocabulaire du système RàPC, il présente toutes les informations sur les définitions et les structures utilisées. Ce container est l'ensemble des éléments de représentation subatomiques utilisés pour représenter les cas dans la base de cas [Lieber et al., 2004] ;
2. **Mesures de similarité** : contient les fonctions et formules nécessaires pour évaluer la similarité entre le cas cible et le ou les cas sources. Ces fonctions sont définies en fonction du domaine d'application du système RàPC. Elles sont utilisées pour déterminer, lors de la phase de remémoration, les cas sources les plus appropriés au nouveau cas cible ;
3. **Le container d'adaptation** : ce container contient les règles de transformation de la solution et les algorithmes décisionnels. Ces derniers permettent de

transformer les solutions des cas sources remémorés pour s'adapter au mieux au cas cible ;

4. **Le container de la base de cas** : représente le contenu et l'organisation de l'ensemble des expériences structurées (cas). C'est le container dans lequel sont stockées toutes les expériences, qui seront exploitées par le processus de l'approche RàPC.

Les trois premiers containers de connaissances du système RàPC sont élaborés avant l'exécution du système, par contre le container de base de cas est remis à jour de manière dynamique. Les containers de connaissances sont fortement liés entre eux [Wilke et al., 1997] et [Wilson, 2001]. En effet un système RàPC qui dispose d'un système de connaissance complète et contenant assez de cas nécessite très peu d'adaptation. Toutefois, lorsqu'un système dispose d'une adaptation puissante, alors juste très peu de cas sont suffisants pour le bon fonctionnement du système RàPC [Haouchine, 2009].

2.4.6 Domaines d'application et conditions d'utilisation du RàPC

Le raisonnement à partir de cas est utilisé dans de multiples domaines d'application. Parmi ces domaines, nous pouvons citer l'enseignement, la médecine, l'industrie, le commerce, la justice, la maintenance, l'analyse financière etc. Le RàPC possède également plusieurs types d'applications dans les différents domaines. Ces applications permettant d'accomplir des tâches de différentes natures comme la planification, la classification, le diagnostic, la gestion des connaissances, la conception, la navigation des robots, la configuration, la sécurité routière et l'aide à la décision qui représente justement notre centre d'intérêt. La littérature des applications RàPC proposent différentes typologies d'applications. Elles sont dépendantes du domaine étudié et de la nature de la tâche à accomplir.

2.4.6.1 Typologies d'applications

Kolodner [Kolodner, 1993] sépare les applications RàPC en deux types :

1. **Les systèmes de résolution de problèmes**, dans ce type d'application la solution proposée au nouveau problème est une dérivée des solutions déjà connues dans la base des cas. Ils regroupent des tâches de diagnostic, de conception et de planification ;
2. **Les systèmes interprétatifs** dans ce type d'application les situations déjà connues et stockées dans la mémoire servent de contexte de base pour la nouvelle situation à résoudre. Ils regroupent des tâches d'interprétation, de justification, d'évaluation et de classification.

Certains auteurs distinguent les applications RàPC, suivant le type d'utilisation, en deux catégories à savoir les applications académiques et commerciales. Toutefois [Althoff et al., 1995] les décomposent suivant le type de la tâche à accomplir :

- **les tâches de classification** : ce type contient les systèmes d'évaluation, de diagnostic de prédiction et de helpdesk ;
- **les tâches de synthèse/planification** : dans ce type on trouve les systèmes de conception, de planification et de configuration.

Althoff et Bartsch-Spörl [Althoff et Bartsch-Spörl, 1996] classent les domaines d'application du RàPC selon les deux tâches suivantes :

1. **Analytiques** : ce sont les applications qui sont sous forme de situations, qui doivent être classifiées afin de retrouver le cas correspondant et adapter ensuite leur solution. On trouve ce type, par exemple dans le diagnostic médical, la classification d'objet et la sélection de produits dans le e-commerce ;
2. **Synthétiques** : ce type d'application demandant une planification et une construction de la solution. Dans ce type il faut proposer une solution aux nouveaux problèmes et décrire les conditions d'applications de la dite solution. On peut citer par exemple le transport, les systèmes d'aide à la décision, l'enseignement et la gestion des ressources.

Althoff [Althoff, 2001] a introduit la notion de la hiérarchie des applications en fonction de la complexité de la résolution de problèmes. Il propose les quatre niveaux hiérarchiques suivants (figure 2.7) :

1. **La classification** se trouve au début de la hiérarchie dans laquelle la solution du nouveau problème est en relation avec la sélection d'une ou plusieurs classes ;
2. **Le diagnostic** vient au deuxième niveau dans la hiérarchie et est considéré comme une extension de la classification. Cette généralisation de la classification touche la connaissance générale du domaine qui va rechercher les caractéristiques nécessaires dans un processus de diagnostic ;
3. **L'aide à la décision** se classe au troisième niveau hiérarchique et se caractérise par la distinction des symptômes des solutions de problèmes, qui sont souvent indirectes et invisibles ;
4. **La gestion des connaissances** est le type d'application le plus général et le plus complexe aussi. Dans ce type aucune méthode de raisonnement ne peut être utilisée directement.

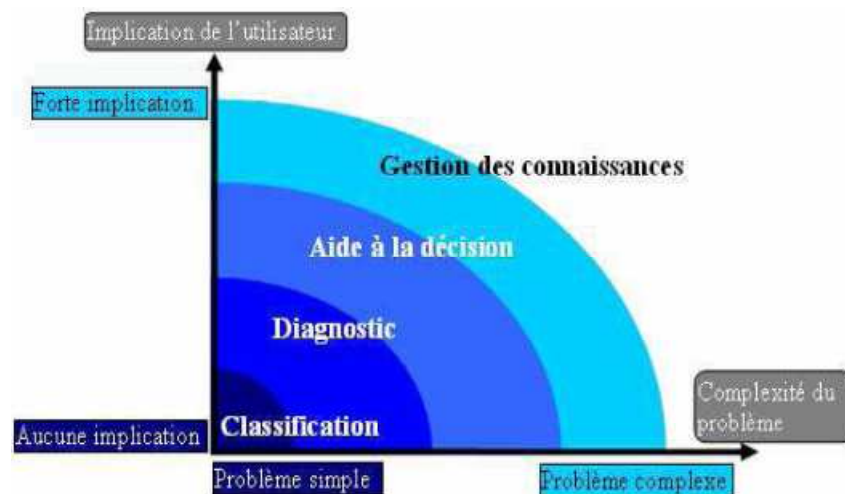


Figure 2.7 : Niveaux hiérarchiques des applications des systèmes de Règle à Partir de Cas selon [Althoff, 2001]

Finalement Richter, cité par [Haouchine, 2009], propose un classement plus conventionnel en se basant sur la catégorisation proposée par [Althoff et Bartsch-Spörl, 1996]. Selon Richter les deux types de systèmes de Règle à Partir de Cas sont :

1. **Résolution de problèmes**, contient les systèmes qui utilisent des tâches analytiques;
2. **Prise de décision** qui s'intéresse aux tâches synthétiques.

Selon les niveaux hiérarchiques présentés par Althoff [Althoff, 2001], nous nous intéressons dans nos travaux de recherches au troisième niveau à savoir l'aide à la décision. Ce niveau concerne les systèmes orientés aide à la décision. Dans notre cas, la complexité du problème à traiter est plus importante que les problèmes de diagnostic et de classification et le degré d'implication de l'utilisateur est également plus important. Ce niveau concerne les systèmes dont la formalisation du cas est plus complexe que les autres cas. De plus, une modélisation des connaissances du système est nécessaire.

Après cet état de l'art des typologies d'application en Règle à Partir de Cas, nous notons que le Règle à Partir de Cas ne s'applique que sous certaines conditions qui sont décrites au paragraphe suivant.

2.4.6.2 Conditions d'applications

L'approche Règle à Partir de Cas présente plusieurs atouts. En effet le Règle à Partir de Cas permet de réutiliser les solutions existantes. Le Règle à Partir de Cas se caractérise par sa capacité d'adaptation face aux changements de l'environnement. Il faut aussi définir de bonnes mesures de similarité et de trouver des stratégies d'adaptation efficaces avec un faible coût [Ivana, 2006]. De plus, la conception et l'application du Règle à Partir de Cas est conditionnée par un ensemble d'hypothèses portant sur le domaine étudié ou le type d'application choisi [Kolodner, 1996] :

- **similarité** : l'utilisation des cas stockés dans la mémoire pour en résoudre de nouveaux problèmes qui lui sont similaires est basée sur l'hypothèse que des problèmes similaires peuvent avoir des solutions semblables [Schank, 1982] ;
- **facilité d'adaptation** : les descripteurs des problèmes des cas cible et source sont peu différents et ces faibles différences peuvent être évaluées ;
- **typicalité** : les problèmes se répètent fréquemment et les cas retenus par le système (les expériences) peuvent être indispensables dans le futur ;
- **régularité** : l'application des mêmes actions (dans le cadre d'adaptation des solutions proposées) dans les mêmes conditions (contexte d'une situation) ont des conséquences semblables voire identiques ;
- **cohérence** : des petits changements qui peuvent arriver dans l'univers demandent seulement des petits changements dans l'interprétation et par conséquent des faibles modifications dans les solutions.

Main et al. [Main et al., 2000] considèrent que l'approche RàPC ne peut être appliquée que si les conditions suivantes sont réunies :

- **la reproduction des situations**, en absence d'une similarité suffisante entre les problèmes à résoudre et les cas résolus stockés dans la mémoire, il faut utiliser une autre solution que l'approche RàPC ;
- **l'existence des modèles sous-jacents dans le domaine**, si la description du cas ne peut tenir compte des facteurs qui ont conduit au succès ou à l'échec d'une solution, ou si le processus de résolution des problèmes est aléatoire alors la méthode RàPC ne serait pas adaptée à ce type d'application ;
- **les nouveaux cas et les exceptions**, si dans le domaine étudié ne se produit pas de nouvelles situations ou d'exceptions alors il vaut mieux mettre en place une modélisation gérée par des règles ;
- **l'apport de l'adaptation des solutions des cas stockés dans la base de cas**, si la création d'une solution n'est pas significative par rapport à l'obtention de la même solution par adaptation alors l'approche RàPC n'est pas le bon choix pour ce type de système. Cet apport est évalué suivant un ensemble de critères tels que le traitement de l'information et le temps nécessaire pour résoudre un problème.

Dans le cas où ces conditions sont réunies, alors la mise en place d'un système fondé sur le RàPC est possible et l'étude du domaine d'application peut débuter. Cette étude dépend essentiellement des caractéristiques de l'application, de leur nature et des buts à atteindre. Les systèmes à base de RàPC diffèrent entre eux par : la structure de cas, la structure de base de cas, l'indexation, la façon d'utiliser les modèles de connaissance, les étapes du cycle et les différentes règles, les algorithmes et méthodes utilisées dans les différentes étapes. Nous pensons que les conditions d'applications cités ci-dessus sont satisfaites pour notre système par la réunion de la typicalité, la similarité, la facilité d'adaptation, la régularité et la cohérence, ceci afin de développer notre RàPC orienté aide à la décision pour prédire une situation dynamique basée sur

les expériences passées. Toutefois, nous allons commencer d'abord par la description des systèmes de RàPC pour des situations statiques.

2.5 Raisonnement à partir de cas pour situations statiques

Le RàPC est utilisé dans de multiples domaines, et nous distinguons deux catégories d'applications : les systèmes RàPC pour situations statiques et ceux pour situations dynamiques. Le RàPC statique est le premier modèle qui a été utilisé lors de l'apparition des systèmes RàPC. Il se caractérise par certaines particularités par rapport aux autres systèmes telles que :

- le concepteur de l'approche RàPC statique doit disposer à l'avance de toutes les caractéristiques décrivant un cas cible pour pouvoir accomplir son modèle.
- les cas sont complètement structurés et souvent représentés par une liste de descripteurs, sous forme de paires $d = (a : \text{attributs}, v : \text{valeurs})$.
- l'indexation pour organiser la base de cas.

Plusieurs travaux concernant le RàPC modèle statique ont été réalisés dans différents domaines. Comme exemple représentatif, nous avons choisi de détailler CHEF [Hammond, 1986] qui est un système de planification dans le domaine de la cuisine. Parmi ces travaux nous pouvons citer aussi PATDEX [Richter et Wess, 1991] qui est dédié au diagnostic des machines complexes, on a aussi le système PAD'IMP [Mille, 1995] de supervision dans le milieu industriel. Dans le domaine de transport on peut citer CREEK [Aamodt, 2004], CASSIOPEE [Bergmann et al., 2003] qui s'intéresse au dépannage de moteurs d'avion. Parmi les travaux dans le domaine médical nous pouvons citer les systèmes : CASEY [Koton, 1988] dédié au diagnostic cardiaque, PROTOS [Bareiss, 1988] pour les problèmes d'audition, ou ISIS [Kahn, 1993] dédié à la radiologie. On a aussi le système RADIX [Corvaisier et al., 1998] qui propose des modélisations de sessions de navigation sur internet effectuées par l'utilisateur pour lui suggérer des sessions similaires. Enfin nous citons un système qui est lié au domaine juridique CABARET [RISSLAND et al., 1995], ce dernier utilise une approche RàPC pour faciliter la recherche dans les bases documentaires juridiques.

2.5.1 La similarité entre les cas en situations statiques

Les mesures de similarité présentées au début de chapitre, peuvent être locales ou globales. La similarité locale entre deux cas est mesurée en fonction de la distance entre les valeurs de mêmes attributs. Cette distance est généralement estimée par des mesures euclidiennes. La similarité globale entre deux cas est habituellement évaluée par une somme pondérée de la similarité locale au niveau de chaque attribut. Or les attributs d'un cas n'ont pas tous la même importance et que cette importance varie d'un cas à l'autre, donc un poids est associé à chaque attribut de chaque cas. Ces poids permettent de pondérer la similarité globale entre deux cas en accordant une grande importance aux attributs les plus " méritants ".

2.5.2 L'adaptation en situations statiques

L'adaptation dans le contexte statique peut varier d'une simple substitution de la valeur d'un attribut ou d'une simple transformation jusqu'à la dérivation complète ou la restructuration complète d'une solution.

Dans la partie suivante, nous présentons un exemple très connu de système RàPC statique.

2.5.3 Système CHEF

CHEF est un système de planification de menus [Hammond, 1986]. CHEF a pour objectif de proposer des recettes qui satisfont un ensemble de contraintes exigées par l'utilisateur. Elles sont liées aux goûts, textures, ingrédients et types de plats choisis par le demandeur. Par exemple on peut avoir un problème cible de la forme suivante (requête exprimé en langue naturelle) : " Je veux une recette de bœuf aux brocolis ". Dans ce cas la solution Sol(cible) proposée à ce problème est sous forme d'un plan décrivant les étapes de préparation d'une recette satisfaisant les contraintes exigées par l'utilisateur [Fadi, 2009]. Le système CHEF sélectionne parmi les plans présents dans la base de cas un plan qui satisfait au mieux les contraintes énoncées dans le cas cible afin de l'adapter au nouveau contexte [Hammond, 1989].

L'étape de **remémoration** de CHEF consiste à sélectionner un plan de la base de cas qui satisfait les buts énoncés dans le cas cible. Le système CHEF procède au calcul de similarité en se basant sur l'importance hiérarchique des descripteurs. Exemple : le type du plat est le plus important, ensuite le type de viande, ensuite type de légumes.

Le processus **d'adaptation** du système CHEF commence d'abord par une adaptation substitutionnelle pour satisfaire au mieux les contraintes énoncées dans le cas cible puis la correction des éventuels problèmes rencontrés dans la solution proposée. Cette dernière étape n'est appliquée que si la première adaptation de la solution n'est pas satisfaisante. L'adaptation comporte ainsi deux étapes [Hammond, 1986] :

1. Au début le système commence par une adaptation qui consiste à substituer les ingrédients dans le cas retrouvé comportant une recette pour satisfaire au mieux les contraintes énoncées. Par exemple pour obtenir une recette de bœuf aux brocolis à partir d'une recette de bœuf aux pois. L'application de cette adaptation consiste à ajouter des brocolis à la recette et à enlever les pois.
2. Un modèle causal est utilisé à la fois pour détecter et pour dégager les problèmes susceptibles d'être rencontrés dans la solution proposée [Hammond, 1986]. Dans cette seconde étape il s'agit d'ajouter, de modifier et de supprimer des étapes dans la recette. Dans l'exemple précédent, une étape dans le plan de la recette consiste à frire tous les ingrédients. Or le système a détecté un problème dans la recette, en fait les brocolis ne sont pas frits convenablement (ils ont été ajoutés entiers). Afin de résoudre ce problème, le système CHEF se

base sur une critique qui stipule que tout légume doit être coupé en morceaux avant d'être frit. Par conséquent CHEF rajoute une étape de préparation dans laquelle les brocolis sont coupés en petits morceaux.

Pour les connaissances d'adaptation, le système contient trois types de connaissances qui entrent en jeu dans l'adaptation :

1. Un ensemble d'opérateurs de substitution prédéfinis ;
2. Un modèle causal, qui est exploité pour prévenir les problèmes susceptibles d'être rencontrés dans la solution proposée ;
3. Un ensemble de critiques, qui se composent des stratégies de réparation à appliquer afin de résoudre ces problèmes.

Pour l'**étape de révision**, CHEF utilise un arbre causal pour détecter et expliquer les problèmes qui peuvent être rencontrés. Ensuite le système procède à la réparation de ces problèmes en se basant sur un ensemble de critiques [Hammond, 1986].

Enfin la **phase d'apprentissage** permet d'ajouter les plans jugés pertinent dans la base de cas. Ces cas sont indexés dans la base de cas suivant les buts à atteindre et les problèmes à éviter.

Dans cette présentation du système CHEF le processus d'adaptation a été spécifiquement détaillé car nous pensons qu'il est le point fort de ce système.

Dans l'approche raisonnement à partir de cas statique un problème doit être complètement décrit avant que ne débute la recherche dans la base de cas. Cette exigence présuppose une expertise du domaine d'application permettant de bien préciser une situation à l'aide de valeurs numériques ou symboliques et de choisir les facteurs indispensables pouvant influencer la résolution de son problème. Toutefois pour plusieurs domaines d'applications particulièrement les problèmes de prédiction, ces aspects sont difficiles à déterminer à l'avance. Le modèle dynamique a donc été proposé pour surmonter ces difficultés. Il est actuellement le modèle le plus répandu. Dans la partie suivante, nous présentons en détail le RàPC pour situation dynamiques.

2.6 Raisonnement à partir de cas pour situations dynamiques

Les applications du raisonnement à partir de cas traitant des situations dynamiques ont vu le jour au début des années 90 et se caractérisent par le fait de traiter des cas cibles de nature temporelle (la situation), et de réaliser une recherche du meilleur cas basée sur une similarité entre historiques (également appelés chroniques). Ces systèmes ont été appliqués dans plusieurs domaines tels que : le système REBECAS (pREdiction BaséE sur des CAS) [Rougegrez-Loriette, 1994] pour la prédiction de la propagation d'incendies de forêts, les systèmes AuRA (*Autonomous of Robot Architecture*) [Likhachev et al., 2002] et SAPED (Système d'Assistance pour Personnes HanDicapées) [Aich, 2007] pour l'assistance des utilisateurs d'un fauteuil roulant, le système CASEP2 (*CAs e-based reasoning System for SEquence Prediction*)

[Zehraoui, 2004] et le système BROADWAY [Jaczynski, 1998] pour la prédiction du comportement d'un internaute par la recommandation de pages web à visiter.

Ces systèmes sont issus de domaines d'application très différents mais ils partagent des contraintes communes :

- ils traitent de situations dynamiques ;
- les observations du monde réel sont représentées par des historiques d'évolution des valeurs de différents paramètres qui représentent le domaine ;
- ils font évoluer leur base de cas au fil du temps en se basant sur l'apprentissage des nouveaux cas.

Nous présentons dans cette partie les différentes méthodes proposées pour développer ces systèmes, en particulier la représentation des cas, l'organisation de la mémoire et le cycle de raisonnement RàPC pour situations dynamiques. Nous proposons ensuite une étude de types de représentation des cas dynamiques possibles, et les avantages et inconvénients de ces différents types. Nous terminons par une étude comparative détaillée des différents systèmes de raisonnement à partir de cas dynamiques RàPCD.

2.6.1 Représentation dynamique des cas

Les systèmes de raisonnement à partir de cas dynamiques se caractérisent par le fait de traiter des cas cibles de nature temporelle (la situation). Une situation dynamique est décrite par un ou plusieurs historiques (ou encore une ou plusieurs séquences), qui représentent l'évolution de un ou plusieurs paramètres, pertinents pour la prédiction de la situation.

2.6.1.1 Représentation d'un historique

Un **premier classement** consiste à distinguer les systèmes où l'historique est représenté par des états ou bien par des événements :

- La représentation par états d'une situation décrit les valeurs des différents paramètres, qui caractérisent le domaine à un instant donné. On trouve cette représentation dans [Bull et al., 1997], qui permet de prédire le risque sanitaire d'une population vers une épidémie. Les différents paramètres recueillis pour décrire la situation sanitaire de la population pendant une semaine sont agrégés dans des états. On trouve aussi cette représentation dans le système ICONS [Schmidt et al., 1998], où les paramètres recueillis chaque jour sont associés à un état correspondant à l'évolution du dysfonctionnement rénal décrit par ces différents paramètres.
- La représentation par des événements permet de décrire le changement significatif des différents paramètres à un instant donné. Cette représentation est donc constituée de plusieurs historiques, appelés aussi trajectoires [Brandau, 1991]. Chaque trajectoire est associée à un paramètre pertinent dans la

description de la situation. Plusieurs travaux ont utilisé cette représentation, comme le système SINS [RAM et Santamaria, 1997], qui permet d'adapter en temps réel des paramètres de contrôle d'un robot. Ces paramètres décrivent d'une part les caractéristiques propres à son déplacement comme la vitesse, la direction, etc. et d'autre part les caractéristiques de l'environnement traversé, comme le nombre d'obstacles par exemple. La représentation par événements a été adoptée également par le système REBECAS, qui est un système de prédiction de la propagation d'un feu de forêt [Rougegrez-Loriette, 1998]. Ce système tente de déterminer à quel endroit se situera le front de flammes à un instant donné. Dans REBECAS, un cas représente de manière continue l'évolution des valeurs des différentes variables d'observations : le relief, la végétation, le vent et la propagation du feu. Ceci depuis le départ d'un feu jusqu'à sa maîtrise. Un événement du relief décrit une hauteur et une forme caractéristique. Un événement végétation décrit un type de végétation (broussailles, vignes, etc.). Un événement vent représente le changement de vitesse du vent.

Un **deuxième classement** consiste à distinguer les systèmes où l'historique est représenté par une succession d'instants ou d'événements instantanés ou encore une représentation d'historiques en temps discret (approche à base de points ou d'instants) et les systèmes qui représentent le cas par des relations entre des intervalles temporels (représentation d'historiques en temps continu). La représentation par des historiques en temps discret a été utilisée dans plusieurs domaines comme celui de la prédiction des feux de forêts REBECAS [Rougegrez-Loriette, 1998] et la recherche d'information BROADWAY [Jaczynski et Trousse, 1998]. La représentation de cas par des relations entre des intervalles est une nouvelle représentation qui est plus riche par rapport à la première. Un cas est représenté par des relations entre des intervalles temporels en se basant sur la logique temporelle d'Allen. Cette représentation a été proposée dans [Jaere et al., 2002]. La représentation de cas par une succession d'instants (temps discret) permet de représenter l'information temporelle de manière simple. Cette représentation est certainement la plus utilisée. La représentation par des relations entre les intervalles est une nouvelle méthode qui est très peu utilisée, vue sa complexité, mais elle représente les cas de façon plus précise et plus complète. Cette représentation est intéressante et peut être utile dans les domaines où la représentation par points ne donne pas de bons résultats.

Une **troisième classification** distingue les systèmes où le cas représente une séquence complète et les systèmes où le cas est une sous-séquence : un segment de la séquence de valeurs décrivant une situation traitée. Le principe de segmentation dépend du domaine d'application pour laquelle le système a été développé. Plusieurs travaux de recherches ont adopté la représentation de cas sous forme d'une séquence complète tels que le système REBECAS [Rougegrez-Loriette, 1998] et le système RADIX [Corvaisier et al., 1997] pour l'assistance de la navigation sur le web et le

système de recommandation dans le domaine industriel PADIM [Fuchs et al., 1995]. La représentation des cas sous forme de sous-séquences a été utilisée dans plusieurs domaines aussi tels que : les systèmes BROADWAY [Jaczynski, 1998], COBRA [Malek et Kanawati, 2001], CASEP [Zehraoui et al., 2003], et CASEP2 [Zehraoui, 2004], pour la prédiction du comportement d'un internaute ou SAPED [Aich, 2007] pour l'assistance des utilisateurs d'un fauteuil roulant. On a aussi l'exemple des systèmes AuRA [Likhachev et al., 2002] et SINS [Ram et Santamaria, 1997], pour l'ajustement des paramètres de déplacement d'un robot.

La représentation d'un cas sous forme d'une séquence entière est coûteuse en temps de calcul et occupe plus d'espace mémoire, mais elle permet au système utilisant ce type de représentation d'avoir des résultats plus précis et plus complets. Or l'inconvénient majeur de cette représentation est que pour certains systèmes l'utilisant, aucune information n'est mémorisée durant l'évolution de la situation. Par exemple pour mettre à jour les connaissances du système, l'ajout de la séquence complète à la base de cas est indispensable, comme dans le cas du système REBECAS. Quant à la représentation de cas par sous-séquences, elle permet de retrouver des similarités entre des parties de cas différents, ce qui permet à ces systèmes de prendre en considération les différents raisonnements effectués durant l'évolution de la situation, comme dans BROADWAY [Jaczynski, 1998] et CASEP [Zehraoui et al., 2003] et elle permet aussi à ces systèmes d'améliorer les résultats et d'éviter les échecs précédents.

Nous avons organisé les représentations de cas (cible / source) dynamiques dans le diagramme ci-après.

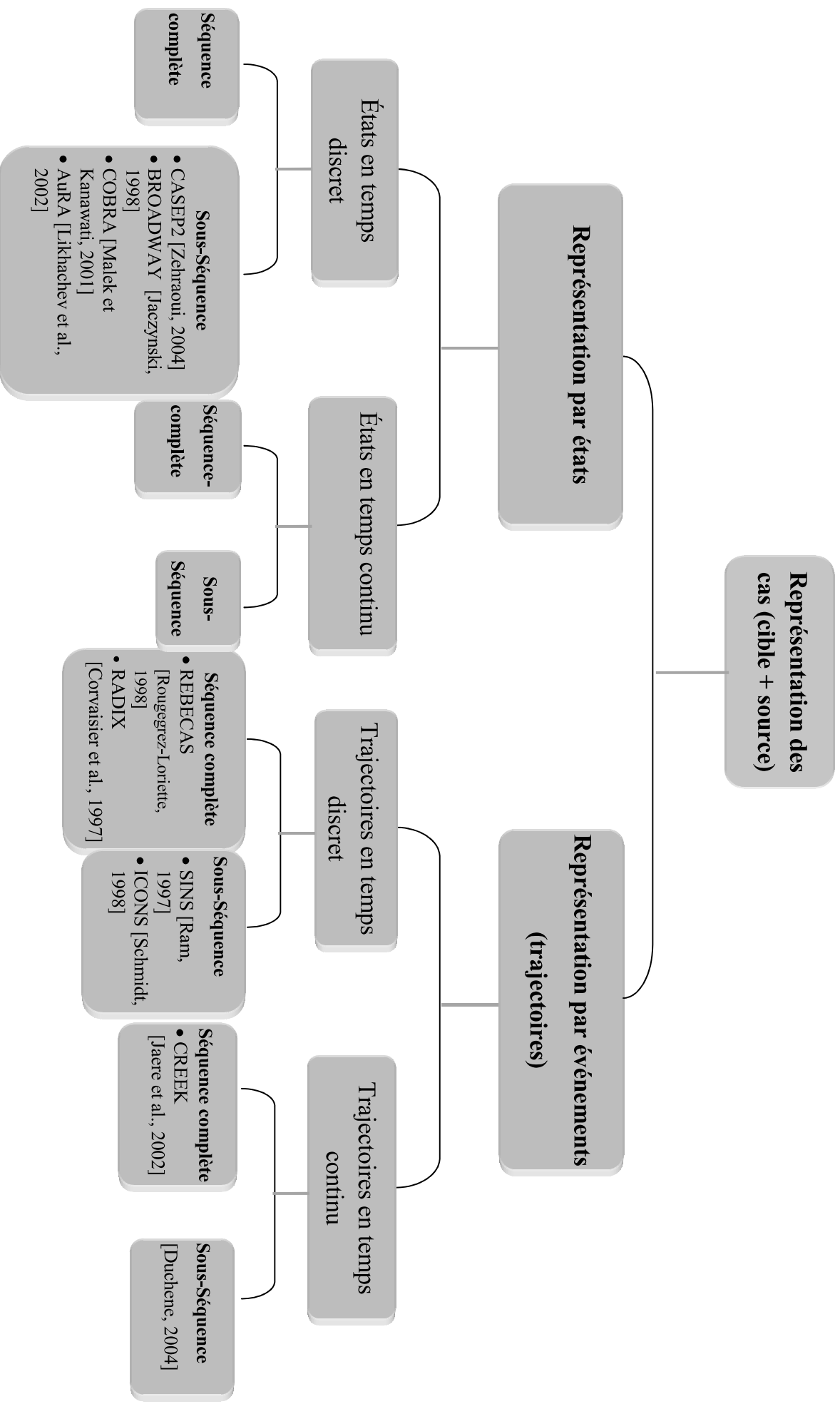


Diagramme des types de représentations des cas dynamiques

On remarque que la représentation la plus utilisée par les systèmes est celle de la représentation par des sous-séquences d'état en temps discret, puis la représentation des cas sous formes de trajectoires en séquence complète et sous-séquence. En analysant ces systèmes on trouve que la première est mieux adaptée pour les systèmes où les variables sont dépendantes alors que la seconde est efficace pour les variables indépendantes.

Avantages / inconvénients des types de représentation des cas dynamiques

Après notre analyse des différents types de représentation des cas dynamiques, nous proposons dans le tableau ci-après la liste des avantages et inconvénients de chacune de ces représentations.

Type de représentation			Avantages	Inconvénients
Représentation par états	états en temps discret	séquence complète	- représentation simple - résultats plus précis et plus complets - mieux adapté pour les systèmes ou les variables sont dépendantes	- coûteuse en temps de calcul - occupe plus d'espace mémoire
		sous-séquence	- représentation simple - pas coûteuse en temps de calcul - occupe moins d'espace mémoire - prend en considération les différents raisonnements effectués durant l'évolution de la situation - mieux adapté pour les systèmes ou les variables sont dépendantes	- pour certain domaines cette représentation ne donne pas de bons résultats
	états en temps continu	séquence complète	- résultats plus précis et plus complets - représente les cas de façon plus précise et plus complète - mieux adapté pour les systèmes ou les variables sont dépendantes	- représentation complexe - coûteuse en temps de calcul - occupe plus d'espace mémoire
		sous-séquence	- prend en considération les différents raisonnements effectués durant l'évolution de la situation - pas coûteuse en temps de calcul - occupe moins d'espace mémoire - mieux adapté pour les systèmes ou les variables sont dépendantes	- représentation complexe

Représentation par événements (trajectoires)	trajectoires en temps discret	séquence complète	- représentation simple - résultats plus précis et plus complets - efficace pour les variables indépendantes	- coûteuse en temps de calcul - occupe plus d'espace mémoire
		sous- séquence	- représentation simple - pas coûteuse en temps de calcul - occupe moins d'espace mémoire - prend en considération les différents raisonnements effectués durant l'évolution de la situation - efficace pour les variables indépendantes	- pour certains domaines cette représentation ne donne pas de bons résultats
	trajectoires en temps continu	séquence complète	- résultats plus précis et plus complets - efficace pour les variables indépendantes	- représentation complexe - coûteuse en temps de calcul - occupe plus d'espace mémoire
		sous- séquence	- prend en considération les différents raisonnements effectués durant l'évolution de la situation - pas coûteuse en temps de calcul - occupe moins d'espace mémoire - efficace pour les variables indépendantes	- représentation complexe

Tableau 2.1 : Avantages et inconvénients des différents types de représentation des cas dynamiques

2.6.1.2 Paramètres cible et source

Dans le raisonnement à partir de cas pour situations dynamiques, une situation est décrite par des historiques, qui représentent l'évolution des paramètres pertinents dans la prédiction de la situation. Ces paramètres peuvent être répartis en deux groupes différents. Le premier groupe est composé de paramètres exerçant une influence sur les valeurs de l'autre groupe de paramètres. Le second groupe est défini comme étant l'ensemble des paramètres soumis à l'influence des paramètres du premier groupe. Les paramètres du premier groupe sont appelés paramètres sources et les paramètres du second groupe paramètres cible (figure 2.8), ou encore événements source et événements cible [Rougegrez-Loriette, 1998]. Cette classification des différents paramètres est obtenue suite à l'application de la phase d'interprétation du cycle du raisonnement à partir de cas sur les paramètres originaux, qui décrit le domaine objet de l'étude.

Dans les systèmes de raisonnement à partir de cas pour situations dynamiques, le but du raisonnement est de déterminer l'évolution future de paramètres cible du cas cible en fonction de la ressemblance d'historiques des paramètres sources du cas cible avec les historiques des paramètres sources des cas sources de la base de cas.

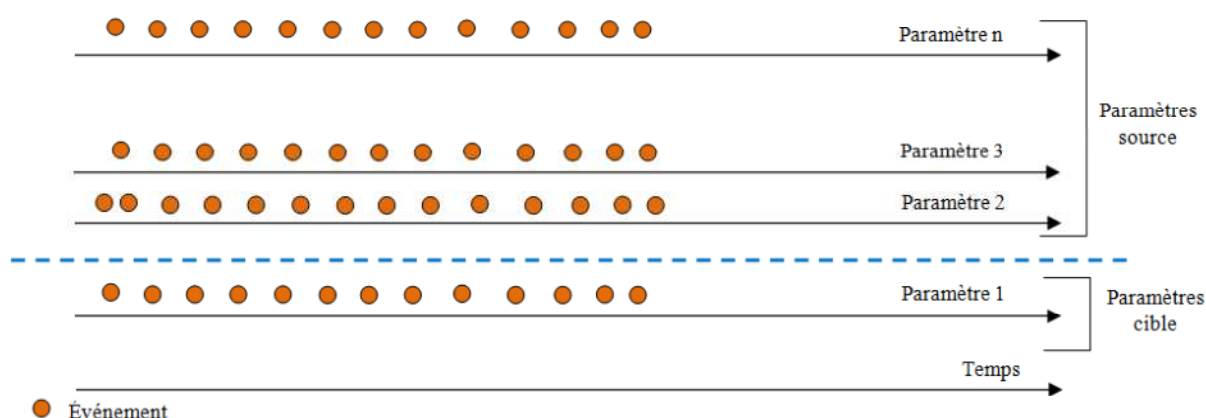


Figure 2.8 : Représentation d'un cas en paramètres cible et source

2.6.1.3 Paramètres passé et futur

En raisonnement à partir de cas traitant de situations statiques, un cas présente principalement deux parties : le problème et sa solution proposée. Dans certains systèmes en plus de ces deux parties, le cas comporte aussi une partie qui concerne les conséquences de son utilisation dans un raisonnement. Dans le raisonnement à partir de cas pour les situations dynamiques, cette distinction n'existe pas. Un cas est employé dans le cadre de l'appariement de ses historiques (ou de l'un de ses historiques), avec l'historique(s) du processus en cours, au moment où le raisonnement a lieu. Le but principal de ces systèmes est de prédire l'évolution future du processus, la partie de l'historique du cas source retournée (suite à l'étape de remémoration), décrit donc les conditions dans lesquelles la suite de l'historique s'est produite. Dans le raisonnement, l'énoncé du problème est représenté par la partie " passée " et la partie

de l'historique qui lui est postérieure représente donc la solution [Rougegrez-Loriette, 1998]. Cette partition des historiques en deux parties : une partie passée et une autre partie future correspond à la partition en couleur verte de la figure 2.9. Dans cette figure on a un exemple d'un cas décrit par des historiques, relatifs chacun à un paramètre ou variable. Ces paramètres sont jugés pertinents dans la prédiction de la situation.

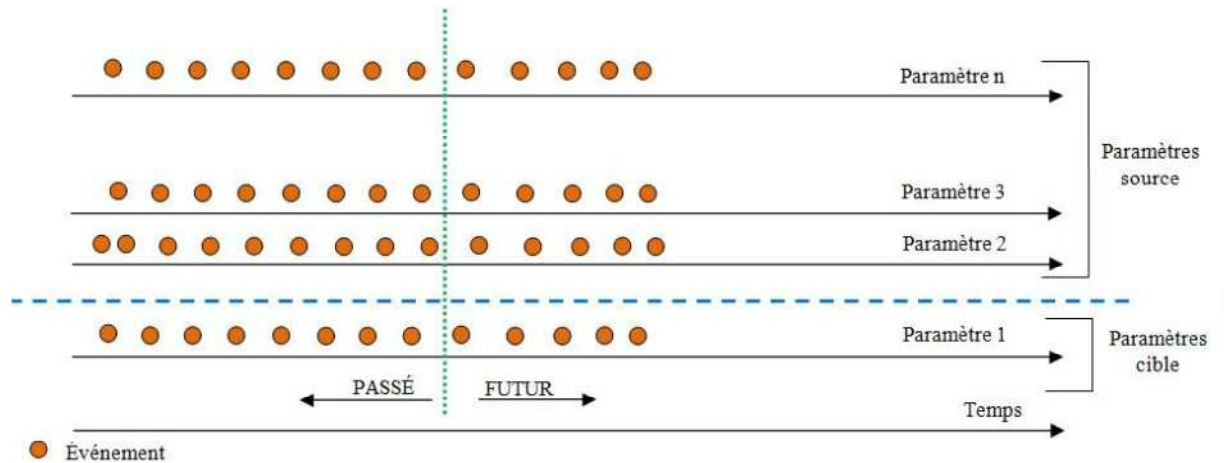


Figure 2.9 : Représentation d'un cas en paramètres passé et futur

La partition peut être fixe comme elle peut être variable, ceci dépend du cas en cours d'étude du système considéré. Plusieurs travaux de recherches ont adopté la partition fixe comme par exemple le système ICONS, où la partie " PASSÉ " de la figure 2.9, contient au maximum une description de la situation journalière relative à la dernière semaine. On trouve le second cas dans le système REBECAS : la partie passée du cas en cours d'étude est composée de tout l'historique de ce cas, de son début à l'instant de la prédiction. La partie " FUTUR " de la figure 2.9, qui est la prédiction que le raisonnement cherche à déterminer, est aussi de longueur fixe ou variable. Elle est par exemple fixée à 3 jours au maximum dans ICONS, par contre elle est illimitée dans le système REBECAS.

2.6.1.4 Rôle du temps

Le temps représente une grandeur importante pour ces systèmes en raison des rôles qu'il joue dans la représentation et le traitement des cas [Navarro et al., 2010]. D'une part le temps est un outil de représentation, permettant de distinguer les événements qui constituent un historique. En effet, un événement est lié à une ou plusieurs valeurs de paramètres. L'indice temporel (un instant par exemple) permet de distinguer sans ambiguïté ces différents événements. D'autre part le temps permet d'établir une corrélation entre les faits, et permet donc d'établir une chronologie du processus considéré. Le temps a un autre rôle, très important aussi dans ce raisonnement. En effet il permet d'indiquer qu'un événement d'un cas correspond au passé du cas en cours d'étude, ou au contraire qu'il peut correspondre à son futur (celui que l'on cherche à déterminer). Cette distinction est réalisée par l'intermédiaire de frontières représentées dans la figure 2.9 par la partition en couleur verte.

2.6.2 Base de cas pour les RàPCD

2.6.2.1 Organisation de la base de cas pour RàPCD

La base de cas où la mémoire du système est une unité d'archivage des cas sources. La structure de la mémoire dépend d'une part de la représentation des cas choisie pour le système à développer et d'autre part des buts de ce système. Dans les systèmes RàPCD, comme pour le cas statique, on distingue deux types d'organisation de la mémoire : une base de cas plate ou hiérarchique [Malek, 1996]. Le processus d'apprentissage dans la structure plate est très simple car il s'agit d'un simple ajout du nouveau cas résolu à la mémoire. Pourtant, cette structure présente une limite au niveau du temps de recherche, qui augmente linéairement avec la taille de la mémoire. On trouve plusieurs systèmes qui ont utilisé ce type d'organisation tels que REBECAS, AuRA et SINS. Dans un système à mémoire hiérarchique, les cas sont structurés dans une organisation hiérarchique composée de deux niveaux. Le bas niveau est une mémoire plate structurée en groupes de cas similaires. Le haut niveau comprend des index, appelés également des clusters, ou encore des prototypes, représentant les propriétés communes d'un groupe de cas similaires stockés dans le bas niveau [Aich, 2007]. L'indexation est accomplie grâce aux méthodes utilisées en apprentissage automatique telles que les arbres de décision, les logiques de description, et les réseaux de neurones, etc. Nous renvoyons le lecteur à [Aich, 2007] pour plus de détails sur ces méthodes.

Remarque :

Dans les systèmes où la représentation des cas est une sous-séquence, la mémoire des systèmes comporte en plus de l'ensemble des cas (les sous-séquences), une autre partie pour stocker les séquences complètes. Par exemple, la mémoire du système CASEP2 comprend deux parties : une pour le stockage des navigations complètes appelée la base de navigations, et une autre composée des sous-navigations (les cas) appelée la base de cas. Dans ce contexte le terme cas est réservé au type d'entités représentant l'unité de base du raisonnement.

2.6.2.2 Maintenance de la base de cas

La maintenance de la base de cas met en application des stratégies de mise à jour de la base de cas, permettant de réviser son organisation et/ou son contenu afin d'améliorer le raisonnement futur du système. La maintenance permet donc d'améliorer et d'optimiser les performances des systèmes, particulièrement pour les systèmes qui sont conçus pour fonctionner sur de longues périodes et qui sont amenés à traiter de grands volumes de données.

Il existe plusieurs stratégies qui déterminent quand et comment un système de RàPC ou RàPCD exécute la maintenance de sa base de cas. Ces méthodes sont décrites par [Leake et Wilson, 1998] et dépendent :

- de la manière de rassembler les données pertinentes pour la maintenance ;
- du moment de déclenchement de la maintenance ;

- des types d'opération de maintenance possibles ;
- de la manière d'exécuter les opérations sélectionnées.

Une base de cas est considérée de bonne qualité si elle permet au système de résoudre le maximum des problèmes possibles de manière correcte en un temps raisonnable [Zehraoui, 2004]. La qualité de la base de cas peut être mesurée en se basant sur les deux critères suivants [Smyth et Keane, 1995] :

1. La compétence, d'un système est mesurée par le nombre de problèmes différents pour lesquels le système apporte une solution efficace.
2. La performance, est mesurée par le temps de réponse nécessaire pour proposer une solution à un cas cible. Cette mesure dépend de la méthode de recherche, ainsi que le coût d'adaptation.

On distingue deux catégories pour la maintenance de base de cas :

- **Stratégie de partitionnement de la base de cas** : dans ce type, il n'y a pas de suppression de cas, mais ils sont partitionnés en plusieurs espaces de recherche, ce qui permet de réduire le temps de recherche et de sélectionner de manière incrémentale les attributs qui sont riches en information et qui peuvent couvrir la structure de la base de cas entière [Yang et Wu, 2000].
- **Stratégie d'optimisation de la base de cas** : cette politique se base sur la réduction de la taille de la base de cas, ce qui diminue le temps de recherche. Cette stratégie consiste à optimiser le contenu de la base de cas suivant deux façons différentes : une stratégie d'ajout de cas et une autre de suppression de cas incohérents [Haouchine, 2009]. Dans chacune d'elle, différentes méthodes sont développées et évaluent la base de cas sous des aspects de compétence [Smyth et McKenna, 1998] ou de performance [Leake et Wilson, 2000].

Stratégie d'ajout de cas

La stratégie d'ajout de cas consiste à construire une base de cas réduite en partant d'une base de cas vide par l'ajout consécutif de cas présentant un critère maximisé de compétence ou de performance, pour plus de détails nous renvoyons le lecteur à [Haouchine, 2006]. Parmi les systèmes qui ont adoptés ce type de maintenance on peut citer le système CASEP2. En effet les concepteurs du système CASEP2 ont proposé une stratégie de maintenance périodique après un certain nombre de navigations enregistrées [Zehraoui, 2003]. Elle consiste à construire, à partir de la base de cas courante, une base de cas réduite (initialement vide) en lui ajoutant au fur et à mesure des cas. Le système utilise deux mesures afin de calculer la pertinence d'un cas : son étendue qui est le nombre de problèmes pour lesquels ce cas a proposé une solution et sa qualité qui représente le taux de ses réussites durant la résolution de nouveaux cas cibles. La stratégie de maintenance dans CASEP2 consiste à garder les cas qui ont de grandes valeurs d'étendues et à défavoriser les autres.

Stratégie de suppression de cas

Cette stratégie consiste à évaluer les cas suivant un critère pour pouvoir les supprimer et ramener la base de cas à un nombre de cas donné. Il existe plusieurs critères d'évaluation tels que la compétence, la redondance et l'inconsistance. Plusieurs travaux de RàPC et RàPCD ont utilisé ces différentes méthodes afin de réduire le contenu de la base de cas et améliorer les performances des systèmes. Par exemple dans le système BROADWAY, l'optimisation du contenu de la base de cas est limitée à la suppression des cas qui ne sont pas utilisés pendant une longue période (ce sont les cas obsolètes). La même chose pour le système SINS, qui limite le nombre de cas de la base afin d'empêcher la création d'une base de cas trop lente.

2.6.3 Le cycle de vie de RàPCD

Le cycle de vie d'un système de RàPCD (figure 2.10) peut être divisé en trois phases principales [Saxena, 2007].

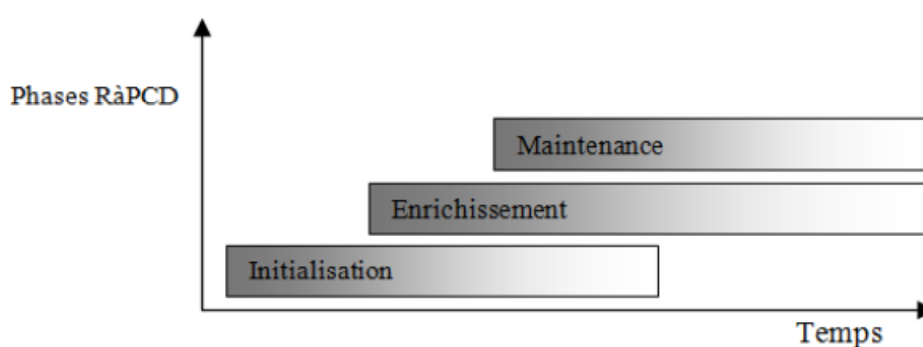


Figure 2.10 : Cycle de vie d'un système RàPCD [Saxena, 2007]

Phase d'initialisation : cette phase comprend la conception de la représentation générale d'information, le choix des caractéristiques appropriées pour l'indexation et le mode d'interaction entre l'application et le système réel. Cette phase comprend aussi l'installation du système RàPCD : au départ le système utilise une base de cas initiale constituée essentiellement des cas prototypes pour fournir une base pour l'application. Cette base de cas initiale contient un nombre limité de cas, proposés par les experts du domaine et par conséquent ne peut pas couvrir toute la gamme des problèmes.

Phase d'enrichissement : la base de cas initiale est utilisée et raffinée à travers les étapes de validation et de mémorisation. Il s'agit d'enrichir la base de cas avec les nouveaux cas au fur et à mesure qu'ils sont rencontrés. C'est un processus continu tout au long de la vie d'un système de RàPCD, et il reste relativement actif jusqu'à ce que l'on obtienne une base de cas presque complète, couvrant une grande partie du domaine d'application.

Phase de maintenance de la base de cas : c'est un processus relativement moins fréquent, mais continu tout au long de la vie du système pour faire face à la non-stationnarité de l'application. On s'attend à ce que cette phase débute après que la phase d'enrichissement soit accomplie. Il existe plusieurs techniques et méthodes susceptibles d'être utilisées en maintenance de la base de cas (paragraphe 2.6.2.2 Maintenance de la base de cas).

2.6.4 Cycle du raisonnement à partir de cas pour les situations dynamiques

2.6.4.1 Cycle du raisonnement statique pour RàPCD

Nous décrivons dans cette section les différentes étapes du raisonnement induit par les systèmes RàPCD pour les situations dynamiques (cas cible/sources dynamiques) et cycle statique. Le raisonnement est fort semblable à celui adopté par les systèmes RàPC pour situations statiques. Néanmoins, des améliorations existent dans le contenu des étapes qu'il comporte pour s'adapter au dynamisme des systèmes.

Étape d'élaboration

Cette phase appelée aussi phase d'interprétation se traduit par une abstraction de l'historique décrit originellement, sous forme de séquences d'événements ou d'états. Cette abstraction doit permettre de représenter le cas en cours d'étude sous une forme qui le rend comparable aux autres cas de la base. Par exemple pour le système REBECAS cette étape transforme le paramètre relief décrit originellement sous forme numérique, en une succession de formes de relief symboliques (exemple crête, col, falaise, etc.) et pour le vent, il s'agit d'exprimer les variations entre deux mesures successives plutôt que leurs valeurs absolues. Enfin cette étape permet de vérifier les longueurs des séquences à comparer.

Étape de la remémoration

Cette phase de raisonnement a pour but de retrouver les cas sources les plus similaires à la situation en cours. Nous distinguons deux principes de recherches pour les systèmes de raisonnement à partir de cas pour les situations dynamiques :

1. Une recherche par l'évaluation de similarité entre des séquences à une seule dimension, appelée également recherche par point de vue [Rougegrez-Loriette, 1998]. Elle s'exécute en différentes étapes. Chacune permet d'évaluer les similarités entre le cas cible et les cas sources selon une seule variable ou paramètre. Le choix du meilleur cas à réutiliser dépend des résultats obtenus dans les différentes étapes. Plusieurs systèmes ont utilisé ce principe de mesure de similarité entre le nouveau cas cible et les cas sources REBECAS, BROADWAY, SAPED et AuRA.
2. Une recherche par l'évaluation de similarité entre des séquences multidimensionnelles [Aich, 2007]. Elle se réalise en une seule étape en tenant compte de tous les paramètres décrivant les cas en même temps. Ce principe de recherche a été utilisé par le système CASEP2 (les entrées sont des matrices de valeurs).

La recherche selon une seule dimension (selon un point de vue) est plus efficace dans le cas où les variables d'observations sont indépendantes. Ce principe permet aussi de simplifier l'étape de recherche du raisonnement et plus spécialement si l'un des types des variables décrivant les cas est numérique [Aich, 2007].

2.6.4.2 Mesures de similarité dans le cas dynamique et cycle statique

La recherche des cas similaires se base sur la mesure de similarité. Il existe de nombreuses mesures qui dépendent essentiellement de la représentation de cas. Nous présentons dans cette partie des mesures de similarités souvent utilisées [Aich, 2007].

- **Alignement de séquences biologiques par programmation dynamique**

L'alignement de séquences est un outil important, qui a été utilisé pour de nombreuses applications de biologie telle que l'analyse phylogénétique, la découverte de gènes, etc. Ces séquences ont la particularité de représenter la structure primaire des macromolécules comme l'ADN et les protéines. Elles sont composées de suites finies d'alphabet de quatre lettres pour l'ADN, et de vingt lettres pour les protéines (les acides aminés).

L'alignement de deux séquences comparées consiste à les transformer en deux autres séquences de la même longueur. Le but est de trouver les transformations nécessaires pour passer d'une séquence à une autre.

Une transformation est appelée une *édition* et elle peut être soit une insertion ou une suppression d'un symbole, soit une substitution d'un symbole d'une séquence en une autre à la même position [Charras et Lecroq, 1998]. Pour chaque opération d'édition on lui associe un coût. Soit les deux séquences suivantes $A[a_1, a_2, \dots, a_n]$ et $B[b_1, b_2, \dots, b_m]$ de longueurs respectives n et m . Needleman et Wunsch [Needleman et Wunsch, 1970], ont proposé une solution pour aligner les deux séquences en se basant sur la programmation dynamique. Cette solution est basée sur les trois étapes suivantes :

1. Cette étape est réservée au choix des coûts des différentes éditions : insertion d'un nouveau symbole dans la séquence A ($\text{Insertion}(a_i)$) ou suppression d'un symbole de la séquence B ($\text{Suppression}(b_j)$), substitution d'un symbole de la séquence "A" par un autre de la séquence "B" ($\text{substitution}(a_i, b_j)$).
2. Dans cette phase on procède au remplissage de la matrice d'édits "EDIT" utilisée par la suite pour aligner les deux séquences. L'élément "EDIT[i, j]" qui est la valeur stockée dans la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne de la matrice "EDIT", est égale au minimum entre les trois valeurs ($\text{EDIT}[i-1, j-1] + \text{Substitution}(a_i, b_j)$), ($\text{EDIT}[i, j-1] + \text{Insertion}(b_j)$) et ($\text{EDIT}[i-1, j] + \text{Suppression}(a_i)$).
3. Dans cette dernière étape, on calcule l'alignement des deux séquences à partir de la matrice "EDIT" : il faut partir de l'élément $\text{EDIT}[n+1, m+1]$, et procéder par le chemin inverse vers l'élément $\text{EDIT}[1, 1]$, dans chaque étape intermédiaire on vérifie à partir de quel voisin l'élément $\text{EDIT}[i, j]$ a été calculé. S'il s'agissait de l'élément $\text{EDIT}[i-1, j-1]$ alors dans ce cas a_i est aligné avec b_j , s'il s'agissait de l'élément $\text{EDIT}[i-1, j]$, alors a_i est aligné avec le symbole "/", sinon b_j est aligné avec le symbole "/".

Cette mesure permet de comparer des séquences de longueur différentes, selon un seul point de vue et pour des types de séquences symboliques.

- **Distance de Minkowski**

Soit les deux séquences $A[a_1, a_2, \dots, a_n]$ et $B[b_1, b_2, \dots, b_n]$ de même longueur n . Ces deux séquences peuvent être comparées grâce à une distance métrique L_p appelée distance de Minkowski. Elle est calculée par la formule suivante :

$$L_p = (\sum_{i=1}^n |a_i - b_i|^p)^{\frac{1}{p}}$$

Le calcul de la distance L_p est en $O(n)$ où n est la longueur des séquences comparées.

Pour $p = 2$, on obtient la distance euclidienne.

La distance de Minkowski permet de comparer des séquences de même longueur, selon un seul point de vue et pour des séquences de valeurs numériques.

- **Recherche de la plus longue sous-séquence commune (Longest Common SubSequence LCSS)**

Cette mesure permet de comparer des séquences multidimensionnelles, elle est basée sur le principe de la plus longue sous-séquence commune et elle a été utilisée par Vlachos [Vlachos et al., 2002]. Le but est de compter le nombre de couples de points considérés comme similaires lorsqu'on parcourt les deux séquences comparées.

La similarité entre deux points est déterminée selon un seuil σ : une valeur a_i d'une séquence A est considérée similaire à une valeur b_j d'une autre séquence B si $|a_i - b_j|$ est inférieure ou égale au seuil σ .

On définit aussi une constante β qui sert à contrôler l'écart de temps maximum entre deux points de chacune des séquences pour qu'ils puissent être similaires.

Soit les deux séquences temporelles (A, B) , un entier σ et un réel β . On définit la mesure de similarité entre les deux séquences, qu'on note $LCSS_{(\sigma, \beta)}(A, B)$, selon le processus récursif suivant : on initialise le processus par les deux derniers points $(fin(A), fin(B))$ des séquences (A, B) comparées. Si l'une des deux séquences est vide alors la mesure de similarité est nulle et le processus s'arrête. Sinon, si les deux derniers points des séquences sont similaires, alors la mesure de similarité dans ce cas vaut " 1 " plus la similarité calculée entre les deux séquences privées de leurs derniers points $(reste(A), reste(B))$. Enfin, dans le cas où les deux derniers points des séquences A et B ne sont pas similaires, alors la mesure de similarité est égale au maximum de la similarité entre une séquence et l'autre privée de son dernier point [Aich, 2007]. La formule suivante présente le principe de mesure de similarité entre les deux séquences A et B .

$$LCSS_{(\alpha, \beta)}(A, B) = \begin{cases} 0 & \text{si } A \text{ ou } B \text{ est vide} \\ 1 + LCSS(reste(A), reste(B)) & \text{si } |fin(A) - fin(B)| < \delta \\ \max\{LCSS(reste(A), B), LCSS(A, reste(B))\} & \text{sinon} \end{cases}$$

La mesure LCSS permet de comparer des séquences de longueur différentes, pour des séquences multidimensionnelles et pour des types des séquences hétérogènes.

Il existe d'autres mesures de similarité comme la distance par déformation temporelle (*Dynamic Time Warping : DTW*) proposée au début dans le domaine de la

reconnaissance de la parole [Yehia et al., 1998]. Elle est utilisée actuellement dans plusieurs travaux de comparaison de séquences temporelles.

Différentes mesures de similarités entre des séquences ont été présentées dans cette partie du manuscrit. Le tableau 2.2 présente une étude comparative entre les différentes mesures présentées à partir de trois critères [Aich, 2007] :

1. Les types des séquences comparées : numériques, symboliques ou hétérogènes.
2. La dimension des séquences : unidimensionnelle si les données proviennent d'un seul paramètre, ou multidimensionnelle dans le cas contraire.
3. Les longueurs des séquences comparées.

	Type des données	Dimension	Longueur
Algorithmes d'alignement	symbolique	unidimensionnelle	différente
DTW	numérique	unidimensionnelle	différente
LCSS	hétérogène	multidimensionnelle	différente
Distance de Minkowski	numérique	unidimensionnelle	même longueur

Tableau 2.2. Comparaison de différentes mesures de similarités entre séquences [Aich, 2007]

Étape d'adaptation / réutilisation des cas

Cette phase du cycle de raisonnement consiste à réutiliser les cas sources retrouvés durant l'étape de recherche pour proposer une nouvelle solution au cas cible. L'étape d'adaptation est absente pour certains systèmes de RàPC. En effet de nombreux chercheurs du domaine pensent que l'adaptation est une tâche très difficile, et qu'elle dépend fortement du domaine d'application. D'autres au contraire considèrent que l'adaptation est fondamentale pour les systèmes de RàPC et que ces derniers ne peuvent pas se concevoir sans prise en compte de l'adaptation [Mille, 1996]. Nous renvoyons le lecteur à [Kolodner, 1993] pour plus de détails. Ci-après nous présentons les techniques les plus utilisées dans le domaine :

- **L'adaptation par substitution** : dans cette méthode, la solution du cas source retrouvée est utilisée directement pour résoudre le nouveau problème (le cas cible). L'expert ou l'utilisateur peuvent intervenir au niveau de la solution proposée et la modifier.
- **L'adaptation par transformation de la solution** : la solution du cas source remémorée dans l'étape précédente est transformée en une nouvelle solution. Cette méthode est réalisée grâce à l'application de règles d'adaptation, ce qui nécessite d'avoir des connaissances sur le domaine d'application, permettant de savoir comment des différences au niveau de deux cas cibles peuvent être considérées au niveau de leurs solutions. On distingue deux types de transformations, soit l'adaptation des valeurs des paramètres, ou bien l'adaptation de la structure de la solution du cas source. Plusieurs systèmes utilisent une adaptation par transformation de la solution tels que : SINS, AuRA et REBECAS.

- **L'adaptation générative** : cette méthode peut être utilisée si on dispose pour chaque cas source stocké dans la base de cas, des traces de raisonnement menant aux solutions. L'adaptation générative consiste à appliquer le même raisonnement au nouveau problème. En effet ces traces de raisonnement seront re-exploitées et rejouées pour le cas cible. Ce type d'adaptation a été utilisée dans les deux systèmes CASEP2 et BROADWAY.
- **L'adaptation hiérarchique** : ce type d'adaptation nécessite que les cas sources soient stockés d'une manière hiérarchique sur plusieurs niveaux d'abstraction. La solution du cas source remémorée en phase de recherche sera d'abord adaptée au niveau d'abstraction le plus élevé, puis une démarche descendante est appliquée : le cas est adapté à des niveaux plus fins où certaines informations sont ajoutées.

Étape de révision

L'objectif de l'étape de révision est d'évaluer la solution proposée et de la corriger si elle est insatisfaisante. Elle peut être réalisée soit de manière interactive avec l'intervention des experts soit d'une manière automatique en se basant sur les connaissances du domaine. Certains systèmes de RàPC ont proposé d'associer aux cas sources des indices indiquant la qualité de leurs utilisations dans le cycle de raisonnement (succès ou échecs). Ces indices permettent de préciser l'efficacité d'un cas source pendant leur utilisation et leur adaptation afin de résoudre de nouveaux cas cibles. Après chaque cycle de raisonnement, le système procède à la mise à jour de ces indices. On peut citer l'exemple du système BROADWAY : l'utilisateur peut évaluer les pages web proposées par le système afin de les visiter. Cette évaluation se déroule pendant la phase de révision et se poursuit jusqu'à la fermeture de la navigation courante. Si un cas a proposé au moins une page parmi l'ensemble des pages recommandées confirmées, alors son nombre de succès est incrémenté. Pour les systèmes SAPED, CASEP2 et SINS chaque cas contient deux attributs succès et échec représentant sa qualité. Ces attributs permettent d'évaluer les solutions proposées. Les systèmes procèdent à la mise à jour de ces attributs après chaque cycle de raisonnement. Pour le système CASEP2, ces attributs sont utilisés aussi lors de la maintenance de la base de cas.

Étape d'apprentissage

L'apprentissage consiste à enrichir la base des cas, par l'intégration des nouveaux cas après la proposition des solutions par le système pour ces derniers. La littérature propose plusieurs types d'apprentissage. Ci-après nous présentons les techniques les plus utilisées :

1. **L'ajout du nouveau cas cible dans la base de cas** : si la similarité entre le nouveau cas cible et le meilleur cas source remémoré lors de la phase de recherche est éloignée, alors le cas cible doit être inséré dans la base de cas. Par exemple pour les deux systèmes SINS et AuRA, le nouveau cas cible est

sauvegardé dans la mémoire s'il n'est pas trop similaire au cas source retrouvé lors de la phase de remémoration;

2. **La modification du cas source retrouvé dans la phase de recherche** : si le cas cible et le cas source sont suffisamment similaires, alors le cas source est modifié ou complété, sans insérer le cas cible dans la base des cas. Cette stratégie présente certains inconvénients, d'une part le nombre de cas à insérer dans la mémoire sera très limité puisque les nouveaux cas cibles résolus ne sont pas tous insérés, d'autre part le risque de perdre ceux qui ont été appris durant les tous premiers raisonnements. Avec cette stratégie d'apprentissage, il est préférable de limiter le nombre de modifications des cas sources ;
3. **La révision de la structure de la mémoire** : dans le cas d'une mémoire hiérarchique, une réorganisation de cette dernière est nécessaire après chaque ajout de nouveaux cas. Par exemple pour les nouveaux cas appris par le système CASEP2 durant son utilisation, ils sont insérés dans une zone de mémoire appelée *la zone atypique*. Si la taille de cette zone dépasse un certain seuil, alors le réseau de neurones exploité par le système CASEP2, procède à l'insertion de ces cas dans la base en leur affectant un neurone comme index.

2.6.5 Etude comparatif des principaux systèmes RàPCD

Il existe plusieurs systèmes de raisonnement à partir de cas pour situations dynamiques, concernées principalement par les problèmes de prédiction. Nous présentons dans la section suivante différents systèmes avec leur principe et les méthodes déployées. Nous nous intéressons plus particulièrement aux points suivants : la représentation du cas, la phase de remémoration, la mesure de similarité, etc. Ces informations sont résumées dans un tableau de la forme suivante :

Nom du système [références]	objectif du système et domaine d'application : le but de l'approche ainsi que le domaine d'application du système.
Représentation de cas	
Remémoration / Mesure de similarité	
Description de l'approche	

2.6.5.1 REBECAS

REBECAS [Rougegrez-Loriette, 1994]	le but du système REBECAS (pREdiction BasÉE sur des CAS) est la prédiction de la propagation du feu de forêt. Il permet de déterminer à quel endroit se situera le front de flammes à un instant donné.
Représentation de cas	séquence complète.
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des historiques de valeurs (programmation dynamique).
Description de	dans REBECAS, le cas est une direction de propagation d'un feu entier depuis

l'approche

son départ jusqu'à sa maîtrise (ou axe de propagation). Un cas est relatif à une seule direction d'un incendie unique, c'est-à-dire associé à un seul départ de feu. La partie problème dans un cas en système REBECAS représente de façon continue l'évolution des valeurs des différents paramètres pertinents d'observations (figure 2.11) : le relief, la végétation, et le vent. La propagation du feu représente la partie solution.

Dans chaque cas, un ensemble d'historiques (des événements) est décrit pour chaque variable observée. Un historique du relief représente une hauteur et une forme caractéristique de ce dernier. Un historique de la végétation représente un type de végétation (vignes, broussailles, conifères, etc.). Un historique du vent décrit le changement de vitesse du vent à partir du dernier événement décrit et il décrit aussi la distance et l'heure associées.

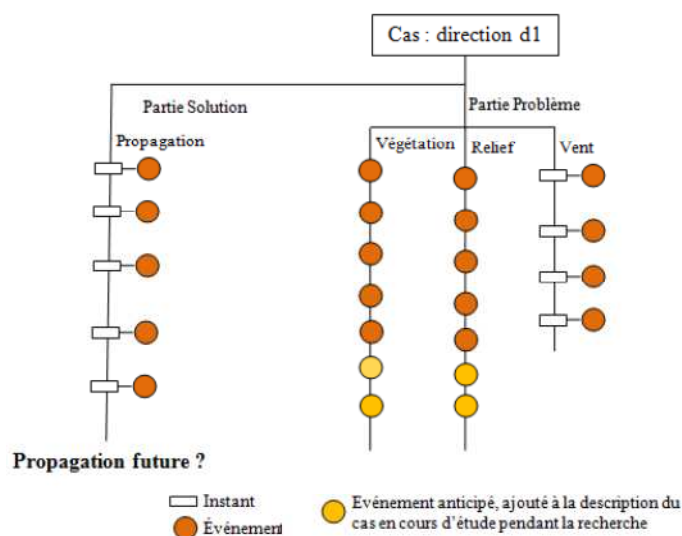


Figure 2.11 : La représentation d'un cas dans le système REBECAS suivant une direction de propagation

Les cas sources sont créés à partir des comptes rendus que rédigent les sapeurs pompiers après chaque sinistre important. La figure 2.11 présente la représentation d'un cas suivant une direction de propagation. Le nouveau cas cible (la situation) est représenté par les différents historiques de l'évolution des valeurs des variables (relief, vent et végétation) depuis le départ du feu dans une direction donnée. La phase de recherche dans REBECAS permet de retrouver un cas qui présente les historiques d'événements les plus similaires à ceux du feu en cours. Cette recherche s'effectue selon chaque point de vue. Pour cela des algorithmes de programmation dynamique ont été utilisés afin de mettre en correspondance ces historiques. Avant de procéder à la recherche du cas similaire, une étape d'élaboration pour la mise en adéquation des séquences est effectuée. Pour le paramètre relief, cette étape se traduit par la transformation des événements représentés initialement par des valeurs numériques, en une succession de formes de relief symboliques (crête, col, falaise, etc.). Pour le vent, l'étape d'élaboration permet de transformer les variations du vent représentées initialement par des valeurs absolues en une variation entre deux mesures successives. Ensuite la mise en adéquation se poursuit par la vérification des longueurs des séquences à comparer.

Après cette étape d'élaboration, la recherche de la meilleure séquence consiste à évaluer la distance entre la situation en cours et les différents cas sources en utilisant la programmation dynamique. La meilleure séquence retrouvée dans chaque direction, suivant chaque point de vue, est alors retournée. A la fin de l'étape de recherche, un seul cas doit être choisi, en se basant sur les deux critères

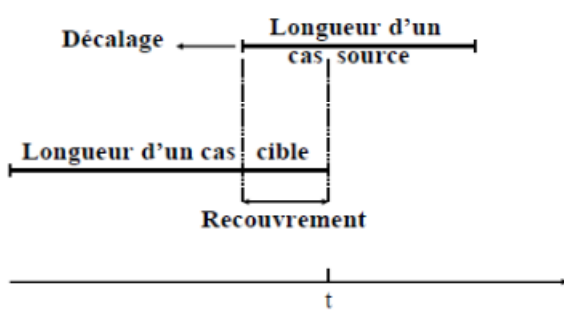
	suivants : les distances calculées entre le cas cible et les cas sources pour les différents paramètres (d_i), et les décalages temporels entre les séquences des variables des cas sources considérées similaires à celle du cas cible. Le meilleur cas est donc celui qui minimise la somme des distances (d_i) et qui minimise aussi les décalages temporels entre ces séquences.
--	--

2.6.5.2 AuRA

AuRA [Likhachev et al., 2002]	le système AuRA (<i>Autonomous of Robot Architecture</i>) a été développé afin de commander de manière réactive un robot mobile dans un environnement réel.
Représentation de cas	sous-séquences de longueurs différentes
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des historiques de valeurs (distance euclidienne).
Description de l'approche	le principe de fonctionnement du système AuRA se base sur des architectures réactives telles que "avancer vers le but", "éviter un obstacle", etc. Le principe de navigation d'un robot adopté par le système AuRA vers un but prédéfini, consiste à générer un plan à suivre par le robot pour atteindre ce but [Aich, 2007]. Un cas dans le système AuRA est composé de deux parties : la partie problème qui représente la situation du fauteuil tel que les distances des obstacles observés autour du fauteuil et la position du robot par rapport au but, et la partie solution représentée par les différents paramètres des schémas de réaction. Au départ, le robot commence à se déplacer en déclenchant le schéma de réaction "aller vers le but". Ensuite, en se déplaçant vers ce but, le système peut détecter des obstacles proche du fauteuil alors dans ce cas il exécute le cycle de raisonnement du système AuRA afin d'adapter le déplacement du robot. Le but est de trouver des nouveaux schémas de réaction permettant d'éviter les obstacles rencontrés et par conséquent de conduire le robot vers le but. Dans le système AuRA, la phase de recherche est composée des quatre étapes suivantes : • préparation des données : la situation initiale du robot représentée par la partie problème du cas cible est décrite par les deux vecteurs suivantes : le premier vecteur définit les variables spatiales de l'environnement. L'auteur du système AuRA, propose de diviser l'espace libre autour du robot en "k" zones. Ensuite il procède au remplissage d'un vecteur nommé vecteur des caractéristiques spatiales par les valeurs de probabilité de traverser chacune de ces zones. Le second vecteur (vecteur des caractéristiques temporelles) est composé des vitesses du robot à court et long terme normalisées par sa vitesse maximale. Pour plus de détails sur le principe de calcul de ces deux vecteurs, nous renvoyons le lecteur à [Likhachev et Arkin, 2001], • mesure de similarité spatiale : dans cette étape, la base de cas plate du système AuRA est examinée entièrement pour extraire des cas sources qui présentent des similarités spatiales avec le nouveau cas cible. Le système AuRA utilise la distance euclidienne afin de comparer le vecteur des caractéristiques spatiales du nouveau cas cible avec tous les cas sources de la base de cas, • mesure de similarité temporelle : dans cette phase les cas sources trouvés dans l'étape précédente sont tous parcourus afin de retrouver ceux qui présentent une similarité temporelle par rapport au cas cible. Le système

	AuRA utilise la distance euclidienne afin de comparer le vecteur des caractéristiques temporelles du nouveau cas cible avec tous les cas sélectionnés lors de l'étape précédente. Les cas sources retournés après cette étape présentent donc des similarités spatio-temporelles avec le nouveau cas cible. • dans cette dernière phase de l'étape de recherche, un cas est sélectionné au hasard parmi les cas sélectionnés lors de l'étape précédente.
--	--

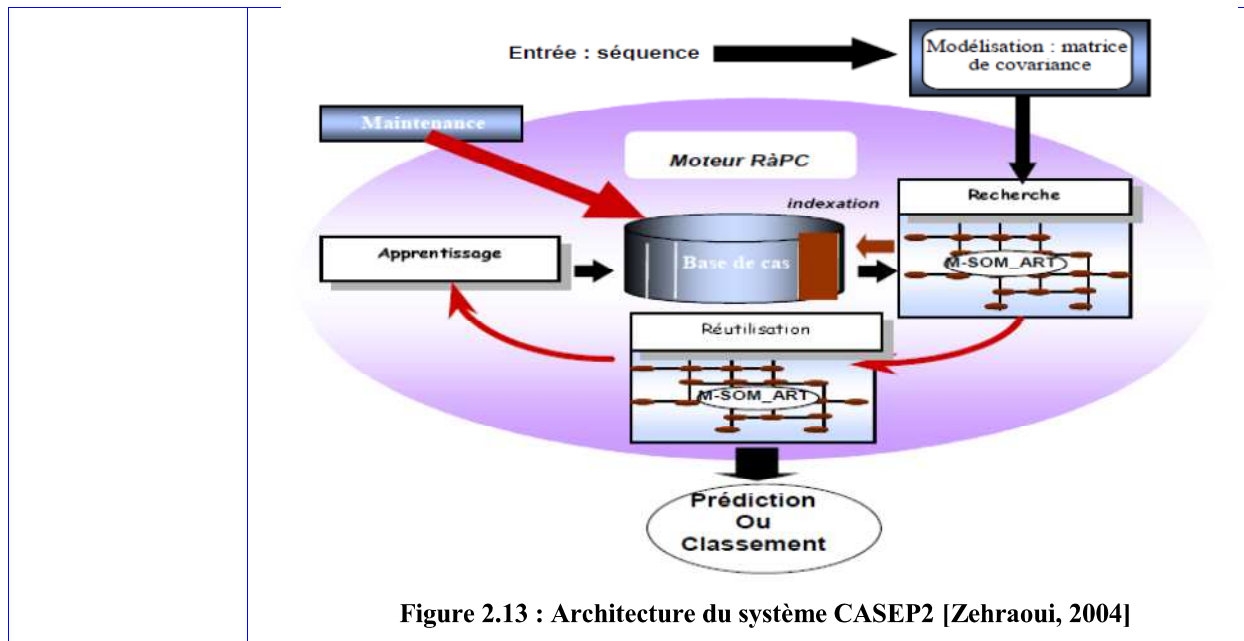
2.6.5.3 SINS

SINS [Ram et Santamaria, 1993]	le système SINS (<i>Self-Improving Navigation System</i>) a pour but de guider un robot mobile grâce à des schémas de réaction.
Représentation de cas	sous-séquences de la même longueur
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des historiques de valeurs (écarts quadratiques).
Description de l'approche	SINS a été créé avec le même principe et objectif que le système AuRA. Les deux systèmes sont différents au niveau de la représentation des cas et la méthode de sélection des cas sources similaires au nouveau cas cible. Pour le système SINS, les séquences des paramètres représentant les cas sont fixes. En effet un nouveau cas cible est représenté par les évolutions des valeurs des paramètres pour une durée déterminée, et de la même façon, un cas source représente les évolutions des variables issues d'expériences passées pour une durée donnée. Dans le système AuRA, les longueurs des séquences de valeurs décrivant la partie problème des différents cas sont différentes. Dans SINS, la partie problème d'un cas est représentée par un historique d'évolution de valeurs numériques pour chaque paramètre ou variable d'observation. Les développeurs du système proposent un algorithme de recherche capable d'identifier le cas source dont le recouvrement (figure 2.12) avec le nouveau cas cible est optimal [Aich, 2007]. En effet, le système procède par une mise en correspondance progressive de chaque cas source en remontant dans le temps. La mesure de similarité utilisée par le système repose sur le calcul des écarts quadratiques sur chaque variable (toutes les valeurs des variables des cas sont numériques). Ensuite ils sont agrégés par une moyenne pondérée.  Figure 2.12 : Principe de recherche dans SINS

2.6.5.4 CASEP2

CASEP2	le système CASEP2 (<i>CAs e-based reasoning System for SEquence Prediction</i>) a
---------------	---

[Zehraoui, 2004]	été développé pour la prédiction des comportements d'un internaute à partir de traces de navigations. Le but principal du système CASEP2 est d'adapter de façon dynamique un site web de commerce électronique.
Représentation de cas	sous-séquence de longueurs différentes
Remémoration / Mesure de similarité	recherche multidimensionnelles / comparaison des matrices de covariance dynamiques.
Description de l'approche	un cas, dans CASEP, représente une expérience précise (une sous-séquence) dans la séquence navigation. Une navigation est un ensemble ordonné d'états e_i, correspondant à des informations sur la page visitée par l'utilisateur. Chaque état contient un ensemble de valeurs v_i ($1 \leq i \leq n$) prises par n variables (les caractéristiques) tels que l'adresse de la page, sa taille, son rang dans la navigation etc. Le nombre de pages visitées ou encore le nombre d'états représente la longueur d'une navigation. La partie problème d'un nouveau cas cible correspond aux derniers états e_i de la séquence courante. Elle est représentée par une succession d'états e_i (une sous-navigation) pour les cas sources. Ces sous-séquences sont de longueurs (nombre d'états) différentes. Le but du CASEP2 est d'assister les visiteurs de sites Web dans leur navigation par la proposition d'une succession de pages recommandées à visiter, en prenant en considération les pages déjà visités. La partie solution d'un cas source est un ensemble ordonné d'états e_i correspondant à des pages web à visiter. Parmi les buts du CASEP2 la classification des utilisateurs en catégories d'acheteur ou non acheteur. Cette information de classification est intégrée dans la partie solution des cas. Le système CASEP2 est composé de deux modules (l'architecture du système est présenté dans la figure 2.13) : un module de RàPC et un réseau de neurones. Le réseau permet d'indexer la mémoire. Elle est composée de deux bases : - la base de séquences qui contient des séquences entières ou base de navigations. - la base de cas qui comporte des sous-séquences extraites de ces navigations. La phase de recherche dans le système CASEP2 consiste d'une part à prédire les futures pages web à visiter afin de les recommander à l'utilisateur, et d'autre part à déterminer la classe de l'utilisateur (acheteur ou non acheteur). Initialement, un cas, la séquence décrivant la succession des pages visitées, est modélisé par une matrice de covariance dynamique qui prend en considération d'une part la distribution des états de la séquence et d'autre part leur ordre temporel [Zehraoui et Bennani, 2004]. Ensuite un neurone du réseau sera activé. Si la similarité entre le nouveau cas cible et le prototype associé au neurone activé est suffisante alors cette solution est retournée par le système (elle contient les futures pages Web à visiter ou la classe de l'utilisateur). Sinon, la phase de recherche dans le système CASEP2 est réalisée selon l'une des deux manières suivantes : 1. La recherche des cas sources similaires au cas cible est effectuée seulement dans la partie de la base de cas où le neurone activé est lié. Dans ce cas l'espace de recherche se limite à une partie de la base de cas. 2. La recherche est réalisée dans la zone atypique puisque le neurone activé n'est lié à aucune partie de la base. Cette partie de la mémoire (la zone atypique) consiste à stocker les nouveaux cas appris au cours de l'utilisation du système. Dans ce cas, la zone atypique est examinée séquentiellement pour retrouver des cas sources similaires au nouveau cas cible.



2.6.5.5 BROADWAY

BROADWAY [Jaczynski, 1998]	le système BROADWAY a été créé pour la prédiction du comportement d'un internaute à partir de pages web déjà visitées.
Représentation de cas	sous-séquences de la même longueur.
Remémoration / Mesure de similarité	recherche multidimensionnelles / mise en correspondance des chroniques des variables (une mesure de similarité par variable).
Description de l'approche	dans le système BROADWAY, un cas correspond à une sous-navigation, ou encore à une expérience précise dans une navigation. Le but du système est de proposer à l'utilisateur une ou plusieurs pages à visiter en s'appuyant sur une succession de pages déjà visualisées. Les deux parties problème et solution contiennent un ensemble de pages web jugées pertinentes durant la navigation. Pour la partie problème, elle regroupe l'ensemble des pages web visitées, des sous-navigations de longueur fixe, composé des p dernières pages visualisées par l'internaute. Cette partie problème est représentée par les variables pertinentes d'observations (adresse, contenu, évaluation, ratio d'affichage) pour un contexte donné. Le ratio entre le temps d'affichage et la taille d'une page correspond à la valeur de la variable ratio d'affichage. Le contexte d'une navigation comporte la description instantanée des paramètres pertinents de la navigation (les mots clefs, etc.). La partie solution du cas contient une ou plusieurs pages web recommandées pour l'utilisateur. Dans le système BROADWAY, les séquences décrivant les évolutions des valeurs des différents paramètres sont composées de différents types de valeurs (symboliques et numériques). Les auteurs de BROADWAY ont défini deux fonctions pour effectuer les comparaisons entre le cas cible et les cas sources : une fonction élémentaire de similarité pour chaque variable d'observation et une autre pour les contraintes temporelles. La seconde fonction permet de prendre en considération dans la comparaison des cas, l'ordre dans lequel les pages ont été visitées durant une navigation. La sélection du meilleur cas parmi les cas sources représentant des similarités suffisantes avec le cas cible se déroule selon les étapes suivantes :

	○ La sélection des cas sources (les anciennes navigations) représentant une similarité de contexte (mots clés utilisés pour la recherche, etc.) avec le nouveau cas cible (la nouvelle navigation). ○ Le calcul de la mesure des similarités entre les différents historiques, décrivant les variables d'observations, sélectionnés lors de la première étape. La sélection du meilleur cas sources parmi les cas sélectionnés lors des deux étapes précédentes présente des similarités de contraintes temporelles vis-à-vis du nouveau cas cible.
--	--

2.6.5.6 SAPED

SAPED [Aich, 2007]	le système SAPED (Système d'Assistance pour Personnes HanDicapées), a été créé pour l'assistance des utilisateurs d'un fauteuil roulant considérés lourdement handicapés. Cette assistance consiste à reproduire des trajets fréquemment employés pour limiter le nombre de commandes de direction à transmettre au fauteuil. L'architecture du système est donnée par la figure 2.14.
Représentation de cas	sous-séquence de longueurs différentes
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des historiques de valeurs (une mesure de similarité par variable).
Description de l'approche	dans SAPED, un cas représente une sous-séquence de longueur variable (un segment de trajet). Un trajet est défini par sa forme géométrique représentée par une séquence d'orientations et par l'environnement dans lequel il est exécuté (décrit par une séquence de rétrécissements). Un cas dans le système SAPED est composé de deux parties : la partie problème constituée des variables : - distance : représente la distance parcourue entre deux orientations et/ou rétrécissements successifs ; - rétrécissement : indique la représentation symbolique de l'espace libre observé devant le fauteuil ; - retour arrière : représente le comportement <i>retour arrière</i> du fauteuil. Elle permet de différencier la situation d'avancement du fauteuil de son recul ; - orientationCasPrécédent : cette variable décrit le changement d'orientation relativement à celle suivie jusqu'à l'instant du déclenchement du cycle RàPC. La deuxième partie d'un cas en système SAPED, qui représente sa partie solution, est composée des deux paramètres : <i>IdentificateurTrajet</i> et <i>PositionTrajet</i>. Ils permettent d'identifier les points de départ et d'arrivée du trajet. La partie solution permet d'identifier la prochaine orientation du fauteuil. Le système SAPED associe à chaque cas source des variables appelées "compétences d'un cas", indiquant la qualité de leurs utilisations dans le cycle de raisonnement (succès ou échecs). La mémoire en SAPED est constituée de deux parties : une première partie appelée base de cas dans laquelle les cas sont stockés avec leurs solutions et une autre partie appelée base de trajets qui contiennent des trajets complets. SAPED propose une structure de base de cas hiérarchique à trois niveaux : le premier niveau contient des prototypes qui regroupent les cas ayant approximativement les mêmes valeurs du paramètre distance, le deuxième niveau mémorise les parties "problèmes" des cas, et le troisième niveau stocke les parties solutions des cas. Le but du système consiste à prédire les futures orientations du fauteuil. Au début le fauteuil commence à se déplacer à partir d'un point quelconque, afin d'atteindre un lieu initialement non connu. Le système récupère les valeurs

d'orientation du fauteuil par rapport à la direction choisie et le rétrécissement indiquant l'espace libre devant le fauteuil, ceci après chaque 10 cm parcourus. L'étape d'élaboration dans SAPED, permet de redéfinir la représentation numérique des valeurs des variables de sa partie "problème", à une représentation symbolique.

Le fauteuil continu à se déplacer et à recueillir les données. Lors du changement d'orientation et/ou de rétrécissement et/ou l'activation du comportement *retour arrière*, la partie problème d'un nouveau cas cible est élaborée.

L'étape de recherche dans le système SAPED consiste à choisir un trajet, déjà effectué considéré similaire au nouveau trajet emprunté afin qu'il soit reproduit. Cette recherche s'effectue selon deux modes :

- ✓ Mode 1 : ce mode de recherche s'exécute s'il n'y a aucun trajet source en cours de reproduction. Dans ce cas le système procède à la sélection des cas sources similaires au nouveau cas cible élaboré. Par suite, les trajets sources contenant au moins un cas source parmi ceux extraits précédemment sont extraits de la base de trajets. Un de ces trajets sera sélectionné afin de le reproduire,
- ✓ Mode 2 : ce mode consiste à reproduire un des trajets sélectionnés lors du premier mode de remémoration. Le but est de mesurer de manière continue la similarité entre les nouveaux cas cibles du nouveau trajet emprunté et les cas sources composant le trajet en cours de reproduction.

La phase de recherche dans le système SAPED s'effectue selon chaque point de vue, ou encore suivant chaque variable. Au départ la recherche s'effectue selon la variable distance en utilisant la distance euclidienne comme mesure de similarité. La phase d'adaptation des solutions proposées par les cas sources est absente dans le système SAPED.

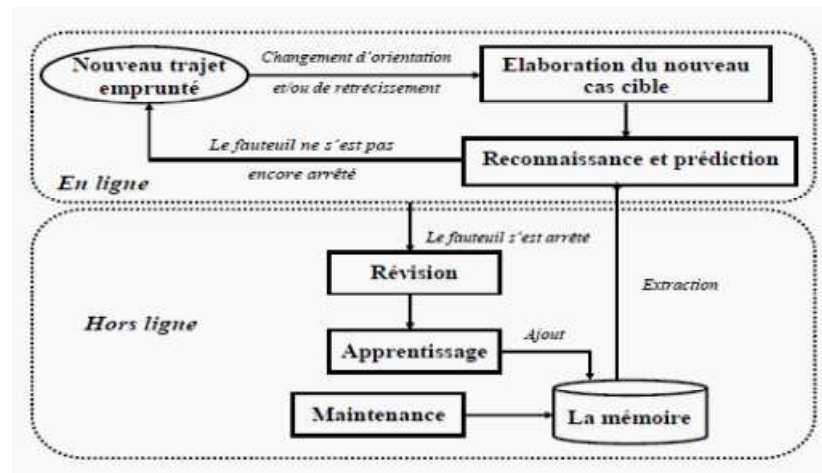


Figure 2.14 : L'architecture du système d'assistance SAPED [Aich, 2007]

Finalement nous signalons que le système SAPED considère chaque changement dans le trajet comme un nouveau cas cible.

2.6.6 Synthèse comparative

Dans cette section plusieurs systèmes traitant des situations dynamiques ont été présentés. Ils réutilisent les expériences passées pour prendre des décisions adaptées aux nouvelles situations rencontrées. Le tableau suivant (tableau 2.3) compare ces différents systèmes selon plusieurs critères. Les critères : représentation de cas,

remémoration, révision et l'apprentissage ont été proposés par Aich [Aich, 2007]. Nous ajoutons à cette étude comparative les critères suivants : structure de la base de cas, mesure de similarité, l'adaptation et révision.

	REBECAS	AURA	SINS	CASEP2	BROADWAY	SAPED
Représentation de cas	Séquence complète	Sous-séquences de longueurs différentes	Sous-séquences de la même longueur	Sous-séquence de longueurs différentes	Sous-séquences de la même longueur	Sous-séquence de longueurs différentes
Structure de la base de cas	Plate	Plate	Plate	Hierarchique	Hierarchique	Hierarchique
Remémoration	Recherche par point de vue	Recherche par point de vue	Recherche par point de vue	Recherche multidimensionnelles	Recherche multidimensionnelles	Recherche par point de vue
Mesure de similarité	Comparaison des historiques de valeurs (prog. dynamique)	Comparaison des historiques de valeurs (distance euclidienne)	Comparaison des historiques de valeurs (écarts quadratiques)	Comparaison des matrices de covariance dynamiques	Mise en correspondance des chroniques des variables (une mesure de similarité par variable)	Comparaison des historiques de valeurs (une mesure de similarité par variable)
Adaptation	Transformationnelle	Transformationnelle	Transformationnelle	Adaptation générative	Adaptation générative	Non
Révision	Non	Oui	non	Oui	Oui	Oui
Apprentissage	Ajout manuel de nouveaux cas	-Modification des cas existants -Ajout automatique nouveaux cas	- Modification des cas existants -Ajout automatique de nouveaux cas	-Ajout automatique de nouveaux cas -Apprentissage de la structure du réseau de neurones	-Ajout automatique de nouveaux cas - Modification des cas existants de neurones	-Ajout automatique de nouveaux cas - Réorganisation des cas existants

Tableau 2.3 : Comparaison entre des systèmes de RàPC traitant des situations dynamiques

2.7 Conclusion

Nous avons classé les systèmes RàPC en deux catégories d'applications : les applications statiques et dynamiques. Pour le cas statique, nous n'exploitons pas ce type de RàPC pour développer notre système. Nous justifions ce choix par le fait que dans l'approche orienté RàPC statique, un problème doit être complètement décrit avant que ne débute la recherche dans la base de cas. Or dans notre situation, le cas cible est un tracé qui évolue avec le temps, le raisonnement à partir de cas doit donc prendre en compte cette évolution de manière incrémentale. Il s'agit donc du raisonnement à partir de cas pour situation dynamique avec certaines particularités importantes.

Dans le cas des systèmes RàPC traitant de situations dynamiques. Ils concernent principalement des applications de prédiction. Ces systèmes se distinguent par une représentation temporelle des cas. Quand au cycle de raisonnement, les systèmes actuels RàPCD exploitent un cycle semblable à celui adopté par les situations statiques. Néanmoins ces systèmes considèrent chaque changement dans la situation traitée comme un nouveau cas cible, et par la suite ils exécutent le même cycle statique à nouveau.

L'objectif principal de notre thèse est d'apporter une solution à ce problème. Nous nous appuyons sur les systèmes multi-agents, paradigme d'intelligence artificielle, qui représente actuellement l'un des moyens mis à la disposition des chercheurs pour modéliser et construire les systèmes dynamiques distribués, pour proposer une nouvelle architecture du cycle de Raisonnement à Partir de Cas, capable de surmonter les limites de l'architecture actuelle. Dans le chapitre suivant nous présenterons le raisonnement à partir de cas dynamiques multi-agents.

Chapitre 3 : Raisonnement à Partir de Cas Dynamiques Multi-Agents

3.1 Introduction	87
3.2 Agents, systèmes multi-agents et méthodologies	87
3.2.1 L'intelligence artificielle distribuée	87
3.2.2 Agents.....	88
3.2.2.1 Définitions.....	88
3.2.2.2 Propriétés des agents	90
3.2.2.3 Typologie des agents	90
3.2.2.4 L'environnement des agents.....	93
3.2.3 Les systèmes multi-agents.....	94
3.2.3.1 Définitions.....	94
3.2.3.2 Raisonnement et communication dans les SMA	96
3.2.3.3 La communication dans les SMA.....	96
3.2.3.4 Modes de coopération.....	97
3.2.3.5 Organisation multi-agents	98
3.2.4 Méthodologies de développement des SMA	100
3.2.4.1 Agent-UML	100
3.2.4.2 AALAADIN et AGR.....	101
3.2.4.3 AMOMCASYS	102
3.2.4.4 MOISE+	102
3.2.4.5 Voyelles.....	102
3.2.4.6 GAIA	103
3.2.4.7 ROADMAP	103
3.2.4.8 MaSE.....	104
3.2.4.9 O-MaSE.....	104
3.2.4.10 PASSI	104
3.2.4.11 ADELFE.....	105
3.2.4.12 INGENIAS	105
3.2.5 Les plateformes de développement SMA.....	105
3.2.5.1 Zeus	105

Certaines parties de ce chapitre ont été publiés dans le journal international IJCSI [ZOUHAIR et al., 2012a] et le proceeding du IDC 2012 [ZOUHAIR et al., 2012b].

3.2.5.2 MADKIT	106
3.2.5.3 AgentBuilder	107
3.2.5.4 JADE	107
3.2.5.5 JACK	108
3.2.5.6 Étude comparative des plateformes agents.....	108
3.3 Systèmes RàPC fondés sur SMA	109
3.3.1 Classification des systèmes RàPC fondés sur SMA.....	109
3.3.1.1 Raisonnement à partir de cas orienté agent	110
3.3.1.2 Raisonnement à partir de cas orienté multi-agents	110
3.3.2 Principaux systèmes RàPC fondés sur les SMA	111
3.3.2.1 Système FPL.....	111
3.3.2.2 Système CICLMAN	111
3.3.2.3 Système CCBP.....	113
3.3.2.4 Système MCBP.....	115
3.3.2.5 Système CBR-TEAM.....	116
3.3.2.6 Système ProCLAIM.....	117
3.3.2.7 Système RoboCats.....	118
3.3.2.8 Système AMAL.....	120
3.3.2.9 Système POMAESS	121
3.3.2.10 Système S-MAS	123
3.3.3 Etude comparative des RàPC multi-agents	125
3.4 Conclusion.....	128

3.1 Introduction

Plusieurs systèmes RàPC ont été développés pour traiter les situations dynamiques, mais souffrent de limites dans la gestion des paramètres dynamiques, autrement dit des paramètres qui varient au cours du temps. Le système doit donc être doté d'un mécanisme pour détecter automatiquement l'évolution de ses paramètres et s'adapter aux changements de la situation. La complexité et la dynamique de la situation à traiter, a amené les chercheurs à explorer de nouvelles pistes pour développer les systèmes RàPC pour situations dynamiques, tels que les Systèmes Multi-Agents (SMA). Les SMA ont montré leur pertinence pour la conception d'applications dynamiques, et distribuées. Ils sont employés au sein des systèmes RàPC pour permettre à un groupe d'agents de partager leurs expériences. En effet l'expérience individuelle d'un agent peut être limitée, par conséquent leurs connaissances et prédiction aussi. Ainsi, les agents d'un système SMA sont capables de coopérer avec d'autres agents pour une meilleure prédiction de la situation.

Ce chapitre est organisé de la manière suivante. Dans la section 3.2, nous allons présenter le concept d'agent et leur typologie, ensuite nous décrivons les systèmes multi-agents. En deuxième partie, nous présentons les différentes méthodologies d'analyse et de conception des SMA ainsi que les plateformes liées à ces derniers. Nous terminons cette partie par une comparaison de ces méthodologies et plateformes SMA. La section 3.3 est consacrée aux systèmes RàPC fondés sur les SMA, nous commençons cette partie par une classification de ces systèmes, ensuite nous allons présenter les principaux systèmes RàPC basés sur les SMA et nous terminons cette section par une étude comparative de ces systèmes en section 3.3.3.

3.2 Agents, systèmes multi-agents et méthodologies

Les systèmes multi-agents permettent la modélisation et la conception des systèmes dynamiques, distribués et complexes. Ils représentent donc l'étude et la conception d'organisations d'agents artificiels pour obtenir des systèmes intelligents [Demazeau et Müller, 1991]. L'agent est une entité autonome qui à un objectif. Il évolue dans un environnement et il est capable d'interagir avec d'autres agents, éventuellement répartis sur le réseau, en se basant sur des langages et protocoles de communication.

3.2.1 L'intelligence artificielle distribuée

L'Intelligence Artificielle (IA) est apparue avec comme objectif initial de concevoir des machines intelligentes de capacités intellectuelles comparables à celles des êtres humains. L'intelligence artificielle a vu le jour début des années 50, probablement dans l'article d'Alan Turing *Computing Machinery and Intelligence* [Turing, 1950], où Turing propose une expérience connue sous le nom de *test de Turing* dans une démarche et expérience permettant de qualifier une machine de *consciente*.

En 1956, le terme d'intelligence artificielle est proposé, durant la conférence de Dartmouth, par Minsky et McCarthy et est depuis considéré comme un domaine de recherche.

Au milieu des années 70, l'Intelligence Artificielle Distribuée (IAD) est apparu, comme une branche à part entière de l'IA. En effet l'évaluation des domaines d'application de l'IA pour recouvrir des domaines complexes et hétérogènes tels que l'aide à la décision, la reconnaissance et la compréhension des formes a montré les limites de l'approche classique de l'IA qui se base sur une centralisation de l'expertise. L'IAD met donc l'accent sur l'aspect social et considère que l'interaction est la base de la constitution d'organisations capables de mener à bien des tâches complexes [Bond, 1988]. Par conséquent, l'intelligence est le fruit d'un ensemble d'individus en interaction.

On distingue principalement trois axes de distribution de l'intelligence artificielle [Labidi, 1993], [En-Naimi, 2013] :

- **L'Intelligence Artificielle Parallèle (IAP)**, qui vise la parallélisation des approches existantes en IA. Il concerne le développement des techniques et capable d'améliorer les performances des systèmes d'intelligence artificielle sans s'intéresser à la nature du raisonnement ou au comportement intelligent du système ;
- **La Résolution Distribuée de Problèmes (RDP)** concerne la décomposition de problèmes entre plusieurs modules spécialisés qui coopèrent en partageant leurs connaissances et concourent chacun à la résolution [Durfee, 1989] ;
- **Les Systèmes Multi-Agents (SMA)** [Ferber, 1995], [Wooldridge et Jennings, 1995], [Demazeau, 2001], s'intéressent à des comportements intelligents qui sont le produit de l'activité coopérative de plusieurs entités intelligentes nommées agents. En effet il s'agit de faire coopérer un ensemble d'agents dotés d'un comportement intelligent et de coordonner leurs buts et leurs plans d'actions pour la résolution d'un problème.

3.2.2 Agents

3.2.2.1 Définitions

Dans la littérature plusieurs définitions des agents ont été proposées, parmi les principales d'entre elles nous citons, celle de Ferber [Ferber, 1995], [Wooldridge et Jennings, 1995], [Demazeau et al., 1990] et [Florez, 1999].

Selon [Ferber, 1995], un agent est une entité physique ou virtuelle :

- a. qui est capable d'agir dans un environnement ;
- b. qui peut communiquer directement avec d'autres agents ;

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

Cette définition met l'accent sur plusieurs termes importants tels que : entité physique *versus* virtuelle, capacité, ou objectif. D'après Ferber une entité physique est quelque chose qui agit dans le monde réel : un robot ou un avion par exemple. Alors que l'entité virtuelle est un composant logiciel, un module informatique. Elle met l'accent aussi sur la capacité d'action d'un agent et sur son autonomie, puisqu'il dispose de tendances et d'objectifs. Ceci ne peut se produire que dans un environnement où l'agent est en interaction avec d'autres agents.

Le concept d'agent a été l'idée de base pour concevoir l'intelligence artificielle distribuée. L'agent est capable d'agir, et non pas seulement de raisonner comme dans les systèmes d'intelligence artificielle classique.

Jennings [Jennings et al., 1998] a défini l'agent comme " système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu ". Les notions situé, autonomie et flexible sont définies comme suit :

- Situé : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement, par exemple systèmes de contrôle de processus, ou systèmes embarqués ;
- Autonome : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne ;
- Flexible : par flexibilité, les auteurs entendent :
 - réactivité : un système réactif maintient un lien constant avec son environnement et répond aux changements qui y surviennent ;
 - pro-activité : un système proactif génère et satisfait des buts. Son comportement n'est donc pas uniquement dirigé par des événements ;
 - capacités sociales : un système social est capable d'interagir ou coopérer avec d'autres systèmes.

Selon Demazeau, [Demazeau et al., 1990], un agent est une entité intelligente, agissant rationnellement et intentionnellement, en fonction de ses buts propres et de l'état actuel de ses connaissances.

Enfin, selon Florez, [Florez, 1999] l'agent peut être défini comme une entité interactive qui existe en tant qu'une partie d'un environnement partagé par d'autres agents. C'est une entité conceptuelle qui perçoit et agit avec initiative ou en réaction dans un environnement où d'autres agents existent et interagissent les uns avec les autres, sur la base de connaissances partagées de communication et de présentation.

3.2.2.2 Propriétés des agents

Parmi les principales propriétés et caractéristiques des agents, nous pouvons citer [Ferber, 1995], [Demazeau et al., 1990], [Jennings et al., 1998], [Wooldridge et Jennings, 1995], [Cardelli, 1999], [Nfaoui, 2008] :

- **L'apprentissage et l'adaptabilité** : l'agent peut être capable d'apprendre à partir de ses expériences et s'adapter à des changements dans des environnements dynamiques.
- **Le raisonnement** : un agent peut décider quel objectif et but poursuivre ou à quel événement il doit réagir, comment agir pour accomplir son but, et enfin comment abandonner un but pour se consacrer à un autre.
- **La mobilité**: dans certaines applications, les agents doivent avoir la capacité de migrer d'un nœud à un autre dans un réseau tout en protégeant leur état.
- **La sociabilité** : les agents interagissent entre eux et échangent les informations grâce à des langages de communication. L'interaction peut être dans un but coopératif ou bien compétitif. Parmi les comportements sociaux des agents nous pouvons citer la :
 - **communication** : l'agent est capable de s'engager dans des communications complexes avec des agents, des utilisateurs, des applications, ou des ressources distribuées.
 - **coopération** : l'agent est capable d'établir des liens avec les autres agents dans un cadre coopératif afin d'accomplir ses tâches.
 - **négociation** : l'agent est capable de négocier avec d'autres agents pour prendre une décision.

Dans le cadre de cette thèse, nous nous intéressons à l'adaptation de la plupart de ces propriétés fondamentales à savoir : l'autonomie, l'apprentissage et l'adaptabilité, la réactivité, la pro-activité, le raisonnement et le comportement social.

3.2.2.3 Typologie des agents

Il existe plusieurs typologies d'agents, groupées dans des classes ou famille en se basant sur leurs propriétés et définitions. Nous présentons dans ce qui suit, les principaux modèles d'agents les plus utilisés. Dans cette description nous retrouvons

des caractéristiques communes aux agents de notre système, et nous présentons ces classes selon les auteurs [Ferber, 1995], [Bratman et al., 1988], [Nwana, 1996], [Hsairi, 2010].

Ferber [Ferber, 1995], distingue deux grandes catégories d'agents : réactifs et cognitifs :

- **Les agents réactifs** ont un comportement de type stimuli/réponse. Dès que ces agents perçoivent une modification de leur environnement, ils répondent par une action programmée. Ils ne possèdent pas une représentation complète de leurs environnements et ils n'ont pas ou peu de mémoire et donc ils n'ont pas de capacité d'apprentissage. Souvent ce type d'agents est considéré comme n'étant pas intelligent. Chaque agent pris indépendamment possède une représentation partielle et faible de l'environnement et il n'a pas de but. Toutefois ils sont collectivement capables d'exécuter des actions évoluées et coordonnées [Ferber, 1995].
- **Les agents cognitifs** possèdent un modèle de raisonnement qui provient d'une métaphore du modèle humain. Ils possèdent une représentation explicite de leur environnement, des autres agents et d'eux-mêmes. Ils peuvent planifier, coordonner leurs comportements, mémoriser des situations passées pour les analyser et prévoir les évolutions de ses actions. Nous citons ici le modèle le plus représentatif de cette famille d'agent : BDI (Believe – Desire - Intention) [Bratman et al., 1988].

Agents réactifs	Agents cognitifs
Pas de représentation explicite	Représentation explicite de l'environnement
Pas de mémoire de son historique	Peut tenir compte de son passé
Fonctionnement stimulus /réponse	Agent complexe
Grand nombre d'agents	Petit nombre d'agents

Tableau 3.1 Agent cognitif vs Agent réactif [Ferber, 1995]

Architecture BDI

BDI (Belief, Desire, Intention) est l'une des classes importantes des modèles cognitifs. Ce modèle repose sur trois aspects principaux, qui définissent la rationalité d'un agent intelligent [Bratman et al., 1988], [Turing, 2002] :

B = Belief = Croyance : les croyances d'un agent sont les informations que possède l'agent sur son l'environnement et sur les autres agents qui partagent le même environnement. Ces informations (croyances) peuvent être incorrectes, incertaines ou

tout simplement incomplètes. Toutefois les connaissances de l'agent sur lui même, constitue les connaissances supposées vraies de l'agent.

D = Desire = Désir : les désirs d'un agent sont les états de l'environnement de l'agent lui-même, qu'il aimerait voir réaliser. Autrement dit ce sont les objectifs que se fixe un agent. Dans le cas où l'agent a des désirs contradictoires, il doit choisir un sous-ensemble de désirs à accomplir.

I = Intention = Intention : les intentions d'un agent sont les actions qu'un agent a décidé de réaliser afin d'accomplir ses désirs. Ils sont sous la forme de plans à exécuter tant que l'objectif correspondant n'est pas atteint.

Nwana, [Nwana, 1996] propose une typologie des agents générale et complète :

Mobilité : les agents peuvent être classés selon leur mobilité, nous distinguons deux sous classes d'agents statiques ou mobiles. L'agent mobile se déplace d'une machine à une autre dans le contexte des réseaux distribués afin de coopérer avec les autres agents. Pour l'agent statique, il reste où il a été implémenté dès le début et durant toutes les phases de son cycle de vie.

Délibératifs ou réactifs : ces agents possèdent un modèle de raisonnement symbolique interne, capables de planifier et de négocier afin d'accomplir une coordination avec d'autres agents. Les agents réactifs n'ont aucun modèle symbolique interne et ils ne possèdent pas une représentation complète de leurs environnements. Ces agents agissent en utilisant un comportement de type stimulus/réponse suite au changement de l'état courant de l'environnement dans lequel ils cohabitent.

Rôle : les agents peuvent être classés selon leurs rôles. Comme par exemple la recherche et la gestion d'informations sur Internet.

Autonomie : les agents ont des états internes et des buts qu'ils cherchent à atteindre.

Coopération : les agents ont besoin d'une capacité sociale, c'est-à-dire la capacité d'interagir avec d'autres agents et éventuellement avec les humains.

Apprentissage : les agents intelligents peuvent être capables d'apprendre comment réagir ou interagir avec leur environnement.

Polyvalent ou pluridisciplinaire : l'agent peut être classé en tenant compte de sa capacité d'avoir plusieurs objectifs.

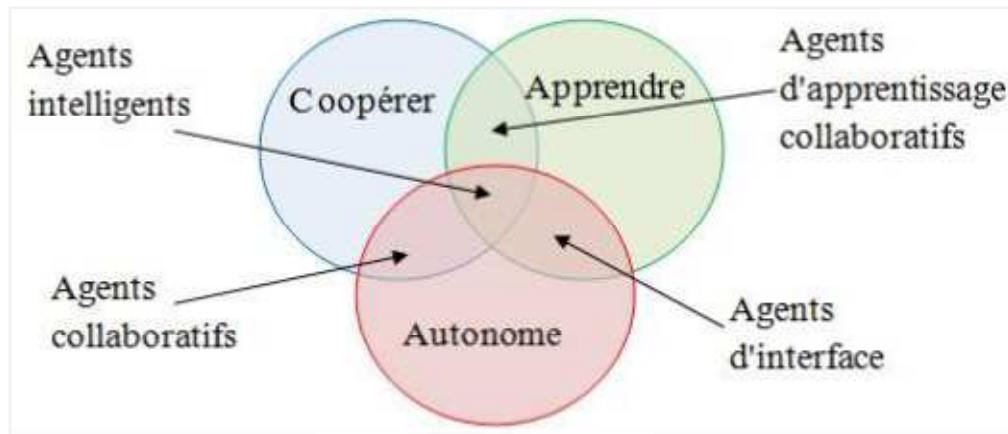


Figure 3.1 : Typologie des Agents selon [Nwana, 1996]

Hsairi, [Hsairi, 2010], distinguent trois types d'agents :

Les agents collaboratifs sont des agents autonomes et réactifs. Ils ont des capacités d'initier des actions de sociabilité et de coopérer avec les autres agents pour réaliser leurs objectifs. Ils doivent pouvoir négocier afin d'arriver à des accords acceptables. Les agents collaboratifs peuvent apprendre mais l'apprentissage n'est pas une caractéristique fondamentale requise pour ce type d'agents.

Les agents d'interface sont des agents autonomes capables d'apprendre. Cette classe d'agents est caractérisée par la métaphore d'assistant personnel collaborant avec l'utilisateur dans le même environnement [Mesa, 1997].

Les agents intelligents : selon l'association française de normalisation un agent intelligent est un " objet utilisant les techniques de l'intelligence artificielle : il adapte son comportement à son environnement et en mémorisant ses expériences, se comporte comme un sous-système capable d'apprentissage : il enrichit le système qui l'utilise en ajoutant, au cours du temps, des fonctions automatiques de traitement, de contrôle, de mémorisation ou de transfert d'informations " [AFNOR, 1987].

3.2.2.4 L'environnement des agents

L'environnement d'un agent représente tout ce qui est extérieur à l'agent. Il existe plusieurs classes d'environnement : social, physique, déterministe, ou continu :

Environnement social versus physique : l'environnement social d'un agent correspond aux autres agents qu'il connaît, et avec lesquels l'agent est en interaction par échange de messages. L'environnement physique est constitué des ressources matérielles existant dans le champ de perception de l'agent ou des agents du système. Autrement dit ce sont les ressources que l'agent peut percevoir ou sur lesquels il peut agir. Par exemple, la nourriture et les obstacles représente l'environnement physique d'une colonie de fourmis.

Environnement accessible : c'est celui dans lequel un agent peut obtenir une information complète, exacte, dans les délais et à jour sur l'état de son environnement. Dans un environnement inaccessible, les informations disponibles sont partielles. Par

exemple, l'Internet est un environnement inaccessible car il est impossible de tout connaître.

Environnement continu versus discret : un environnement est considéré continu si le nombre d'actions et de perceptions possibles est infini. Dans le cas contraire, l'environnement est dit discret. Par exemple, dans un environnement simulé tel qu'un éco-système, le nombre d'actions ou de perceptions qu'une entité virtuelle (fourmi par exemple) peut avoir est limité ; l'environnement est alors discret. Sur Internet, le nombre d'actions qui peuvent être effectuées par les utilisateurs est illimité.

Environnement déterministe : dans un environnement déterministe, une action a un effet unique et certain. Dans ce cas, l'état suivant de l'environnement est complètement déterminé par l'état courant. Dans un environnement non déterministe, l'effet d'une action n'est pas toujours garanti. Par nature, le monde physique réel est un environnement non déterministe.

Environnement dynamique vs statique : l'état d'un environnement dynamique dépend des actions des processus appartenant à cet environnement (processus internes), et il dépend aussi des actions d'autres processus externes. Aussi, les changements de l'environnement ne peuvent pas être prédits par le système.

3.2.3 Les systèmes multi-agents

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

3.2.3.1 Définitions

Ferber [Ferber, 1995] définit un système multi-agents (SMA), comme étant composé des éléments suivants (figure 3.2) :

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

Dans le cas où $O = A$ et E est l'ensemble vide, les relations R définissent un réseau: chaque agent est lié directement à un ensemble d'autres agents (ses accointances), et on appelle ce type de système SMA purement communicant. C'est un SMA ne disposant pas d'environnement et dans lequel les agents ne font que communiquer entre eux.

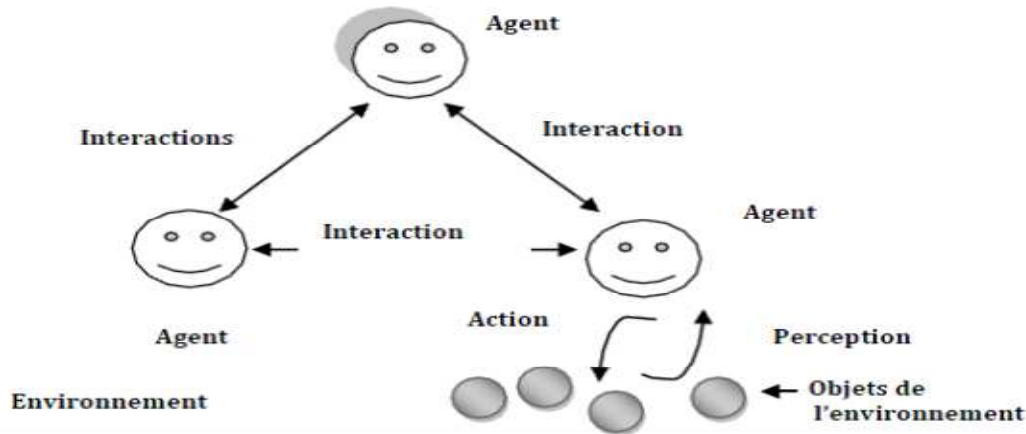


Figure 3.2 : Représentation d'un SMA [Ferber, 1995]

Pour Danflous un SMA est défini comme étant « *un ensemble d'entités qui coordonnent leurs connaissances, buts, expériences et plans pour agir ou résoudre des problèmes, incluant le problème de coordination inter agents lui-même* » [Danflous, 2000].

Au sein d'un SMA, les agents ont une visibilité partielle sur leur environnement, et par conséquent, ils ne peuvent avoir qu'un champ d'actions limité sur l'ensemble des objets de cet environnement [Abed et al., 2013]. De ce fait et afin de résoudre un problème, les agents sont amenés à coopérer, à négocier et à communiquer pour échanger des informations et pour mieux coordonner leurs actions individuelles et locales. Ceci oblige à définir les rôles des agents et les mécanismes de coordination et de communication entre eux. Un système multi-agents peut-être [Drogoul, 2003] [Sabouret, 2009] :

- Ouvert : les agents y entrent et en sortent librement. Ils peuvent apparaître ou disparaître du système, par exemple dans un réseau mobile ad-hoc certains agents deviennent inaccessibles ;
- Fermé : le nombre des agents reste toujours le même ;
- Homogène : tous les agents sont construits sur le même modèle, et implémentés de la même façon ;
- Hétérogène : les agents ne sont pas construits sur le même modèle sont implémentés différemment et manipulent des modèles de données différents ;
- Faible couplage : un agent ne connaît pas a priori les capacités des autres agents. Dans ce cas l'agent doit les découvrir et interpréter leurs effets sur le système.

3.2.3.2 Raisonnement et communication dans les SMA

Le raisonnement [Mbala, 2003] dépend des compétences des agents, du type de communication et du mode de coopération :

- **La distribution des tâches et des compétences** comprend l'affectation d'une compétence ou d'une tâche à un seul agent, ou bien le partage de cette dernière entre deux ou plusieurs agents. Dans ce second cas les agents sont alors en concurrence, ce qui impose l'ajout d'un mécanisme de résolution de conflit à ce niveau ;
- **Le type de protocole de communication** qui peut être :
 - **une mémoire commune ou tableau noir** : dans ce cas les échanges d'informations et de solutions partielles se font au travers de cette mémoire commune ou tableau noir. Toutes les informations sont écrites sur le tableau noir et chaque agent peut les consulter (le destinataire concerné n'est pas choisi par l'émetteur du message).
 - **un échange de messages pair-à-pair** : les communications dans ce cas s'effectuent par envoi de messages d'un agent à un autre. Dans ce cas le choix du destinataire du message doit être fait de manière pertinente ce qui complexifie le raisonnement.
- **Le mode de coopération** dépend des compétences de chaque agent.

3.2.3.3 La communication dans les SMA

La communication des agents permet l'échange des informations, la négociation, la coopération et la coordination [Azaiez, 2007]. Il existe deux modes de communication entre les agents :

- Indirecte par partage de ressources, dans ce cas, les différents composants ne sont pas directement liés entre eux. La communication se fait à travers l'environnement ou bien par le biais d'un tableau noir.
- Directe par envoi de messages, dans ce cas, les connaissances du système sont distribuées entre les différents agents. Elle est intentionnelle et les agents communiquent entre eux par envoi de messages à un ou plusieurs destinataires. La communication directe, se base sur trois éléments principaux :
 1. Le langage de communication permet de structurer les messages échangés entre les agents. Les langages de communication les plus utilisés et les plus connus sont KQML (Knowledge Query and Manipulation Language) [Finin et al., 1995] et ACL (Agent Communication Language) [FIPA, 1999].
 2. L'ontologie permet de fournir une terminologie compréhensible et un vocabulaire pour tous les agents. Elle sera régie par des règles et des contraintes qui permettront de définir un consensus sur le sens des termes.
 3. Les mécanismes de communication permettent de stocker, rechercher et adresser les messages aux agents. Ils sont présents dans les plateformes multi-agents [Azaiez, 2007].

3.2.3.4 Modes de coopération

La coopération entre agents permet la résolution globale d'un problème. Les agents travaillent ensemble pour satisfaire un but commun ou individuel. Ferber [Ferber, 1995] définit la coopération entre les agents d'un SMA comme une situation dans laquelle, soit l'ajout d'un nouvel agent permet d'accroître les performances du groupe, soit l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels.

Selon Ferber [Ferber, 1995] il existe six modes de coopération entre les agents :

- **Le regroupement et la multiplication** : le principe de base du regroupement est issu de l'observation du monde animal. En effet il permet de former des blocs homogènes, contenant des spécialistes de tous les domaines, et une communication entre les spécialistes. La multiplication permet de prévenir les faiblesses individuelles possibles. Si par exemple le groupe des agents ne dispose que d'un spécialiste pour un domaine particulier, et que celui-ci venait à disparaître, alors la survie du groupe serait en péril.
- **La communication** : est indispensable à la coopération. Elle permet aux agents, via l'envoi de messages structurés, d'échanger des informations, de déléguer des tâches ou de négocier. Elle permet aussi à l'agent d'élargir leur perception de l'environnement, puisqu'elle permet aux agents d'exploiter celles des autres agents ainsi que leurs connaissances.
- **La spécialisation** : ce mécanisme permet d'avoir des agents spécialisés dans une tâche.
- **La collaboration** par partage de tâches et de ressources consiste à répondre à la question *qui fait quoi ?* Ceci peut être réalisé par la répartition des tâches via des mécanismes d'offre et de demande. Ferber distingue deux mécanismes :
 - un agent coordinateur centralise les offres et les demandes et les répartit ensuite au mieux, par appel d'offre (*Contract net protocol*).
 - les agents peuvent directement offrir ou demander des services aux autres agents et donc tout agent peut être à la fois offrant ou demandeur (pas d'organe centralisateur).
- **La coordination** d'actions permet d'éviter des problèmes dans le comportement global du système. Cela nécessite d'effectuer des actions supplémentaires comme de déterminer l'ordre d'exécution des tâches et de synchroniser leurs activités.
- **La résolution de conflits** par arbitrage et négociation : les agents peuvent entrer en conflit pour plusieurs raisons : conflits de buts ou d'intérêts, accès à des ressources ou proposition de plusieurs solutions au même problème. Dans le cas d'arbitrage on procède à la rédaction de lois et règles de comportement que les agents devront respecter. La résolution d'un conflit entre deux ou plusieurs agents peut aussi être accomplie par des négociations directes entre les agents [Mbala, 2003].

3.2.3.5 Organisation multi-agents

L'organisation d'un système multi-agents permet de définir l'architecture globale du système, et de décrire les interactions et les relations entre les agents. Nous pouvons distinguer, deux principales visions du concept organisation [Coutinho et al., 2005], [Azaiez, 2007] :

1. Organisation pré-établie : dans ce cas la structure organisationnelle du SMA ainsi que le comportement global sont pré-établis. plusieurs contraintes peuvent être exprimées au niveau de la topologie, des protocoles d'interaction, des normes utilisées ainsi que le déroulement des activités [Vazquez-Salceda et al., 2004].
2. Organisation émergente : dans ce cas l'organisation émerge du comportement global des différentes composantes du système [Georgé et al., 2005]. Pour ce type de structure, les éléments faisant partie de l'organisation sont préalablement connus et leur comportement individuel est défini. Néanmoins, le comportement global est inconnu.

Les topologies permettent de décrire la structure organisationnelle du système multi-agents, et permettent d'imposer certaines contraintes au niveau des interactions entre les agents. Strugeon [Strugeon, 1995] identifie quatre types de topologies [Azaiez, 2007] :

1. Les topologies à structure hiérarchique : les systèmes qui adoptent ce type d'organisations sont généralement rigides et se caractérisent par l'existence d'un agent central qui communique les ordres aux autres agents afin de les exécuter.

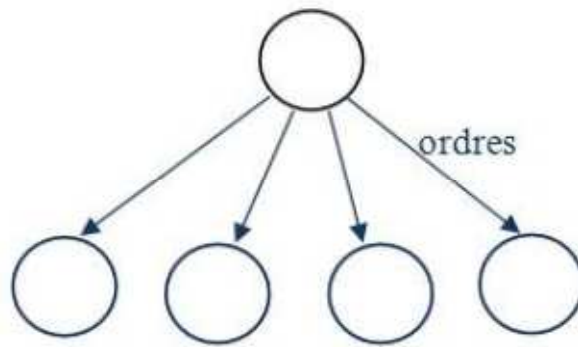


Figure 3.3 : Topologie à structure hiérarchique [Strugeon, 1995]

2. Les topologies de type marché : les systèmes qui adoptent ce type de topologie sont composés d'agents coordinateurs et d'agents exécutants. Cette structure est décentralisée autour de plusieurs objectifs opérationnels.

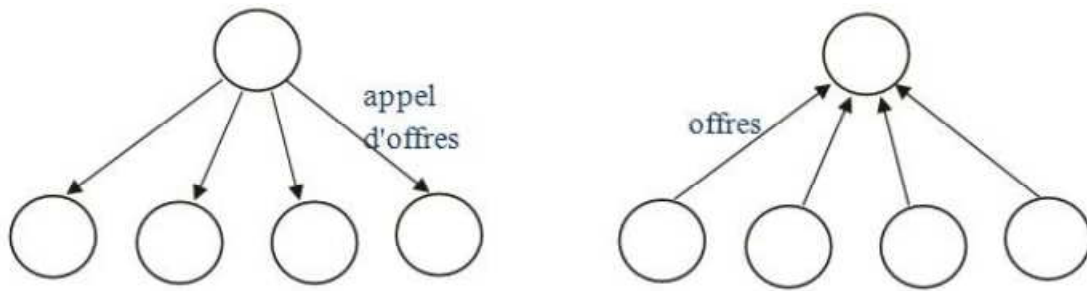


Figure 3.4 : Topologie à structure de marché [Strugeon, 1995]

3. Les topologies de type communauté : le contrôle est fortement distribué, et les éléments possèdent les mêmes capacités.

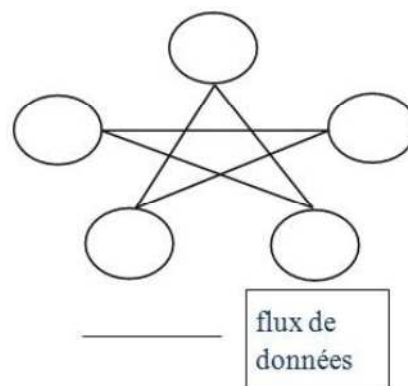


Figure 3.5 : Topologie à structure de communauté [Strugeon, 1995]

4. Les topologies de type société : le contrôle est décentralisé. Chaque agent cherche à accomplir son objectif opérationnel et le comportement global du système est ajusté par des principes de négociation.

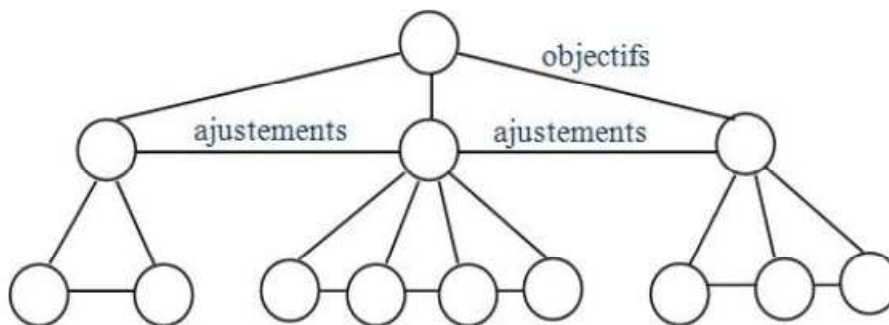


Figure 3.6 : Topologie à structure de société [Strugeon, 1995]

Un SMA est un groupe d'agents en interaction. De ces interactions peut émerger un comportement prévu ou un comportement émergent inattendu. Les comportements qui apparaissent peuvent être classés en trois catégories [Marcenac et Calderoni, 1997], [Azaiez, 2007] : l'émergence structurelle, l'émergence de comportements et l'émergence de propriétés. Selon Müller [Müller, 2002], un phénomène est émergent si :

- il y a un ensemble d'agents en interaction entre eux au sein d'un environnement ;
- la dynamique des agents en interaction produit un comportement global qui peut être une structure stable, une trace de production ou n'importe quel invariant statique ou dynamique ;
- le comportement global est observé soit par un observateur extérieur (sens faible) soit par les agents eux-mêmes (sens fort).

Les travaux sur l'émergence tentent de développer des outils, des méthodes et des mécanismes afin de maîtriser le processus d'émergence et par conséquent de comprendre le comportement émergent [Azaiez, 2007]. Ces travaux se basent, entre autre, sur :

- la théorie des systèmes complexes adaptatifs : l'émergence fait référence aux patterns de la macro niveau apparaissant dans les systèmes d'agents en interaction ;
- la théorie des systèmes dynamiques non linéaires issue de la théorie du chaos ;
- l'école synergetique qui étudie l'auto-organisation dans les systèmes physiques ;
- la thermodynamique.

3.2.4 Méthodologies de développement des SMA

Actuellement, il existe plusieurs méthodologies et outils de conception de SMA. La plupart de ces propositions méthodologiques sont inspirées des résultats du domaine du génie logiciel orienté objets, et il n'y a pas encore une méthode standard pour le développement des systèmes multi-agents. Dans cette section nous présentons plusieurs méthodes et nous justifions notre choix d'utiliser A-UML.

3.2.4.1 Agent-UML

AUML (Agent based Unified Modeling Language) est une méthodologie de développement des systèmes multi-agents proposée par Bauer, Odell et al [Bauer et al, 2001]. C'est une extension d'UML réservé à l'analyse et la conception orientée agent. Il vise à proposer un ensemble de notations mieux adaptées au paradigme multi-agents, en étendant le langage UML. Elle supporte tous les diagrammes des phases d'analyse et de conception avec une modification de ces diagrammes afin de représenter l'interaction entre les agents et l'état interne d'un agent.

Dans AUML :

- les objets sont remplacés par les agents et les rôles ;
- des notations ont été ajoutées pour représenter les interactions complexes, telles que des nouveaux types de branchements dans les diagrammes de séquence afin de prendre en compte l'indéterminisme du comportement d'un agent. Ces branchements, voir figure 3.7, peuvent être AND, Or et XOR ;
- la représentation de l'interaction entre les agents et l'état interne d'agent ;

- l'extension de contenu du message échangé entre agents ;
- la spécification des rôles dans les différents diagrammes, où les classes du diagramme de séquences génériques sont annotées Classe/Rôle.

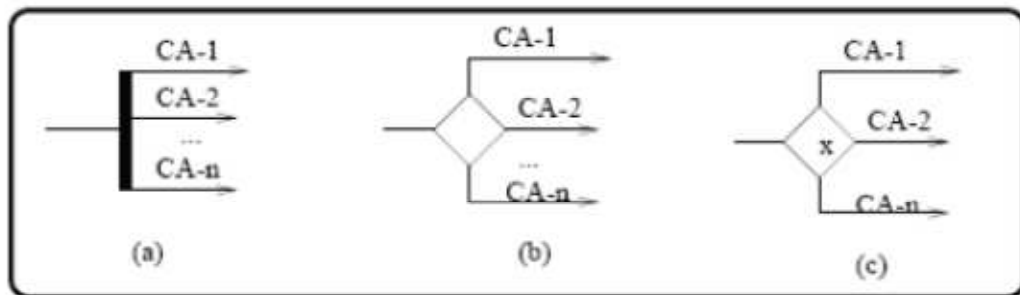


Figure 3.7 : Les branchements définis dans AUML

AUML est en accord avec les spécifications existantes de FIPA et de l'OMG [Bauer et al, 2001].

Il existe d'autres extensions du langage UML, M-UML [Saleh et El-Morr, 2004] pour modéliser les agents mobiles et AORML (Agent-Object Relationship Modeling Language) [Wagner et Tulba, 2003], pour spécialiser UML à un domaine particulier.

3.2.4.2 AALAADIN et AGR

AALAADIN est un méta-modèle proposé par Ferber et Gutknecht [Ferber et Gutknecht, 1998]. Il se base sur les concepts d'agents, de groupes et de rôles. Le modèle AGR (Agent, Groupe, Rôle) [Ferber et al., 2004] est une évolution de la méthodologie AALAADIN [Ferber et Gutknecht, 1998] (Figure 3.8) :



Figure 3.8 : Organisation modèle AGR

- Agent : les agents dans AALAADIN et AGR sont des entités autonomes actives et communicantes pouvant jouer un ou plusieurs rôles au sein des différents groupes. AALAADIN et AGR n'imposent aucune contrainte sur l'architecture interne des agents, ainsi aucun modèle particulier prédéfini pour décrire le comportement des agents.
- Groupe : c'est une structure dynamique, qui permet de représenter les différents niveaux d'organisation dans lequel un ensemble d'agents interagit. Par exemple

deux agents ne peuvent communiquer que s'ils appartiennent à un même groupe.

- Rôle : c'est la représentation abstraite du fonctionnement d'un agent au sein d'un groupe. Un agent peut avoir plusieurs rôles et un même rôle peut être joué par plusieurs agents. Enfin le rôle d'un agent est local à un groupe et il n'est reconnu qu'au sein du groupe.

AGR est un modèle générique permettant de s'adapter à différents types de systèmes multi-agents. Toutefois sa structure ne permettant pas de représenter l'ensemble des aspects d'un système multi-agents tels que l'environnement du système, ou les interactions entre les agents [Lebrun, 2012].

La méthode AALAADIN a donné lieu au développement de la plateforme MadKit [Gutknecht et Ferber, 2000].

3.2.4.3 AMOMCASYS

AMOMCASYS (Adaptable MOdeling Method for Complex Administrative SYStems) [Emmanuel et al., 2001, 2003], est une méthodologie pour l'analyse et la modélisation des systèmes et administratifs. Elle est constituée de quatre modèles principaux : un modèle de données, un modèle d'activités, un modèle de traitements et un modèle dynamique. Les quatre modèles permettent de définir et de décrire les différents rôles en fonction de leurs objectifs, de leurs organisations, ou de leurs interactions. [Lebrun, 2012]. AMOMCASYS permet de définir les règles (les pré-requis et les post-conditions), qui composent les rôles. Les règles permettent aussi d'ordonner l'exécution des rôles selon leur degré de priorité. Les scénarios sont définis en fonction du contexte ainsi que les stratégies des rôles. Toutefois, cette méthodologie ne décrit pas par précision la dynamique de l'environnement.

3.2.4.4 MOISE+

MOISE+ (Model of Organization for multi-agent SystEms) [Hübner et al., 2002], [Boissier et al., 2011], basée sur trois concepts : modèles de l'organisation (les modèles structurels), modèles traitant des aspects de structuration du système (rôles) et les modèles fonctionnels (relations).

3.2.4.5 Voyelles

Voyelles est une approche de haut niveau pour l'analyse, la conception et l'implémentation d'un système multi-agents. Elle a été proposée par Yves Demazeau [Demazeau, 2001], et elle repose sur la décomposition de la vue d'un système multi-agents suivant cinq dimensions ou modèles : Agent, Environnement, Interaction, Organisation et Utilisateur (A+E+I+O+U). Un système multi-agents selon la méthodologie Voyelles est un ensemble d'agents (A) qui évolue dans un

environnement (E), soumis à une organisation (O), en interaction entre eux, avec leurs rôles (I) et qui est centré sur l'utilisateur (U).

Voyelles permet aux concepteurs d'utiliser des formalismes, des langages ou des notations de leur choix pour spécifier chaque lettre du système (A+E+I+O+U). Toutefois elle propose des exemples permettant de les guider. Par exemple pour modéliser les agents, le concepteur peut utiliser des logiques formelles, des arbres, des automates, ou des graphes. Pour la modélisation de l'environnement on peut aussi utiliser des formalismes très différents tels que l'expression de différentes configurations d'arbres. Enfin pour les interactions entre les agents les protocoles exprimés par des diagrammes de séquences ou des machines à états sont souvent utilisés pour représenter ces interactions [Bernon et al., 2009].

La méthode Voyelles a donné lieu au développement de la plateforme MASK (*Multi-agent system kernel*) [Bernon et al., 2009].

3.2.4.6 GAIA

GAIA est une méthodologie où le système multi-agents est vu comme une organisation composée de rôles interagissant entre eux [Wooldridge et al., 2000]. Elle est créée pour l'analyse et la conception de tous types de systèmes multi-agents et met en avant deux niveaux, un macro-niveau qui se concentre sur la modélisation d'un groupe d'agents et un micro-niveau qui se focalise sur l'agent. GAIA considère un système multi-agents comme une structure composée d'une ou plusieurs organisations.

Actuellement, il n'existe pas encore un outil propre à GAIA, capable d'implémenter la méthodologie GAIA. De plus les ressources disponibles sur GAIA sont limitées, et elle exige un haut niveau de maîtrise de la logique temporelle. Ceci la rend difficile à adopter pour les développeurs [Bernon et al., 2009].

3.2.4.7 ROADMAP

ROADMAP (Role Oriented Analysis and Design for Multi-Agent Programming), est une méthodologie pour l'analyse et la conception orientée rôle pour la programmation multi-agents [Juan et al., 2003]. ROADMAP est une évolution de la méthodologie GAIA. Elle permet de développer des applications dont lesquelles les structures et le comportement d'agents évoluent d'une manière dynamique.

La méthode ROADMAP a donné lieu au développement de l'outil REBEL (Roadmap Editor Built for Easy Development), capable d'aider les développeurs à identifier les modèles de buts et de rôles. Comme pour GAIA, ROADMAP nécessite la maîtrise de la logique temporelle [Juan et al., 2003].

3.2.4.8 MaSE

MaSE (Multi-Agent System Engineering) modélise les SMA en termes d'objectifs, de rôles, d'agents, de tâches et d'interactions [Deloach, 1999]. Elle considère un agent comme un objet particulier, doté d'intelligence ou non, avec une capacité de coordination à travers la conversation avec les autres agents. La méthodologie MaSE se compose d'une phase d'analyse et d'une phase de conception.

L'analyse se concentre sur la production des rôles que le système doit remplir afin d'atteindre ses buts. Elle contient les étapes suivantes :

1. Identifier les rôles ;
2. Identifier les cas d'utilisations et créer les diagrammes de séquences ;
3. Transformer les buts en ensembles de rôles : créer un modèle de rôle puis définir un modèle de tâches concurrentes.

La phase de conception met l'accent sur :

1. La création de classes d'agents et les interactions ;
2. La création des classes internes d'agents ;
3. La définition de la structure finale du système (diagrammes de déploiement).

L'outil du génie logiciel AgentTool [AgentTool, 2013] permet de développer les SMA en se basant sur MaSE. Il fournit également un soutien automatisé pour la transformation des modèles d'analyse en agents. Par contre la méthodologie MaSE n'est pas en mesure de gérer la mobilité.

3.2.4.9 O-MaSE

O-MaSE (Organization-based Multi-Agent System Engineering) est une extension de MaSE. O-MaSE considère un système multi-agents comme une organisation sociale des agents [Deloach, 2005].

3.2.4.10 PASSI

PASSI (*Process for Agent Societies Specification and Implementation*) adopte des modèles de conception et concepts provenant de l'ingénierie orientée objet et de l'intelligence artificielle en utilisant UML [Cossentino et al., 2005]. PASSI est une méthodologie générique qui peut être appliquée à n'importe quel domaine. Elle a la particularité de prendre en compte la modélisation des agents mobiles, tout en respectant les spécifications FIPA. La réactivité et la pro-activité des agents sont définies dans un ensemble de modèles comportementaux. Les aspects sociaux sont intégrés dans le modèle de société d'agents.

PASSI fournit cinq modèles qui permettent la description des différents concepts déterminés au niveau du méta-modèle : modèle de collecte des besoins, modèle de société d'agent, modèle d'implémentation, modèle de code et modèle de déploiement [Cossentino et Potts, 2001].

La modélisation avec PASSI est maintenue par l'outil PTK (PASSI Toolkit) [Chella et al., 2004].

3.2.4.11 ADELEFE

ADELEFE (Atelier de Développement de Logiciel à Fonctionnalité Emergente) permet de guider les développeurs lors de systèmes fortement dynamiques en s'appuyant sur les théories adaptatives des SMA [Bernon et al., 2005]. ADELEFE propose un processus, des notations et des outils en se basant sur la notation UML. Ces systèmes sont basés sur des agents coopératifs qui exhibent des comportements sociaux : composés par un ou plusieurs agents coopératifs qui observent des règles de coopération et qui ont une représentation du monde. Les aspects dynamiques peuvent être modélisés grâce à des machines à états finis et des diagrammes d'interaction.

3.2.4.12 INGENIAS

La méthodologie INGENIAS [Pavon et Gomez, 2003], est une approche générale et flexible pour le développement des systèmes multi-agents. Elle repose sur la division du problème en plusieurs aspects qui forment les différentes vues du système. INGENIAS définit des méta-modèles capables de décrire les différents aspects d'un système multi-agents, ainsi que leurs relations.

La méthodologie INGENIAS met à la disposition des chercheurs un environnement de développement nommé IDK (INGENIAS Development Kit) [Azaiez, 2007].

3.2.5 Les plateformes de développement SMA

Parmi les plateformes de développement existantes, nous avons choisi de présenter Zeus, MadKit, AgentBuilder, JADE et JACK.

Nous terminerons cette section, par une étude comparative entre ces différentes plateformes afin de justifier notre choix de l'environnement d'implémentation utilisé pour réaliser notre prototype.

3.2.5.1 Zeus

Zeus a été développée par *l'Agent Research Programm* du *British Telecom Intelligent System Research Laboratory*. Elle est dédiée à la conception d'applications à base d'agents collaboratifs [Collis et al., 1998], plus spécialement les systèmes économiques basés sur la planification ou l'ordonnancement.

La plateforme Zeus se base sur les concepts d'agent, de buts, et de tâches à réaliser par les agents du système pour atteindre leurs buts, et de faits pour représenter les croyances des agents.

Pour Zeus, un agent est constitué des trois couches suivantes :

- la couche de définition comprend les capacités de raisonnement ainsi que les algorithmes d'apprentissage ;
- la couche organisationnelle contient la base de connaissance et les accointances de l'agent ;
- la couche de coordination définit et décrit les interactions de l'agent avec les autres agents.

La plateforme Zeus offre une méthodologie basée sur quatre phases pour la construction d'agents :

- l'analyse du domaine : pour la modélisation des rôles. Pendant cette phase, Zeus utilise le langage UML, par contre aucun outil logiciel n'est fourni ;
- la conception : pour spécifier les buts et les tâches qui permettent de les réaliser. Cette spécification s'accomplit à travers du langage naturel, mais elle n'est supportée par aucun outil ;
- le développement en quatre étapes : la création de l'ontologie, la création des agents, la configuration de l'agent utilitaire et de l'agent tâche et enfin l'étape d'implémentation ;
- le déploiement décrit la façon d'exécuter le système multi-agents ainsi que le choix de la visualisation et du suivie de cette exécution. Cet outil permet d'identifier plusieurs visions sur l'organisation, et les interactions ayant lieu dans le groupe d'agents. Elle permet aussi la décomposition des tâches et des états internes des agents.

Enfin, Zeus est un environnement intégré de développement des SMA basé sur Java ce qui permet aux développeurs de réutiliser un ensemble de bibliothèques java afin de créer leurs agents.

3.2.5.2 MADKIT

La plateforme MADKIT (*Multi-Agent Development Kit*) a été développée à l'université de Montpellier II, par Ferber et Gutknecht [Ferber et Gutknecht, 1998]. La plateforme MADKIT supporte le développement de divers types de systèmes, mais elle semble bien adaptée pour les applications de simulation. Elle est équipée d'un environnement de développement graphique qui facilite la construction des applications.

MADKIT est basée sur les concepts du modèle AGR (Agent, Rôle, Groupe) [Ferber et Gutknecht, 1998], voir figure 3.8, développé au niveau de la méthodologie AALAADIN. Le modèle n'exige aucune contrainte sur l'architecture interne des agents. L'agent est défini comme une entité active communicante et qui joue des rôles dans différents groupes. La définition de l'agent est laissée aux développeurs afin de choisir le modèle le plus adapté aux agents de leurs applications. MADKIT fournit une API (*Application Programming Interface*) qui permet de construire l'agent, en

spécialisant une classe d'agent abstraite. Le noyau de MADKIT permet de lancer les agents, en proposant des services de gestion des groupes et de communication.

3.2.5.3 AgentBuilder

AgentBuilder [AgentBuilder, 2014] est une plateforme permettant de construire des agents intelligents. La réalisation des comportements des agents s'effectue en se basant sur le modèle BDI (Belief-Desire-Intention), et les communications inter-agents s'effectuent via le langage KQML [Azaiez, 2007].

La plateforme AgentBuilder est composée d'une interface graphique et d'un langage orienté agent qui permettent la définition des croyances, des engagements et des actions. Il permet aussi la description des ontologies et des protocoles de communications entre les agents. Le langage Radl (*Reticular Agent Definition Language*) permet de définir les agents, et les règles du comportement de l'agent.

3.2.5.4 JADE

La plateforme JADE (*Java Agent Development Framework*) [Bellifemmine et al., 1999], permet de développer et d'exécuter des applications distribuées basées sur le concept d'agents et d'agents mobiles. Cette plateforme est compatible avec le standard FIPA (*Foundation for intelligent physical agents*).

La plateforme JADE permet d'implémenter les agents avec les six propriétés suivantes :

- Autonomie : un agent JADE dispose de son propre *thread* pour le contrôle de ses actions, la prise de décisions et la gestion de son cycle de vie.
- Réactivité : les agents sont capables de percevoir les événements de leurs environnements et réagissent en fonction de ces événements.
- Aspect Sociaux : les agents présentent des aspects sociaux dans leurs comportements, leurs permettant de communiquer et d'interagir entre eux par l'intermédiaire de messages asynchrones. La communication est un type d'action qui peut intégrer un plan d'action.
- Dynamicité : les agents ont la capacité de découvrir d'une manière dynamique d'autres agents et de communiquer avec eux.
- Offre du service : chaque agent propose et tient à jour un ensemble de services. Un agent est capable de chercher des agents qui offrent les services dont il a besoin.
- Mobilité : les agents dans JADE sont implémentés dans des conteneurs et ils ont la possibilité de se déplacer.

JADEX [Pokahr et al., 2005], [Sudeikat et al., 2004] est un composant complémentaire à JADE, capable d'intégrer et d'implémenter l'architecture interne des agents. Il prend en compte aussi l'architecture des agents hybrides. En ce qui concerne les agents proactifs, JADEX se base sur le modèle BDI.

L'implémentation de la plateforme JADE est fondée sur Java, par conséquent elle peut être répartie sur plusieurs machines. Toutefois, JADE n'offre pas de méthodologie mais plusieurs méthodologies génèrent des codes compatibles avec la plateforme JADE, telles que GAIA, INGENIAS.

3.2.5.5 JACK

JACK [JACK, 2004] est un environnement de développement des SMA basé sur Java. Pour atteindre ces objectifs, l'agent est programmé avec un ensemble de plans ou d'intentions. Chaque plan définit la manière d'atteindre son but dans des contextes différents. En passant à l'exécution, l'agent cherche en permanence à accomplir ces buts (désirs), en adoptant des plans appropriés (intentions) et en s'appuyant sur les données (croyances) de l'état de l'environnement [Tarek, 2006].

Actuellement, il n'existe pas d'API dans JACK pour le développement et l'utilisation des protocoles d'interaction entre les agents. Les protocoles d'interaction sont implémentés sous forme de plans et l'activité du protocole est définie d'une manière procédurale. La plateforme JACK n'intègre pas les langages de communication tels que FIPA-ACL ou KQML, ce qui pose des difficultés pour exprimer la sémantique des messages entre les agents et les informations qu'ils comportent.

3.2.5.6 Étude comparative des plateformes agents

Pour comparer les différentes plateformes présentées, nous choisissons les critères suivants : type de systèmes visés, modèles d'agents supportés, langage d'implémentation, hétérogénéité, sécurité, conformité FIPA et type de licence (Tableau 3.2).

Plateforme	ZEUS	MADKIT	AgentBuilder	JADE	JACK
Types de systèmes visés	Systèmes collaboratifs	Applications de simulation	Tous types de systèmes	Systèmes distribuées, Agents mobiles	Tous types de systèmes
Modèles d'agents supportés	Réactifs, proactifs	Réactifs	BDI	Réactifs, proactifs et BDI	BDI
Langage d'implémentation	JAVA	JAVA	JAVA	JAVA	JAVA
Hétérogénéité	Oui	Oui	Non	Oui	Non
Sécurité	Oui	Oui	N/A	Oui	Oui
Conforme FIPA	Oui	Non	Non	Oui	Non
Type de licence	Open source	Libre pour l'utilisation éducative et payante	Payant, version démonstration disponible pour	Open source	Payant, version démonstration valable 60 jours

		pour les autres cas	quelques jours		
--	--	---------------------	----------------	--	--

Tableau 3.2 Comparaison des plateformes SMA

Après une analyse du tableau de comparaison, nous remarquons que les plateformes MADKIT, AgentBuilder et JACK ne sont pas conformes aux normes FIPA. Pour ZEUS, le principal inconvénient de cette plateforme réside dans le fait qu'elle ne permet pas de développer tous les types de systèmes SMA. La plateforme JADE répond parfaitement à nos attentes, en plus :

- Elle permet le développement de SMA conformes aux normes FIPA ;
- Les agents développés sous JADE, sont entièrement écrits en Java ;
- JADE assure une communication transparente par échange de messages dans le langage normalisé FIPA-ACL ;
- JADE est une plateforme de développement d'agents Open Source ;
- Communauté importante de développeurs et d'utilisateurs ;
- Exécution distribuée et concurrente des agents.

Dans la première partie de ce chapitre, nous avons présenté les agents et les systèmes multi-agents. Cette technologie représente actuellement l'un des moyens mis à la disposition des chercheurs pour modéliser et construire les systèmes dynamiques et distribués. Dans la section suivante nous présenterons le raisonnement à partir de cas fondé sur les SMA.

3.3 Systèmes RàPC fondés sur SMA

Les systèmes multi-agents ont été appliqués dans des systèmes de raisonnement à partir de cas pour résoudre certains scénarios ou l'apprentissage par renforcement. Par exemple, Haynes et al [Haynes et al., 1998] appliquent le raisonnement à partir de cas au problème proie/prédateur, où chaque prédateur permet d'apprendre les cas du comportement des autres agents. Utilisant les cas stockés, un prédateur peut prédire le mouvement des autres prédateurs afin d'améliorer leur coordination.

3.3.1 Classification des systèmes RàPC fondés sur SMA

Le raisonnement à partir de cas basé sur les systèmes multi-agents est utilisé dans plusieurs domaines d'application. Ils peuvent être classés en utilisant comme critères l'organisation et la gestion des connaissances par le système (par exemple, base de cas unique vs base de cas multiple), et le traitement des connaissances par le système (par exemple, traitement des connaissances par un seul agent vs traitement des connaissances par plusieurs agents ou traitement multi-agents). Dans le domaine des systèmes RàPC fondés sur les SMA, une caractéristique notable est l'autonomie des agents. Deux facteurs clés gouvernent l'autonomie d'un agent : (1) sa capacité à déterminer s'il est capable de résoudre un problème seul, et (2) sa capacité à interagir avec d'autres agents, par la négociation, la coopération et la collaboration pour obtenir une solution globale à un problème donné. En se basant sur ces facteurs, nous classons

les systèmes RàPC fondés sur les SMA en deux catégories d'applications (nous signalons que c'est un travail original) [Zouhair et al., 2012] :

1. Les applications multi-agents dans lesquels chaque agent utilise le raisonnement à partir de cas en interne pour son propre besoin (raisonnement à partir de cas niveau agent).
2. Les systèmes multi-agents (SMA) dont la démarche est un raisonnement à partir de cas (raisonnement à partir de cas niveau SMA).

3.3.1.1 Raisonnement à partir de cas orienté agent

Le raisonnement à partir de cas niveau agent est le premier modèle qui a été utilisé lors de l'application des agents au RàPC. Pour ce type de système, chaque agent est capable de rechercher les cas similaires au cas cible dans sa propre base tel que le système POMAESS (e-service) [Yu et al., 2003], ou dans des bases de cas distribuées comme le système MCBR [Leake et Sooriamurthi, 2001]. Il est capable aussi d'exécuter le cycle du RàPC. Nous pouvons aussi citer AMAL [Ontañón et Plaza, 2007] pour les systèmes d'argumentation multi-agents, et CCBR pour la planification personnalisée d'itinéraire [Ginty et Smyth, 2001].

3.3.1.2 Raisonnement à partir de cas orienté multi-agents

Une seconde catégorie regroupe les systèmes où le RàPC est un SMA. Pour ce type d'application les agents qui constituent le système collaborent ensemble pour exécuter le cycle du raisonnement RàPC. C'est une démarche RàPC, qui est plus riche par rapport à la première. En effet l'expérience individuelle d'un agent seul peut être limitée, par conséquent leurs connaissances et leurs prédictions aussi. Ainsi, les agents sont capables de coopérer avec d'autres agents pour une meilleure prédiction de la situation. Nous pouvons citer comme exemples ProCLAIM [Tolchinsky et al., 2006] pour les systèmes d'argumentation multi-agents et CBR-TEAM [Prasad et al., 1995] système qui gère un condensateur de vapeur.

Le tableau suivant présente les principaux systèmes de raisonnement à partir de cas statiques/dynamiques fondés sur les SMA, classés selon leur type de traitement des connaissances.

Cas cible	Cycle RàPC	Systèmes RàPC Multi-agents	
		RàPC niveau Agent	RàPC niveau SMA
Statique	Statique	FPL [Plaza et al., 1997], CCBR [Ginty et Smyth, 2001], AMAL [Ontañón et Plaza, 2007]	ProCLAIM [Tolchinsky et al, 2006]
Dynamique	Statique	CICLMAN [Soh et Luo, 2003], RoBoCats [Marling et al., 2003], POMAESS [Yu et al., 2003], S-MAS [Cristian Pinzón et al., 2011]	MCBR [Leake et Sooriamurthi, 2001], CBR-TEAM [Prasad et al., 1995]

Tableau 3.3 : RàPC fondé sur SMA avec cas cible statique/dynamique [Zouhair et al., 2014]

3.3.2 Principaux systèmes RàPC fondés sur les SMA

Nous présentons les principaux systèmes RàPC fondés sur les SMA avec leur principe et les méthodes déployées. Chaque application sera décrite par son nom par une ou plusieurs références bibliographiques, le type du système (RàPC niveau agent ou SMA), la représentation du cas, la phase de remémoration, la mesure de similarité, et la description du système.

3.3.2.1 Système FPL

Federated Peer Learning (FPL) [Plaza et al., 1997]	Plaza, Arcos et Martin proposent une approche du raisonnement à partir de cas coopératif dans le domaine de la purification des protéines.
Type RàPC	système orienté agent.
Représentation de cas	un cas dans FPL est représenté par un graphe.
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des historiques de valeurs (programmation dynamique).

Dans le système FPL (*Federated Peer Learning*) chaque agent utilise la même représentation d'information, ce qui permet aux différents agents de communiquer sans passer par une phase intermédiaire de traduction. Chaque agent est capable de résoudre des problèmes en se basant sur ses propres expériences. FPL est un système coopératif où chaque agent a son propre point de vue pour chaque problème. L'approche FPL fonctionne selon deux modes de coopération : DistCBR et ColCBR.

- dans DistCBR un agent peut transmettre des problèmes à d'autres agents. Chaque agent qui reçoit un problème d'un autre agent doit le résoudre selon ses propres connaissances et ses méthodes de résolution de problèmes.
- dans ColCBR, un agent peut envoyer des problèmes aux autres agents en précisant la méthode de résolution qui doit être utilisée. Ces agents sont donc juste une prolongation de la mémoire de l'agent émetteur des problèmes.

3.3.2.2 Système CICLMAN

CICLMAN [Soh et Luo, 2003]	Le système CICLMAN (<i>Combining Individual and Cooperative Learning for Multi-Agents Negotiations</i>), est un RàPC fondé sur les SMA, développé dans le cadre du projet <i>Autonomous Negotiating Teams</i> , pour la réallocation des ressources d'un processeur.
Type RàPC	système orienté agent.
Représentation de cas	liste des attributs.
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des historiques de valeurs.

Le but du projet est de savoir si l'apprentissage combiné (à la fois coopératif et individuel) peut apporter de meilleures performances pour le système multi-agents que

l'apprentissage individuel seul. Un cas dans l'approche de Soh et Luo est une liste de propriétés d'un processeur avec ses valeurs associées à un instant donnée. Chaque cas est lié à un agent. Chaque agent possède deux bases de cas privées : une pour stocker les cas initiaux et l'autre pour les réponses de leur négociation avec les autres agents. Les agents négocient entre eux pour trouver une meilleure solution à un problème donné en se basant sur le raisonnement à partir de cas via une stratégie dynamique de négociation. La recherche de solution d'un problème dans le système de CICLMAN commence tout d'abord par la recherche individuelle d'un agent. Si la solution n'est pas satisfaisante, alors une approche de recherche coopérative entre plusieurs agents sera exécutée.

• L'apprentissage individuel

Dans ce type d'apprentissage, les agents résolvent les problèmes si les cas sont contenus dans leurs bases locales de cas en gardant l'historique de la qualité de chaque cas résolu. Il recherche alors le cas le plus semblable dans la base de cas. Le cas retenu sera adapté comme solution au problème courant (via une stratégie de négociation avec les autres agents). À la fin de la négociation, l'agent décide si le nouveau cas sera stocké dans la base de cas. Chaque cas contient son historique d'utilisation, et en se basant sur cet historique, un agent évalue la qualité de la solution adaptée. Lorsque la qualité de solution d'un problème n'est pas suffisante, un protocole d'apprentissage coopératif est exécuté pour obtenir une meilleure solution proposée par l'un des autres agents afin de remplacer l'ancienne solution. L'architecture générale de l'apprentissage individuel est décrite dans la figure suivante (Figure 3.9).

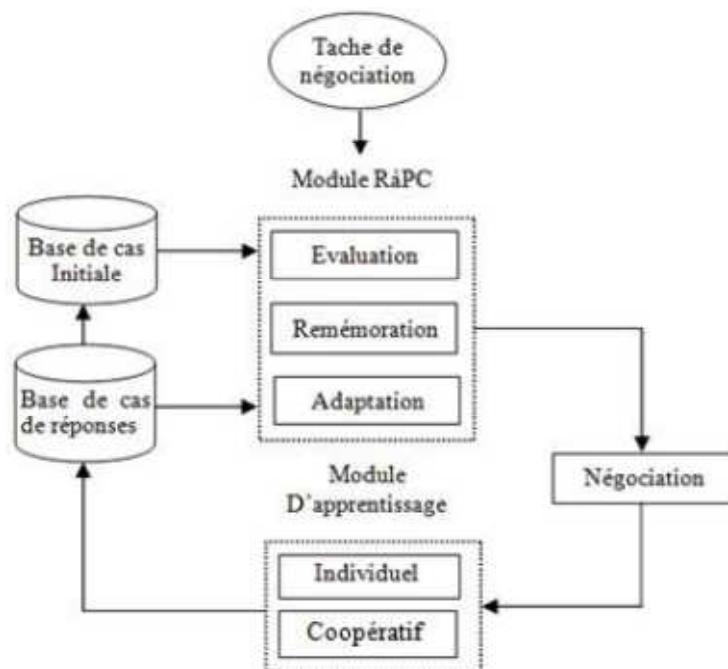


Figure 3.9 : L'approche d'apprentissage individuel [Soh et Luo, 2003]

• L'apprentissage coopératif

Quand la base de cas d'un agent ne contient aucun cas similaire au problème, la base de cas d'un autre agent peut le contenir. Soh et Luo proposent alors un apprentissage coopératif. La figure 3.10 montre le déroulement des opérations d'apprentissage coopératif : le profilage d'un cas, le déclencheur de l'apprentissage coopératif, la demande d'aide à l'agent voisin, l'échange de cas (demande et réponse), et l'adaptation au cas cible de la solution retrouvée. Soh et Luo proposent plusieurs stratégies de négociation selon le cas traité :

- L'agent évalue le cas pour déterminer s'il retrouve un cas similaire au cas cible, mais dont la solution adaptée au cas cible n'est pas satisfaisante. Dans ce cas l'agent peut demander de l'aide auprès d'autres agents.
- L'agent demande l'aide d'un autre agent qu'il pense expert dans ce sujet. Chaque agent maintient un profil de chaque agent voisin qui documente les relations de négociation entre les deux agents. Parmi les paramètres profilés on trouve le nombre de fois que l'agent a demandé l'aide du voisin en question. L'agent sélectionne le voisin qui a le plus grand nombre de demande.

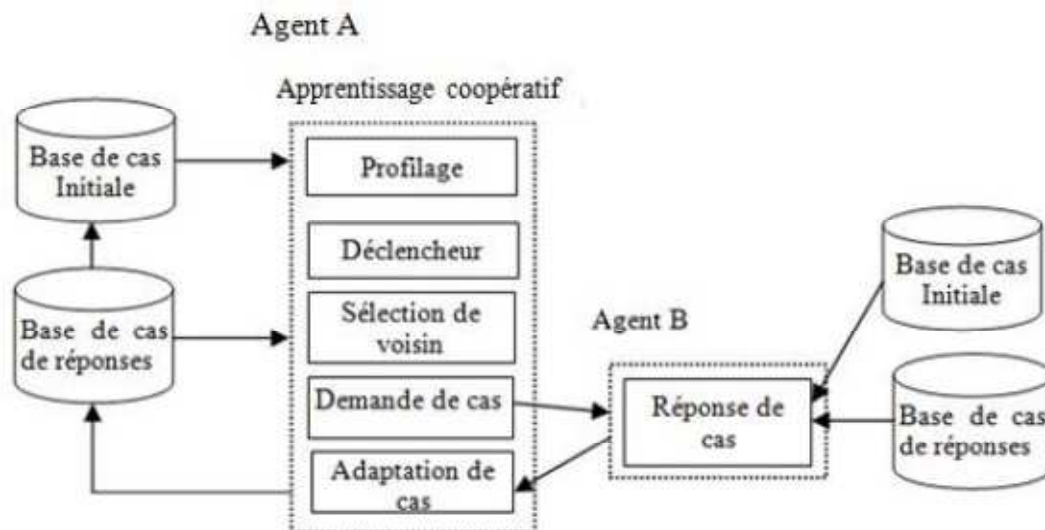


Figure 3.10 : L'approche d'apprentissage coopératif [Soh et Luo, 2003]

On peut déduire du travail de Soh et Luo que le travail des agents en coopération est coûteux en ressource, et donc leur méthode n'utilise ce mode de travail que lorsqu'il est vraiment nécessaire.

3.3.2.3 Système CCBR

CCBR [Ginty et Smyth, 2001]	CCBR (<i>Collaborative Case Based Reasoning</i>), est un système pour la planification d'itinéraire.
Type RàPC	système orienté agent.
Représentation de cas	liste des attributs.

Remémoration / Mesure de similarité	recherche par point de vue / distance géométrique.
--	--

Mc Ginty et Smyth proposent un système pour la planification d'itinéraire, très proche de l'approche FPL. La principale différence entre FPL et CCBR est que dans cette dernière un agent résout tous les problèmes localement. Les besoins de chaque utilisateur pour la planification d'itinéraire sont traités par un seul agent RàPC en se basant sur une base de cas locale (des plans de routes préférées par cet utilisateur). Dans la situation où un utilisateur donné a besoin de planifier un itinéraire dans un territoire inconnu, l'agent concerné par cette requête cherchera à localiser et réutiliser la base de cas d'un autre agent ayant une expérience dans le territoire cible et avec les mêmes préférences.

L'architecture du système CCBR, illustrée par la figure 3.11, est une architecture multi-agents, composée d'un ensemble d'agents RàPC homogènes, $A_1, \dots, A_n$. Les agents ont les mêmes capacités RàPC, mais diffèrent dans leurs expériences et leurs résolutions des problèmes. Lorsqu'un agent A_i cherche à résoudre un problème qui sort de son domaine d'expertise, le problème est envoyé aux autres agents. Si l'un des autres agents A_j a un cas similaire pour résoudre le problème, alors il répond à l'agent A_i , émetteur du problème, en lui montrant comment le résoudre et en donnant l'accès à A_i pour la base de cas de A_j .

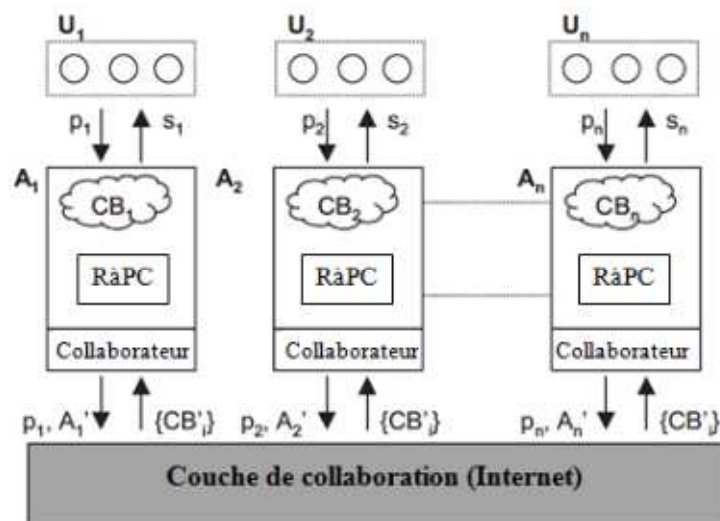


Figure 3.11 : Architecture du système CCBR [Ginty et Smyth, 2001]

Dans CCBR, chaque agent est responsable de l'évaluation d'un problème qui fait partie de son domaine d'expertise. Mc Ginty et Smyth appliquent avec succès l'approche CCBR à la planification personnalisée d'itinéraire, où chaque agent contient une partie du domaine d'expérience afin de répondre à une requête spécifique d'un utilisateur.

3.3.2.4 Système MCBR

MCBR [Leake et Sooriamurthi, 2001], [Leake et Sooriamurthi, 2002]	MCBR (<i>MultiCase-Base Reasoning</i>), traite des systèmes distribués où il y a plusieurs bases de cas disponibles pour la même tâche. C'est un système générique.
Type RàPC	système orienté SMA.
Représentation de cas	liste des attributs.
Remémoration / Mesure de similarité	recherche multidimensionnelles / comparaison des matrices de covariance dynamique.

Leake et Sooriamurthi présentent une autre approche, appelée MCBR (*MultiCase-Base Reasoning*), très proche des travaux de Mc Ginty et Smyth. MCBR traite des systèmes distribués où il y a plusieurs bases de cas disponibles pour la même tâche. Chaque base de cas peut ne pas correspondre exactement au même problème que les autres bases, ou peut refléter des préférences particulières des utilisateurs. Par conséquent les cas doivent être adaptés pour pouvoir les déplacer d'une base de cas à une autre. Selon Leake et Sooriamurthi, il existe deux approches pour traiter les bases de cas multiples (figure 3.12 et 3.13) : une approche *eager* qui permet de fusionner et normaliser toutes les bases de cas dans une seule base centrale, et la seconde approche *lazy* permet de travailler avec une seule base de cas locale et de rechercher les cas dans des bases de cas externes si nécessaire.

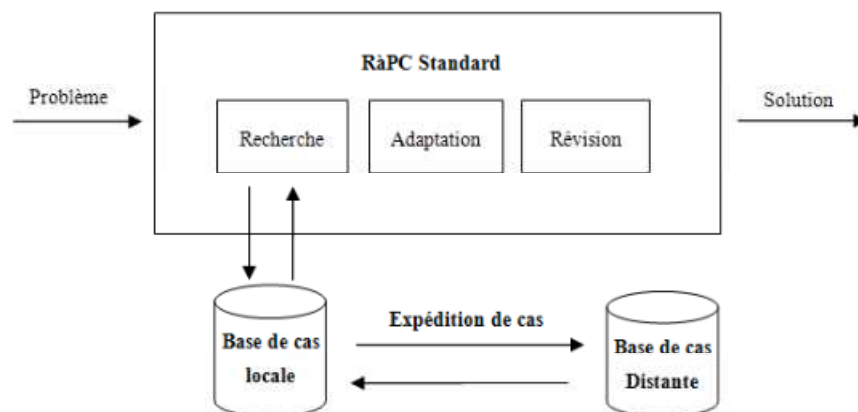


Figure 3.12: Approche RàPC utilisant une base de cas externe lorsque la base de cas locale est insuffisante [Leake et Sooriamurthi, 2002]

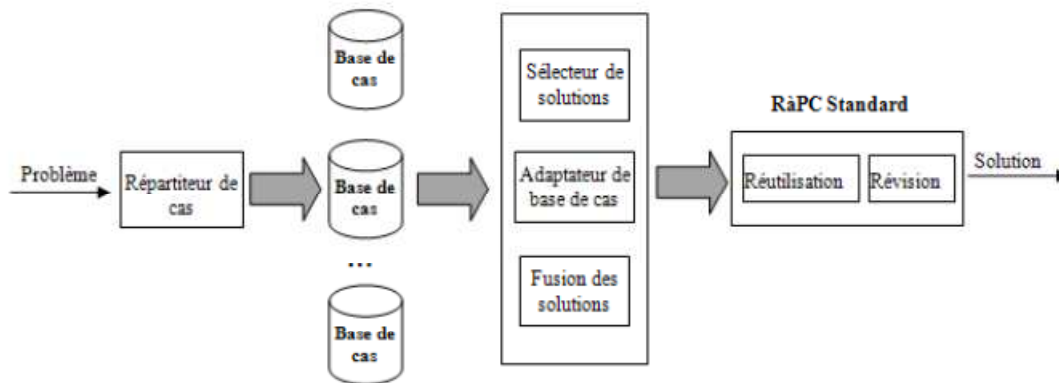


Figure 3.13 : Approche du RàPC multi-base en s'appuyant sur plusieurs bases de cas [Leake et Sooriamurthi, 2002]

Les deux méthodes abordent les problèmes d'adaptation et de modification d'un cas appartenant à une base de cas externe afin de l'ajuster dans la base de cas locale, alors que l'approche *lazy* a un problème supplémentaire d'expédition de cas. Leake et Sooriamurthi [Leake et Sooriamurthi, 2002] présentent plusieurs stratégies pour accomplir l'adaptation et la modification des cas entre les bases de cas : aucune adaptation, interpolation linéaire et approximation locale, et deux stratégies pour résoudre le problème d'expédition des cas : expédition basée sur un seuil *threshold-based dispatching* et expédition basée sur cas *case-based dispatching*. Plus tard [Leake et Sooriamurthi, 2004] ont montré qu'en combinant des techniques appropriées d'expédition du cas et d'adaptation de cas, l'approche *lazy* est plus performante que l'approche *eager*, car l'approche *lazy* maintient toujours le contrôle en résolvant un problème en utilisant les deux types de base de cas locaux et externes.

3.3.2.5 Système CBR-TEAM

CBR-TEAM [Prasad et al., 1995]	CBR-TEAM gère le fonctionnement d'un condensateur de vapeur.
Type RàPC	système orienté SMA.
Représentation de cas	liste des attributs.
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des valeurs via des formules mathématiques

Historiquement c'est le premier système à proposer un RàPC fondé sur les SMA. Dans CBR-TEAM, chaque agent a un point de vue partiel et imparfait de la situation, ce qui permet à chaque agent d'avoir une vision locale pour chaque problème (cas cible). Les cas dans le système CBR-TEAM sont composés de *sous-cas*, répartis entre les différents agents du système (c'est-à-dire chaque agent est intéressé par un sous-ensemble de caractéristiques de chaque cas, qui sera stocké par l'agent). Les agents du système CBR-TEAM accèdent à leurs bases de cas afin de rechercher les meilleurs *sous-cas*, qui peuvent être combinés d'une façon cohérente pour fournir une

solution complète au cas cible. Lorsqu'un utilisateur externe (ou un agent) envoie une requête, il n'y a peut-être aucun agent qui a toutes les informations demandées. Prasad, Lesser et Lander proposent une approche décentralisée qui permet au groupe d'agents de rechercher l'information appropriée dans la base de cas de chaque agent et de l'agréger afin de répondre à une requête.

3.3.2.6 Système ProCLAIM

ProCLAIM [Tolchinsky et al., 2006] [Tolchinsky et al., 2012]	ProCLAIM est un modèle de négociation dans le domaine de la santé (transplantation d'organes humain).
Type RàPC	système orienté SMA.
Représentation de cas	graphe.
Remémoration / Mesure de similarité	recherche multidimensionnelles / distance ontologique + distance entre graphe.

ProCLAIM dispose d'un ensemble d'agents qui tentent de résoudre un problème de classification en agrégeant leurs connaissances et en se basant sur la délibération collaborative. L'approche ProCLAIM se compose d'un agent médiateur MA (*Mediator Agent*) et de plusieurs agents participants PA_i (*Participant Agent*) $1 \leq i \leq n$ (figure 3.14) dans un cadre de collaboration et de prise de décision en se basant sur l'argumentation. L'agent MA contrôle les agents participants de telle sorte que la décision finale soit conforme aux directives des experts du domaine. Les arguments proposés par les agents participants peuvent également convaincre l'agent MA d'accepter des décisions qui s'écartent des principes directeurs.

Le système ProCLAIM définit trois tâches principales pour l'agent MA :

1. Informer les agents participants des propositions de décisions possibles à chaque étape de la délibération ;
2. S'assurer que les arguments présentés sont pertinents (c.-à-d. conformes aux lignes directrices) ;
3. Évaluer les arguments présentés dans le but d'identifier les arguments gagnants et donc de déterminer si une décision proposée est valide. Cette dernière tâche peut exiger la justification et l'explication des arguments fournis par les agents participants et éventuellement la présentation d'arguments supplémentaires.

Afin de mener à bien toutes ces tâches, l'agent MA se base sur quatre sources de connaissances :

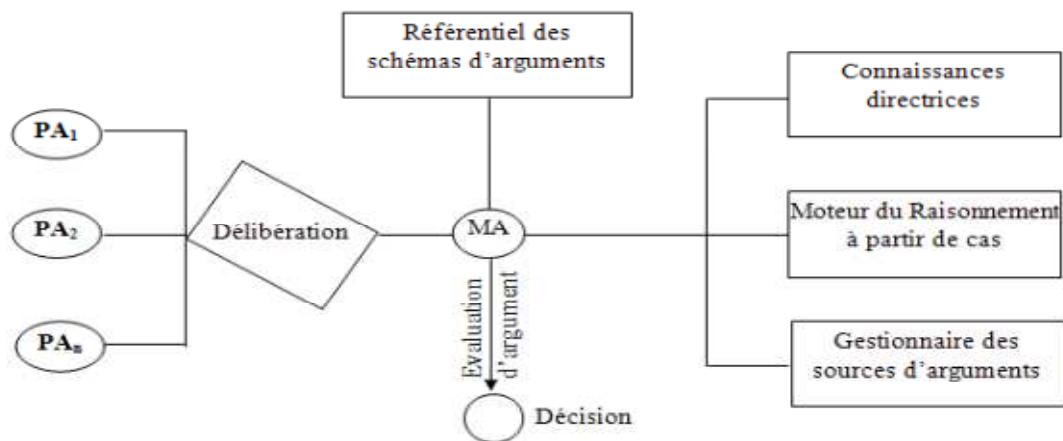


Figure 3.14 : Architecture du système ProCLAIM [Tolchinsky et al., 2012].

1. Référentiel des schémas d'arguments : afin d'orienter les agents promoteurs dans leur délibération, l'agent MA se sert d'une base des arguments et de leurs questions critiques associées, formalisés d'une manière à définir un protocole d'échange basé sur les arguments.
2. Connaissances directrices : ce composant permet à l'agent MA de vérifier si les arguments sont conformes aux connaissances établies.
3. Moteur du raisonnement à partir de cas permet à l'agent MA d'attribuer les points forts aux arguments présentés, ainsi que de fournir des arguments supplémentaires jugés pertinents dans les situations précédentes similaires.
4. Gestionnaire des sources d'arguments : selon la source à laquelle les arguments sont présentés, les points forts de ces arguments peuvent être réajustés par l'agent MA. Ainsi, ce composant gère les connaissances liées aux rôles des agents et/ou aux réputations des agents, qui peuvent autoriser des agents à prendre des décisions exceptionnelles.

3.3.2.7 Système RoboCats

RoboCats [Marling et al., 2003]	RoboCup a pour but de créer une équipe de football robotisée. RoboCats participe à la ligue des robots de petites tailles comportant cinq joueurs.
Type RàPC	système orienté agent.
Représentation de cas	liste des attributs.
Remémoration / Mesure de similarité	recherche par point de vue / comparaison des historiques de valeurs (une mesure de similarité par variable).

RoboCup comporte plusieurs ligues séparées pour des équipes de robots de petites tailles, de tailles moyennes, de grandes tailles, plus une ligue du football de simulation. Le but de ces compétitions est de créer une équipe de football robotisée capable de battre la meilleure équipe humaine du football dans la coupe du monde de l'année

2050. D'ici là, de nombreux problèmes liés aux agents physiques pour les environnements dynamiques en temps réel devront être explorés et résolus.

Le raisonnement à partir de cas est utilisé dans le système RoboCats afin que les agents se déplacent sur un plan tout à la fois d'une manière individuelle et par équipe. Le système RoboCats permet aussi aux robots joueurs (les agents physiques) de fonctionner dans un environnement dynamique et en temps réel. Plus précisément, RoboCats est capable de :

- préciser la position du gardien de but ainsi que celle des autres joueurs ;
- avoir des informations sur l'équipe : déplacements, réactions ;
- reconnaître l'état de jeu ;
- avoir une stratégie de jeu efficace ;
- avoir une coordination entre les différents joueurs ;
- tester des prototypes en combinant le RàPC et le système multi-agents.

RoboCats utilise une hiérarchie hybride, basée sur une planification distribuée de l'équipe (Figure 3.15). Dans la ligue de RoboCup, une camera est montée au-dessus du terrain de jeu afin de fournir une vision complète pour l'équipe. Des mises à jour régulières de la vision de la caméra alimentent l'agent *Worldview*. Ce dernier analyse les données de la camera pour déterminer l'état du jeu, et rend ses analyses disponibles aux autres agents : agent-meta et aux cinq agents-joueurs. L'agent-meta prend des décisions de haut niveau pour l'équipe. Un agent joueur et un agent robot sont associés à chaque robot physique.

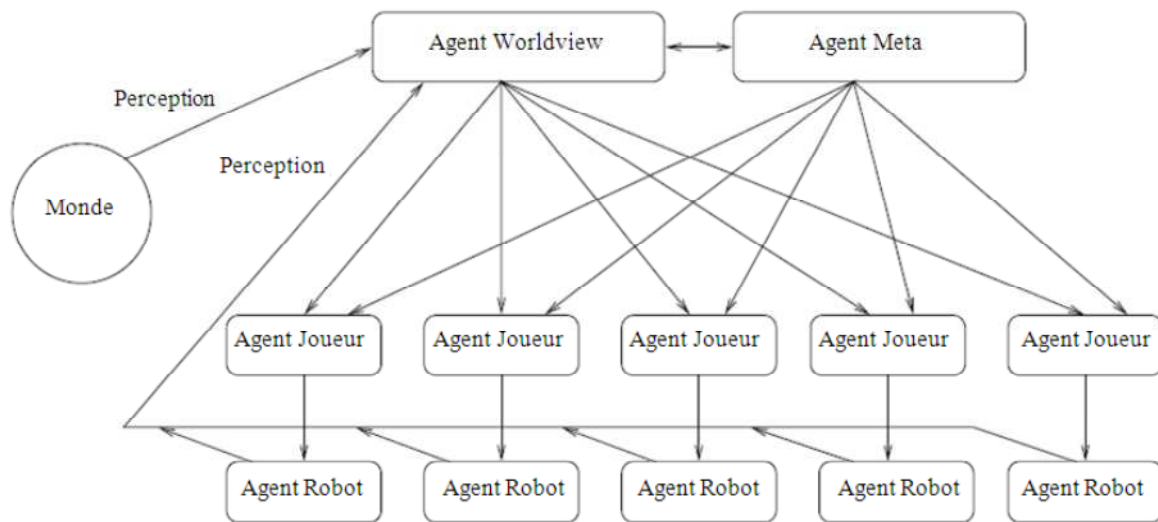


Figure 3.15 : Architecture du système RoboCats [Marling et al., 2003]

Le système RoboCats utilise les expériences passées pour déterminer où le gardien de but devrait se déplacer pour bloquer avec succès un adversaire. Un cas du gardien de but (tableau 3.4) est décrit à un instant donné par des données brutes : la position et l'orientation du gardien de but, la position et l'orientation de chaque attaquant et

défenseur, et la position du ballon. Dans le tableau 3.4, les positions sont représentées par les coordonnées X, Y. Pour les orientations, elles sont exprimées en degrés à partir de l'axe des X.

Position initiale du gardien de but	9 92
Orientation initiale du gardien de but	118
Position initiale de l'attaquant	73 141
Orientation initiale de l'attaquant	309
Position initiale du défenseur	34 60
Orientation initiale du défenseur	103
Position initiale du ballon	65 131
Nouvelle position du gardien de but	11 89
Nouvelle orientation du gardien de but	118
Direction du mouvement du gardien de but	Gauche
Nouvelle position du ballon	42 87
Résultats	succès

Tableau 3.4 : exemple de cas gardien de but

3.3.2.8 Système AMAL

AMAL [Ontañón et Plaza, 2007] [Ontañón et Plaza, 2011]	AMAL, est un système d'argumentation pour la délibération, la négociation, et la résolution des conflits dans le domaine de l'identification des éponges marines.
Type RàPC	système orienté agent.
Représentation de cas	liste des attributs.
Remémoration / Mesure de similarité	recherche multidimensionnelles / comparaison des valeurs via des formules mathématiques.

AMAL basé sur deux buts : (1) la délibération commune, et (2) l'apprentissage à partir de la communication.

1. L'argumentation basée sur la délibération commune comprend une discussion puis un résultat commun d'une situation particulière et enfin l'établissement d'un plan d'action adapté à la situation donnée.
2. L'apprentissage à partir de la communication où les agents sont capables d'apprendre à partir des expériences passées. Les agents sont capables de coopérer avec d'autres agents pour une meilleure prédiction de la situation en se basant sur le processus d'argumentation.

Dans AMAL les agents sont capables de travailler dans le même domaine à l'aide d'une ontologie partagée. Chaque agent est autonome et a seulement accès à sa base de

cas privée. Un cas contient la description du problème P , la solution associée S et la prédiction justifiée de S par la description symbolique D .

Dans AMAL il existe trois types d'argumentations : les prédictions justifiées, contre-arguments, et contre-exemples :

- Une prédiction justifiée est générée par un agent A_i de telle sorte que A_i estime que la bonne solution pour un problème donné P est S , et les preuves fournies sont la justification D .
- Un contre-argument contient une prédiction justifiée générée par un agent A_j avec l'intention de réfuter un argument généré par un autre agent A_i .
- Un contre-exemple est un cas qui contredit un argument α . Ainsi, un contre-exemple est également un contre-argument [Ontañón et Plaza, 2007].

En échangeant des arguments et contre-arguments (y compris les contre-exemples), les agents peuvent argumenter la solution d'un problème donné, c'est à dire qu'ils peuvent engager un processus de délibération commune. Toutefois, ils ont besoin d'une relation de préférence entre les contre-arguments, et d'une politique de décision afin de générer des contre-arguments. Les auteurs définissent aussi une mesure de confiance pour chaque prédiction justifiée, et la prédiction justifiée qui a la confiance la plus élevée sera choisie. Si le processus d'argumentation permet d'arriver à une solution consensuelle, la délibération commune se termine, sinon un vote pondéré est utilisé pour déterminer la solution convenable.

3.3.2.9 Système POMAESS

POMAESS [Yu et al., 2003]	POMAESS (<i>Problem-Oriented Multi-Agent-Based E-Service System</i>) est un service de maintenance à distance.
Type RàPC	système orienté agent.
Représentation de cas	liste des vecteurs.
Remémoration / Mesure de similarité	recherche par point de vue / distance euclidienne.

Dans POMAESS chaque agent est un expert de résolution de problème en se basant sur les différentes connaissances de base et de différents sites concernés par le télé-service.

Le système POMAESS comporte trois types d'agents experts capables d'assurer une qualité de service à distance :

1. Des experts en gestion de production.
2. Des experts de maintenance, dont le rôle est d'optimiser la disponibilité de plateforme de maintenance en minimisant les coûts directs et indirects.
3. Des experts de contrôle, dont le rôle est de s'assurer que le fonctionnement global de la plateforme est cohérent avec la stratégie de cette dernière.

Les auteurs de POMAESS se basent sur les contraintes précédentes et sur les différents modèles de coopération et négociation pour définir leur protocole de négociation :

- Un type de collaboration pour un objectif commun (et non pas un genre de compétition) ;
- Un temps de négociation limité ;
- Le jugement d'un agent sur les propositions des autres ne sera pas absolument accepté ou rejeté, mais c'est un choix à moyen terme. Dans la plupart des situations, aucun agent n'a une connaissance globale de tout le problème, le problème est donc de trouver un médiateur.

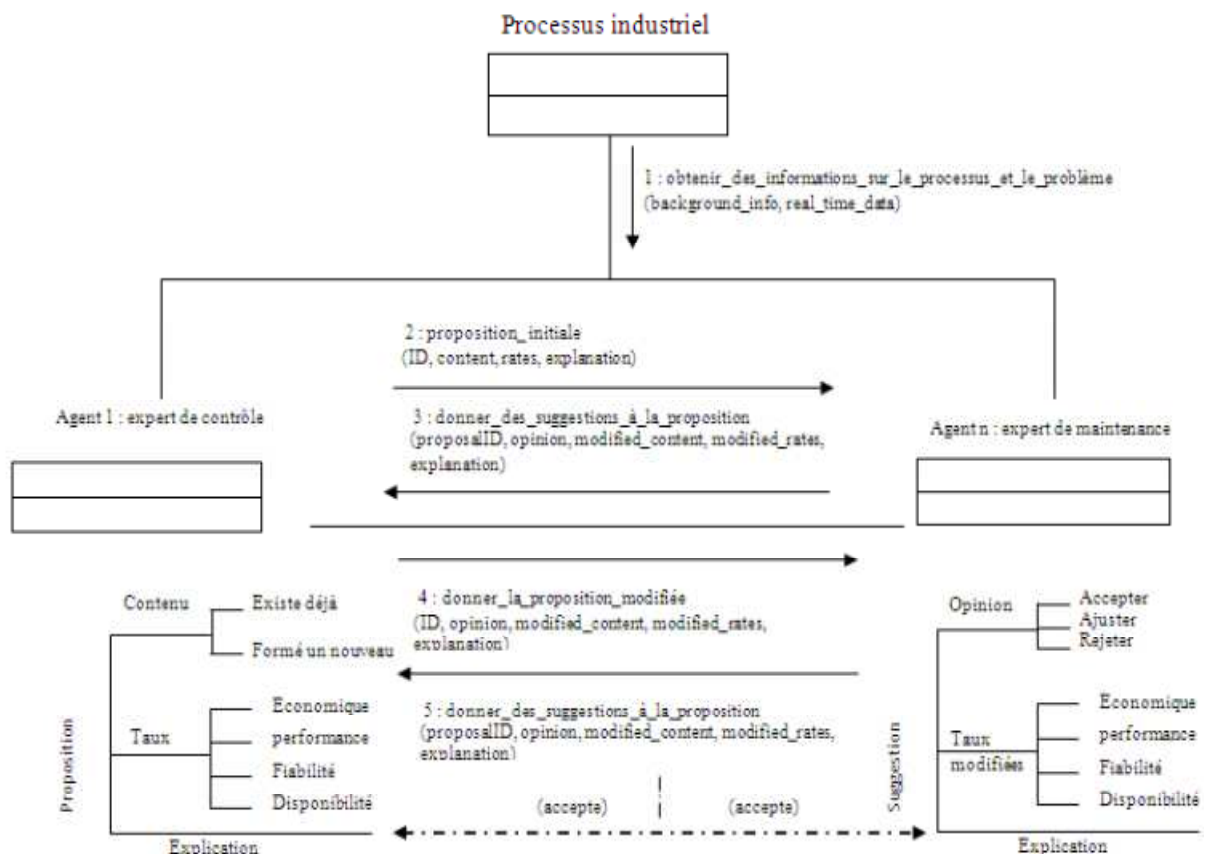


Figure 3.16 : Scénario de négociation dans POMAESS

L'architecture de la coordination dans POMAESS est un système multi-agents composé de deux types d'agents. Le premier est l'agent de négociation NA, qui exécute et organise la négociation, calcule l'ensemble des résultats sur la base des propositions d'agent experts EA, mais ne donne jamais sa propre opinion sur la solution du problème. Le second type est composé d'agents experts EA organisé sous forme de comité. Ils sont experts pour la résolution de problèmes. L'agent de négociation fournit un tableau noir et tous les agents experts peuvent obtenir des informations pertinentes sur le problème en cours et sur les autres agents experts.

3.3.2.10 Système S-MAS

S-MAS [Pinzón et al., 2011]	le système S-MAS permet d'identifier une attaque dans le domaine du service web.
Type RàPC	système orienté agent.
Représentation de cas	liste des attributs.
Remémoration / Mesure de similarité	recherche multidimensionnelles / distance euclidienne.

Les services web sont des applications permettant l'intégration et l'interopérabilité des systèmes répartis. Ils relient des programmes, des bases de données ou des processus métiers à l'aide de XML et de protocoles sur Internet. Toutefois, la sécurité n'a pas évolué proportionnellement au trafic et il n'existe pas de solution simple permettant la sécurisation du trafic entre les différents composants du service. L'une des attaques la plus commune, DoS (*Denial of Service*), a pour but de rendre indisponible un service, ou d'empêcher les utilisateurs légitimes d'un service de l'utiliser.

Le point fort de l'approche S-MAS est sa capacité d'auto-adaptation aux changements qui surviennent dans les techniques d'attaque. S-MAS est une architecture multi-agents hiérarchique et distribuée. Les différentes tâches du système sont distribuées sur les différents agents qui constituent les quatre niveaux de l'architecture présentée dans la figure 3.17.

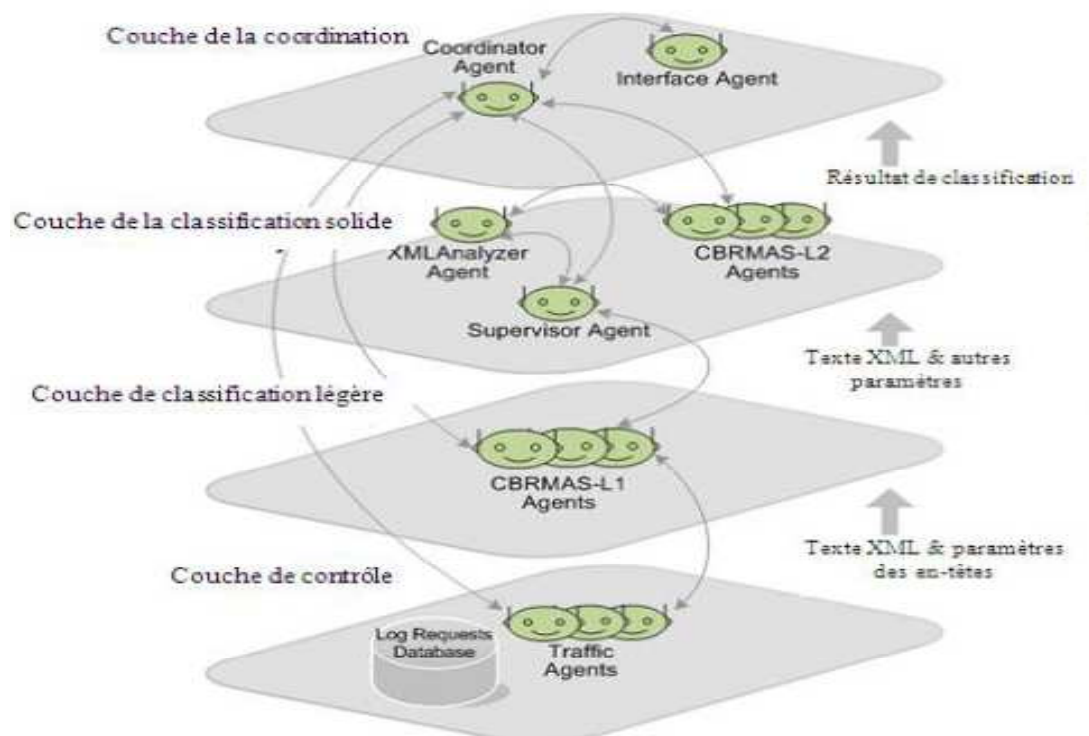


Figure 3.17 : L'architecture du système S-MAS [Cristian Pinzón et al., 2011]

Le but du système S-MAS est de vérifier et classifier un utilisateur d'un service web dans l'une des catégories attaque, non identifiée, pas d'attaque. Afin d'atteindre ce but S-MAS fonctionne en deux étapes. La première phase est réalisée par les agents CBRMAS-L1. Dans l'étape de révision du cycle du RàPC incorporé par les agents CBRMAS-L1 un test de satisfaction est effectué. Si la solution n'est pas satisfaisante alors une deuxième phase de classification est lancée à l'aide des agents CBRMAS-L2. Ci-après nous présentons les agents qui composent l'architecture S-MAS :

Agents de trafic : ces agents ont pour rôle principal d'identifier et de capturer tout le trafic à destination du serveur et plus particulièrement tout trafic qui contient les paquets de message SOAP. Les messages capturés sont envoyés à la couche supérieure afin d'être classés suite à l'exécution du processus de classification.

Agents CBRMAS-L1 : ces agents sont situés à la deuxième couche de l'architecture et sont chargés d'exécuter la première phase du processus de classification. Ce dernier est basé sur les données envoyées par les agents de trafic. Les agents CBRMAS-L1 permettent d'initier une classification en incorporant un moteur de raisonnement à partir de cas. S-MAS intègre une stratégie d'arbre de décision afin de retrouver les cas les plus similaires au cas cible. L'objectif est de réaliser une classification efficace sans consommer beaucoup de ressources.

Agents CBRMAS-L2 : ces agents de la troisième couche complètent le processus de classification. Afin de lancer cette phase, il est nécessaire d'avoir, au préalable, une analyse syntaxique du message SOAP, reçu par le serveur, pour extraire les données requises. Cette analyse est achevée par l'agent XMLAnalyzer. Les agents CBRMAS-L2 permettent de réaliser une classification en incorporant un moteur de raisonnement à partir de cas. Une fois que les données ont été extraites du message, l'étape de réutilisation du cycle RàPC s'exécute en se basant sur l'approche réseaux de neurones.

Agent superviseur: cet agent de la troisième couche du S-MAS, supervise l'agent XMLAnalyzer, car il existe toujours la possibilité d'avoir une attaque pendant le traitement syntaxique du message SOAP.

Agent XMLAnalyzer: cet agent est chargé de l'analyse syntaxique du message SOAP. L'analyse est effectuée en utilisant l'analyseur SAX [Brownell, 2002]. Les données extraites de l'analyse sont envoyés aux agents CBRMAS-L2.

Agent coordonnateur: cet agent de la quatrième couche, est chargé de la supervision et du bon fonctionnement de l'architecture. En outre, il supervise le processus de classification. Chaque fois qu'une classification est étiquetée comme suspecte, l'agent interagit avec l'agent interface afin de lui demander un examen d'expert. Enfin, cet agent coordonne les actions nécessaires pour répondre à ce type d'attaque.

Agent de l'interface: cet agent de la couche supérieure a été conçu pour supporter les différents périphériques (PDA, ordinateur portable, et Workstation). Il est chargé

de faciliter la communication avec l'utilisateur quand une alerte a été envoyée par l'agent coordonnateur.

Dans ce qui suit nous décrivons les étapes du cycle RàPC du système S-MAS exécuté par l'agent CBRMAS-L1.

Phase de remémoration: pendant cette étape, les cas associés au service Web demandé et la mémoire de règles correspondante sont recherchés. Le stockage et la récupération de règles à partir de la mémoire de règles permet de réduire le temps d'exécution du processus de classification. La stratégie de recherche est effectuée comme suit :

- s'il n'y a pas d'arbre associé au service et au masque du réseau, alors il est nécessaire de récupérer les cas pour le service et le masque du réseau:

$$C_{.im} = f_s(C) = \{C_{j.im} \in C / C_{j.i} = C_{n+1.i}, C_{j.m} = C_{n+1.m}\}$$

où $C_{j.i}$ représente le cas j et i l'identifiant du service.

- la mémoire de règles associée à l'ensemble des cas $R/c.im$ est récupérée, où $R/c.im$ représente les règles associées aux cas appartenant au service i et le masque de sous réseau m .

Phase d'adaptation: elle est basée sur l'approche CART, qui est une méthode statistique non paramétrique pour l'extraction de connaissance dans les classifications. L'information extraite est représentée dans un arbre de décision binaire, qui permet à des individus d'être classifiés à partir du nœud racine. La méthode CART peut être séparée dans deux types :

1. Arbre de classification si la variable dépendante est discrète, et
2. Arbre de régression dans le cas d'une variable dépendante continue.

Phase de révision : une fois que l'ensemble des règles a été recherché, la classification du cas $C_{n+1.im}$ est obtenue en utilisant l'ensemble des règles qui ont été utilisées précédemment lors du classement des éléments du même type. Suite à l'étape de révision on a les situations suivantes :

- le cas est classifié comme bon et la révision se termine ;
- le cas est classé comme suspect et le second mécanisme de classification est exécuté ;
- le cas est non identifié, alors dans ce cas il est nécessaire d'exécuter le RàPC de la deuxième phase (CBRMAS-L2).

3.3.3 Etude comparative des RàPC multi-agents

Après avoir exposé les principaux systèmes du raisonnement à partir de cas basés sur les systèmes multi-agents, nous présentons dans cette partie une étude comparative personnelle de leurs caractéristiques en trois tableaux. Le premier décrit le type du système (RàPC niveau agent ou SMA) et le domaine d'application (Tableau 3.5). Le deuxième tableau contient la représentation du cas, la structure et le type de la base de

cas (Tableau 3.6). Le dernier tableau (Tableau 3.7) prend en compte les critères suivants :

- Remémoration permet de savoir la méthode de recherche des cas sources similaires au cas cible.
- Mesure de similarité : définit des fonctions et formules pour évaluer la similarité entre les cas. Elle est utilisée pour la phase de remémoration.
- Adaptation : définit des heuristiques du domaine, généralement sous forme de règles, permettant de transformer ou de modifier la solution des cas sources remémorés pour s'adapter au cas cible.
- Révision permet d'évaluer la solution proposée et de la corriger si elle est insatisfaisante.
- Apprentissage consiste à enrichir la base des cas, par l'intégration de nouveaux cas.

Systèmes	Type RàPC	Domaine du système	Référence
CBR-TEAM	SMA	Condensateur de vapeur	[Prasad et al., 1995]
FPL	Agent	Purification des protéines	[Plaza et al., 1997]
CICLMAN	Agent	Réallocation des ressources CPU	[Soh et Luo, 2003]
CCBR	Agent	Planification d'itinéraire	[Ginty et Smyth, 2001]
MCBR	SMA	Générique	[Leake et Sooriamurthi, 2001]
ProCLAIM	SMA	Argumentation	[Tolchinsky et al., 2012]
RoBoCats	Agent	RoboCup	[Marling et al., 2003]
AMAL	Agent	Argumentation	[Ontañón et Plaza, 2007]
POMAESS	Agent	E-service	[Yu et al., 2003]
S-MAS	Agent	Service Web	[Pinzón et al., 2011]

Tableau 3.5 : Comparatif du type et du domaine

Systèmes	Représentation	Structure de la base	Type de base
CBR-TEAM	Liste des attributs	Plate	Privée
FPL	Graphe	Hiérarchique	Privée
CICLMAN	Liste des attributs	Plate	Privée
CCBR	Liste des attributs	Plate	Privée
MCBR	Liste des attributs	Plate	Privée
ProCLAIM	Graphe	Hiérarchique	Globale
RoBoCats	Liste des attributs	Plate	Privée
AMAL	Liste des attributs	Plate	Privée
POMAESS	liste des vecteurs	Plate	Privée
S-MAS	Liste des attributs	Hiérarchique	Globale

Tableau 3.6 : Comparatif de l'organisation des cas

Systèmes	Remémoration	Mesure de similarité	Adaptation	Révision	Apprentissage
CBR-TEAM	Recherche par point de vue	Comparaison des valeurs via des formules mathématique	Adaptation générative	Non	Ajout de nouveaux cas
FPL	Recherche par point de vue	Comparaison des historiques de valeurs (prog. dynamiques)	Transformationnelle	Non	Ajout manuel de nouveaux cas
CICLMAN	Recherche par point de vue	Comparaison des historiques de valeurs.	Adaptation générative	Oui	Ajout de nouveaux cas
CCBR	Recherche par point de vue	Distance géométrique	Adaptation générative	Non	Ajout de nouveaux cas
MCBR	Recherche multidimensionnelles	Comparaison des matrices de covariance dynamiques	Adaptation générative	Oui	- Ajout automatique de nouveaux cas -Apprentissage de la structure du réseau de neurones
ProCLAIM	Recherche Multidimensionnelles	-Distance ontologique -Distance entre graphe	Adaptation générative	Oui	-Ajout automatique de nouveaux cas - Modification des cas existants
RoboCats	Recherche par point de vue	Comparaison des historiques de valeurs (une mesure de similarité par variable)	Non	Oui	-Ajout automatique de nouveaux cas -Réorganisation des cas existants
AMAL	Recherche Multidimensionnelles	Comparaison des valeurs via des formules mathématiques	Non	Oui	Ajout de nouveaux cas
POMAESS	Recherche par point de vue	Distance euclidienne	Adaptation générative	Oui	-Ajout de nouveaux cas -Réorganisation de la base de cas
S-MAS	Recherche Multidimensionnelles	Distance euclidienne	-Arbre de décision (CBRMAS-L1) -Réseau de neurones (CBRMAS-L2)	Oui	-Ajout de nouveaux cas -Réorganisation des cas existants

Tableau 3.7 : Comparatif du Cycle RaPC pour les différents systèmes

3.4 Conclusion

Les systèmes RàPC fondés sur les SMA, considèrent chaque changement dans la situation dynamique comme un nouveau cas sur lequel il faut ré-appliquer le cycle RàPC classique.

Nous pensons que dans des domaines d'applications dynamiques où la situation traitée change d'une manière dynamique et incrémentale, il est indispensable de prendre en compte ces particularités non seulement dans la représentation des cas et la structure de base de cas mais aussi dans le cycle du RàPC. Dans le chapitre suivant nous présentons notre architecture RàPC multi-agents avec un cas cible dynamique et avec un cycle de raisonnement dynamique.

Chapitre 4 : Système de Raisonnement à Partir de Cas Incrémental Dynamique Multi-Agents : IDCBR-MAS

4.1 Introduction	130
4.2 Architecture globale	131
4.2.1 Cycle dynamique de raisonnement à partir de cas	131
4.2.2 Principe et description du système IDCBR-MAS	134
4.2.3 Couche d'élaboration	137
4.2.3.1 L'agent de requête.....	137
4.2.3.2 L'agent Générateur	137
4.2.3.3 L'agent Traces-C1.....	138
4.2.3.4 Processus d'élaboration de la situation	139
4.2.4 Couche de remémoration.....	140
4.2.4.1 L'agent ILCSS.....	140
4.2.4.2 L'agent Analyseur	140
4.2.4.3 L'agent-Traces-C2.....	141
4.2.4.4 L'agent d'apprentissage	142
4.2.4.5 Processus de remémoration	143
4.2.5 Couche d'adaptation	144
4.2.5.1 L'agent évaluateur	144
4.2.5.2 L'agent-Traces-C3.....	144
4.2.5.3 L'agent d'adaptation	145
4.2.5.4 Processus d'adaptation.....	146
4.3 Cas et base de cas	147
4.3.1 Structure de cas	147
4.3.2 Base de cas	148
4.4 Mesure et tableaux de similarités	149
4.4.1 Mesure de similarité dynamique : ILCSS (Inverse Longest Common SubSequence).....	149
4.4.2 Tableaux de similarités.....	152
4.5 Conclusion.....	152

Certaines parties de ce chapitre ont été publiés dans le journal international IJIMAI [ZOUHAIR et al., 2014b] et le journal international IJARCSSE [ZOUHAIR et al., 2013].

4.1 Introduction

Dernièrement, plusieurs travaux cherchent à repousser les limites du raisonnement à partir de cas statique afin de développer des applications capables de traiter des domaines dynamiques. La complexité des systèmes dynamiques a amené beaucoup de chercheurs à utiliser une modélisation fondée sur le paradigme multi-agents [Ferber, 1995]. Ceci permet aux systèmes RàPCD de surmonter des difficultés rencontrées avec l'utilisation des approches classiques, telle que la représentation temporelle complexe de la situation dynamique. Toutefois la majorité des systèmes RàPC pour les situations dynamiques ne considèrent l'évolution des valeurs du cas cible au fil du temps que comme un nouveau cas cible, par conséquent la recherche des expériences passées similaires à la situation courante recommence au début du cycle à chaque changement de la situation. Donc, la gestion des situations dynamiques utilise le cycle classique du RàPC sans prendre spécifiquement en considération l'évolution des valeurs du cas cible.

À notre connaissance il n'existe pas de RàPC capable de traiter un cas cible dynamique avec un cycle dynamique. Le tableau suivant présente les principaux systèmes de raisonnement à partir de cas statiques/dynamiques classés selon le type du cas cible et du cycle RàPC [Zouhair et al., 2013] .

Cas cible	Cycle RàPC	Systèmes RàPC Classiques	Systèmes RàPC Multiagent	
			RàPC Niveau Agent	RàPC Niveau SMA
Statique	Statique	CHEF [Hammond, 1986], PATDEX [Richter et Wess, 1991], CASELINE [Watson et Marir, 1994], CREEK [Aamodt, 2004], CASEY [Koton, 1988], PROTOS [Bareiss, 1988], RADIX [Corvaisier et al., 1998]	FPL [Plaza et al., 1997], CCBR [Ginty et Smyth, 2001], AMAL [Ontañón et Plaza, 2007]	ProCLAIM [Tolchinsky et al., 2006]
Dynamique	Statique	REBECAS [Rougegrez-Loriette, 1994], AuRA [Likhachev et al., 2002], SAPED [Aich, 2007], SINS [Ram et Santamaria, 1993], CASEP2 [Zehraoui, 2004], BROADWAY [Jaczynski, 1998], SBR [Baptiste, 2011]	CICLMAN [Soh et Luo, 2003], RoBoCats [Marling et al., 2003], POMAESS [Yu et al., 2003], S-MAS [Cristian Pinzón et al., 2011]	MCBR [Leake et Sooriamurthi, 2001], CBR-TEAM [Prasad et al., 1995]
Dynamique	Dynamique			IDCBR-MAS (Notre approche)

Tableau 4.1 : Classement des systèmes RàPC

Dans le cadre de nos travaux de recherche nous avons défini plusieurs objectifs à atteindre:

- la proposition d'un cycle de raisonnement RàPC dynamique capable de traiter les situations dynamiques ;
- la proposition d'une nouvelle mesure de similarité capable de traiter les situations dynamiques. Le succès du raisonnement à partir de cas dépend étroitement de la mesure de similarité utilisée ;
- la contribution, des SMA en tant que concept adapté à la réalisation d'un RàPC dynamique.

Nous proposons dans cette thèse une architecture distribuée fondée sur le raisonnement à partir de cas et le système multi-agents pour l'aide à la prise de décision dans le cadre d'un RàPC dynamique multi-agents. En tenant compte de la complexité, dynamisme et la distribution des systèmes dynamiques, nous proposons un cycle RàPC dynamique capable de gérer ce type de situation. Nous proposons aussi une nouvelle mesure de similarité capable de rechercher les cas sources similaires aux cas cible. La plateforme proposée permet :

- de collecter l'ensemble des données (les traces) de la situation à prédire ;
- d'offrir des mécanismes d'identification dynamique des cas sources (traces anciennes) similaires au cas cible (traces en cours) ;
- d'être capable de collaborer avec tout type de plateformes ;
- d'assurer la synchronisation des interactions entre les différents acteurs de notre système par la proposition de solutions de communication et la résolution de tout conflit entre les agents de notre système en temps réel ;
- d'enrichir la mémoire du système par l'acquisition incrémentale de nouvelles expériences. Cette acquisition permet l'ajout de nouveaux cas jugés pertinents dans la base de connaissance du système.

L'objectif de ce chapitre est de présenter notre plateforme intitulé IDCBR-MAS (*Incremental Dynamic Case Based Reasoning- Multi-Agent System*) pour les systèmes RàPC dynamiques. Cette plateforme utilise un système multi-agents pour implémenter un raisonnement à partir de cas dont le cycle est dynamique. En premier, nous décrivons notre proposition d'un cycle RàPC dynamique en étoile au lieu de cycle RàPC linéaire classique et nous détaillons les processus internes de ses différentes étapes. Ensuite nous présentons une description de notre plateforme IDCBR-MAS à travers ses couches d'agents d'élaboration, de remémoration et d'adaptation. Nous présentons la structure de cas et de base de cas utilisées dans notre plateforme. Nous terminons en expliquant la mesure de similarité et les tableaux de similarités.

4.2 Architecture globale

4.2.1 Cycle dynamique de raisonnement à partir de cas

Actuellement le modèle utilisé pour décrire le cycle du RàPC dans les cas statique et dynamique est le modèle R^5 de [Finnie et Sun, 2003], [Mille, 2006], qui est une

version étendue (avec l'ajout de la première étape) du modèle R^4 introduit par [Aamodt et Plaza, 1994]. Ce modèle R^5 cyclique est illustré par la figure 4.1.

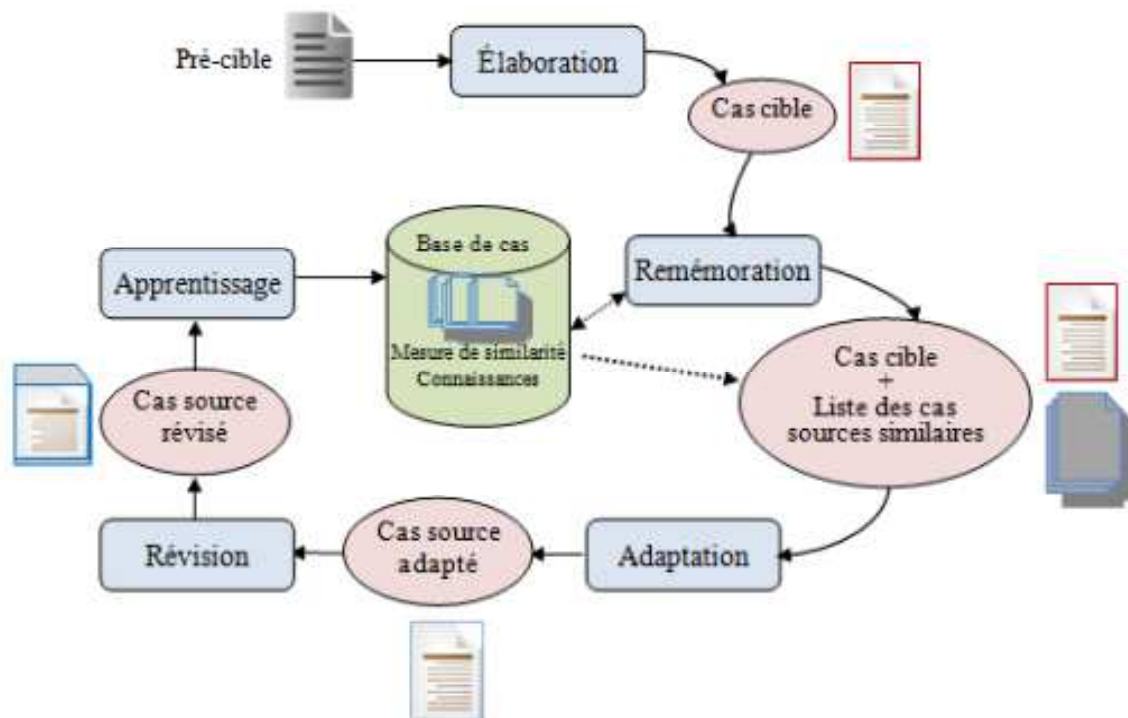


Figure 4.1 : Cycle classique R^5 du RàPC inspiré de [Aamodt et Plaza, 1994]

Nous rappelons que la première phase du cycle R^5 l'élaboration du cas cible consiste à compléter et/ou filtrer la description d'un problème en se basant sur les connaissances du domaine. La seconde phase la remémoration du cas similaire consiste à rechercher dans la base de cas le ou les cas sources les plus similaires au cas cible en se basant sur la mesure de similarité. Ensuite l'étape d'adaptation permet de proposer une solution au cas cible, dans le nouveau contexte propre au cas cible, par la réutilisation de la solution du cas source remémoré dans la phase précédente. La phase de révision permet d'évaluer la solution proposée à l'issue de la phase d'adaptation, et enfin la phase d'apprentissage permet d'ajouter le nouveau cas dans la base de cas.

Le cycle RàPC pour les situations dynamiques utilisées jusqu'à présent est fort semblable à celui adopté par les systèmes RàPC pour situations statiques. Ce modèle du cycle a montré ses limites pour le traitement de situation dynamique. En partant du modèle linéaire, nous proposons donc un nouveau modèle en étoile (figure 4.2). Nous proposons aussi une nouvelle mesure de similarité dynamique, dans une démarche de rechercher les anciennes expériences similaires à la situation en cours d'une façon dynamique. Dans ce qui suit, nous présentons notre modèle dynamique de cycle RàPC.

Nous modifions le modèle R^5 afin qu'il soit capable de traiter les situations dynamiques ce qui amène des changements au niveau de l'ordre des étapes et au niveau du contenu de ces étapes. Notre modèle englobe de nombreux sous-processus spécifiques, car le cycle que nous proposons doit prendre en considération le changement de la situation d'une manière dynamique et incrémentale. Dans notre

système l'ordre des étapes du cycle RàPC n'a pas d'importance : la prédiction de la situation par notre système peut arrêter le cycle RàPC à la phase d'adaptation ou de remémoration et retourner à une étape précédente suite à un changement dans la situation en cours. Donc l'ordre élaboration--> remémoration --> adaptation --> révision --> apprentissage peut subir un changement et certaines étapes peuvent être ré-exécutées plusieurs fois jusqu'à ce que le changement dans la trace cible n'entraîne pas un changement dans la liste des expériences passées considérées comme très similaires à la situation en cours. Nous passons d'un cycle de raisonnement RàPC linéaire à un cycle RàPC en étoile comme le montre la figure 4.2.

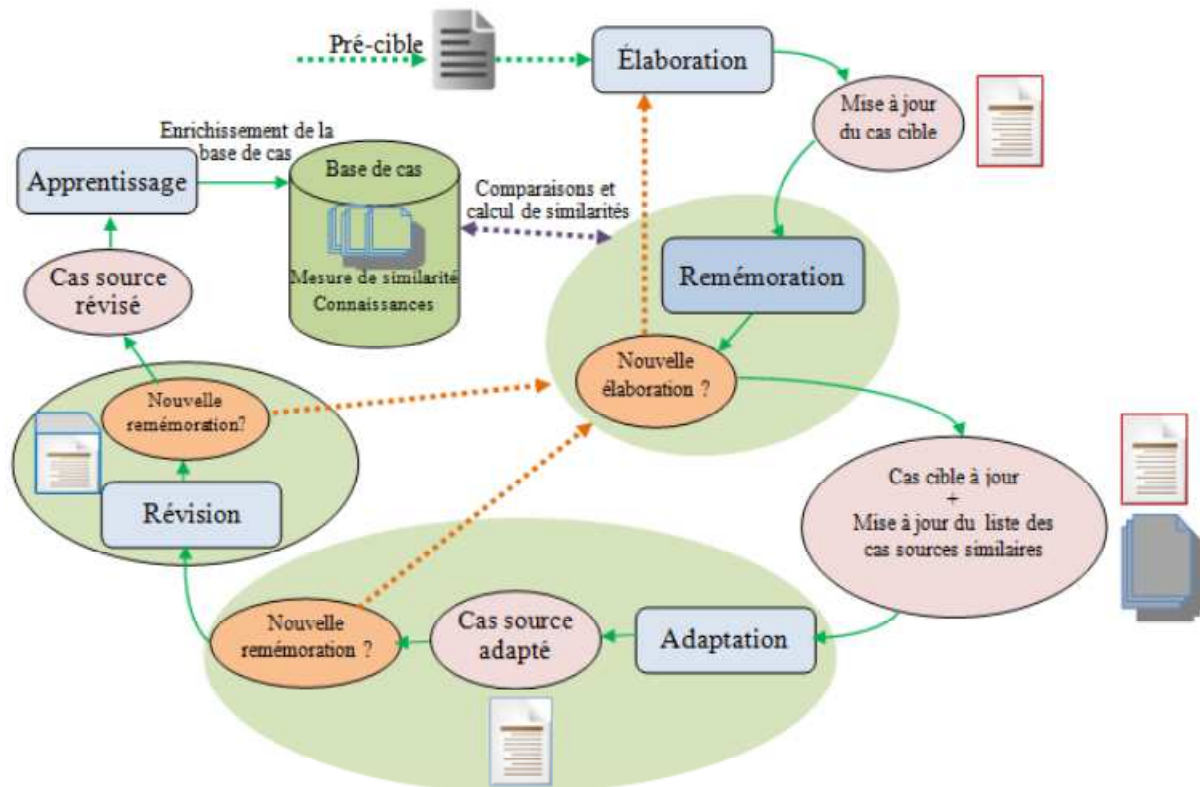


Figure 4.2 : Cycle RàPC dynamique dans le système IDCBR-MAS [Zouhair et al., 2014b]

Le cycle dynamique est une adaptation du cycle classique dans laquelle les changements dynamiques de la situation en cours sont pris en compte. C'est une approche dynamique de l'enchaînement des étapes du cycle RàPC. Le cycle de vie du cycle RàPC dynamique peut être divisé en deux étapes principales :

Étape d'initialisation : cette phase comprend l'élaboration du pré-cas cible en un cas cible initial : un cas cible est tout d'abord élaboré à partir d'une description de situation pré-cible. Ensuite, le cycle se poursuit par la remémoration des cas sources les plus proches au cas cible : le système utilise la mesure de similarité ILCSS (détaillée en fin de ce chapitre) pour trouver les cas sources les plus proches au cas cible, pour ensuite les stocker dans une liste des cas similaires. Donc à la fin de cette étape on aura un cas cible initial et une liste initiale des cas sources similaires au cas cible.

Étape dynamique : cette étape est constituée de plusieurs sous-processus. La détection du premier changement dans la situation pré-cible provoque le démarrage de cette phase. Dans ce cas et à chaque changement dans la situation en cours, c'est-à-dire changement dans le pré cas cible, une **boucle de remémoration** est alors activée comme suit :

- la ré-exécution de l'élaboration du pré-cas cible afin de mettre à jour le cas cible ;
- la ré-exécution de l'étape remémoration en se basant sur la mesure de similarité ILCSS. L'objectif ici est de retrouver dans la base des cas le(s) cas source(s) le(s) plus proche(s) après la mise à jour du cas cible pour disposer d'une mise à jour de la liste des cas sources similaires.

Cette boucle de remémoration se termine dès qu'il n'y a pas de changement dans le pré cas cible et le processus se poursuit par la phase d'adaptation. Grâce à la nouvelle mesure de similarité dynamique ILCSS, on ne refait pas les calculs depuis le début et donc nous comparons juste les données qui viennent d'être ajoutées à la situation à prédire avec la partie correspondante dans les expériences passées. Notre approche prend en considération les résultats enregistrés dans les calculs précédents. Par conséquent notre cycle prend en considération le changement dynamique et incrémental de la situation cible. Puis le cycle passe à l'étape d'adaptation.

Pour l'étape d'adaptation, un cas source sera choisi parmi les cas sources les plus proches remémorés dans la phase précédente comme prédiction de la situation en cours. Après l'exécution de cette étape, deux situations sont alors possibles :

- pas de changement dans le pré cas cible : dans ce cas le cycle passe à l'étape de révision ;
- changement dans le pré cas cible, dans ce cas le cycle retourne à la phase de remémoration.

L'étape de révision consiste à vérifier et s'assurer que le cas choisi est le cas source le plus proche du cas cible. Pendant l'exécution de cette étape, on aura l'une des situations suivantes :

- changement dans le pré-cas cible et dans ce cas le cycle retourne à la boucle de remémoration ;
- pas de changement dans le pré-cas cible et dans ce cas le cas cible sera ajouté dans la base de cas comme nouveau cas source afin d'enrichir la mémoire du système.

Le cycle dynamique est un processus continu. L'idée principale est de permettre d'arrêter l'exécution du cycle à un moment donné dans une étape et de ré-exécuter une étape antérieure suite au changement dans la situation cible.

4.2.2 Principe et description du système IDCBR-MAS

Notre objectif est de développer un outil de raisonnement capable de construire de manière autonome ses connaissances, dans le but d'anticiper une situation dynamique

fondée sur les expériences passées. La prédiction doit être effectuée dans un environnement où les compétences et les connaissances sont distribuées.

Nous nous proposons un système de raisonnement à partir de cas dynamique incrémental IDCBR-MAS. Notre plateforme IDCBR-MAS est un système ouvert, capable de réutiliser les expériences passées. L'architecture générale de notre système est présentée par la figure 4.3. IDCBR-MAS permet de rechercher les anciennes situations similaires à la situation en cours en se basant sur :

- un cycle dynamique de raisonnement à partir de cas ;
- une mesure de similarité capable de rechercher les cas sources très similaires à la situation en cours, qui s'évalue d'une façon dynamique et incrémental.

L'architecture de notre système est constituée d'un ensemble d'agents en relations les uns avec les autres, assurant chacun des tâches et rôles spécifiques. En ce qui concerne la réactivité et la cognition, nos agents auront des stimuli réactifs (transfert de messages, calcul de similarité entre des traces suite à la réception d'un message) ainsi que des stratégies de décisions. Chacun de ces agents, cherche à coopérer et collaborer avec les autres pour prédire une nouvelle situation. Nos agents seront aussi capables de s'adapter et d'apprendre de leur environnement, en particulier, ils apprennent et visent à construire une base des cas afin de bien coordonner leurs décisions. Pour agentifier notre approche, plusieurs contraintes ont été prises en compte :

- chaque étape du cycle RàPC doit être associée à au moins un agent ;
- plusieurs agents d'interfaces ont été définis pour communiquer avec la plateforme sources des traces ;
- un agent est dédié au calcul de similarité entre le cas cible et les cas sources ;
- des agents de coordinations et vérifications ont été définis dans notre architecture ;
- notre architecture contient trois couches d'abstraction, organisée selon les étapes du RàPC ;
- pour la première couche de notre système et pour chaque utilisateur un agent et un seul est chargé de gérer les traces de l'utilisateur.

Le système IDCBR-MAS doit représenter dynamiquement et en temps réel la trace cible, l'analyser et la rapprocher à des traces passées stockées dans une base de cas afin de prédire la trace cible. La trace prédite sera ajoutée à la base de cas afin d'enrichir la mémoire de notre système. Le rapprochement se base sur des valeurs de similarités stockées dans des tableaux de similarités. Ces tableaux contiennent les valeurs de similarités entre la trace cible et chaque traces passées. Cette notion de similarité consiste en un nombre, représentant à quel point la trace cible et les traces passées sont semblables. Elles sont calculées par les agents du système IDCBR-MAS. Nous proposons une plateforme générique (figure 4.3). Elle peut être interfacée avec n'importe quelle source de traces. Nous décrivons dans la section suivante les principaux composants de la plateforme IDCBR-MAS et leurs rôles. Le noyau de notre système multi-agents est un système d'aide à la décision structuré en trois couches. Dans la partie suivante nous présentons ces dernières.

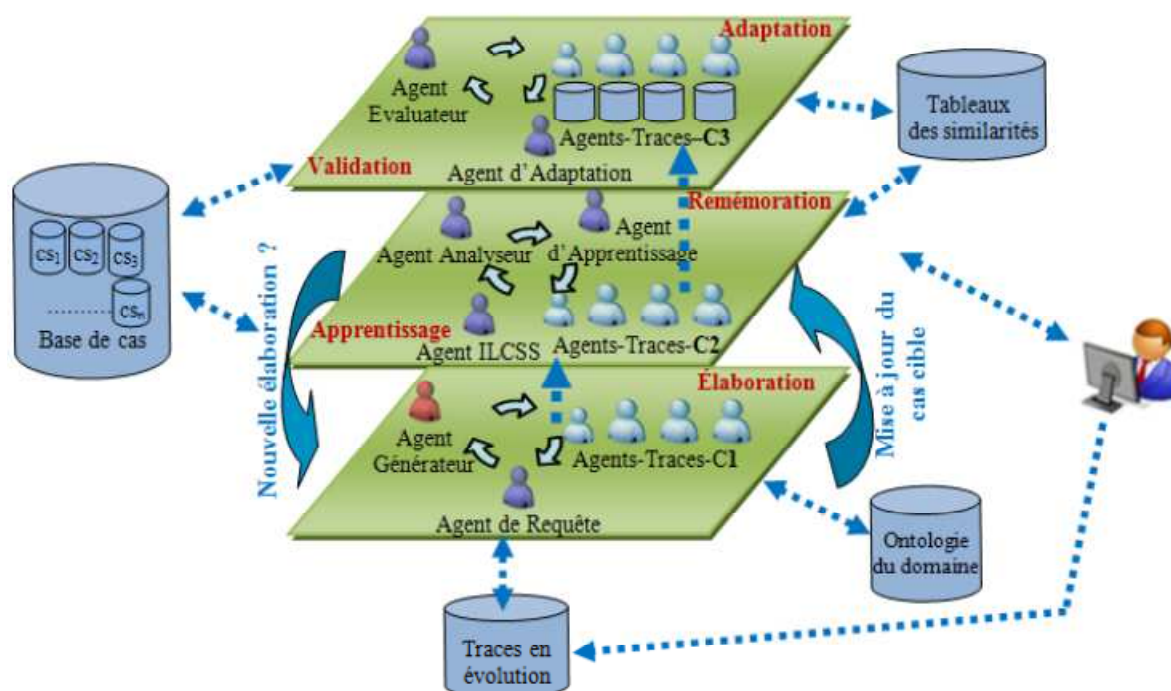


Figure 4.3 : Architecture générale du système IDCBR-MAS

Le rôle des agents de la couche d'élaboration est de gérer les informations venant de l'environnement est de représenter dynamiquement et en temps réel les informations décrivant la situation courante. Cette couche est alimentée par les traces laissées par un utilisateur en interaction avec une plateforme.

La couche de remémoration contient les agents permettant la comparaison entre la situation en cours et les situations passées stockées dans la mémoire du système. La comparaison dans notre plateforme est un processus continu d'évaluation de la ressemblance entre la situation courante et ses scénarios similaires. En effet le changement de la situation courante (suite à l'arrivée de nouvelles informations au système) peut changer la liste des scénarios choisis en tant qu'expériences passées les plus similaires à la situation en cours.

Le rôle des agents de la couche d'adaptation est de prédire la situation en cours par l'adaptation des expériences passées choisies par la couche de remémoration et de vérifier et d'évaluer le choix de ces expériences.

La couche d'élaboration contient deux agents uniques : Agent de requête et Agent générateur. Elle contient l'agent multiple Agent-Traces-C1 : à chaque nouveau cas cible, il est associé un et un seul Agent-Traces-C1.

La couche de remémoration contient trois agents uniques : l'agent Analyseur, l'agent ILCSS et l'agent d'apprentissage. Elle contient l'agent multiple Agent-Traces-C2 : à chaque nouveau cas cible, il est associé un et un seul Agent-Traces-C2.

La couche d'adaptation contient deux agents uniques : l'Agent Évaluateur et l'Agent d'Adaptation. Elle contient l'agent multiple Agent-Traces-C3 : à chaque nouveau cas cible, il est associé un et un seul Agent-Traces-C3.

4.2.3 Couche d'élaboration

La couche d'élaboration contient les trois agents suivants : Agent de requête, Agent-Traces-C1, et Agent générateur.

4.2.3.1 L'agent de requête

Le rôle de l'Agent de Requête (AR) est d'établir le lien entre le système et l'environnement. Il alimente le système en informations, recueillies à partir des fichiers de traces. L'un des buts de cet agent est de vérifier s'il y a un changement dans le fichier traces. Le comportement de l'agent de requête est illustré dans la figure suivante.

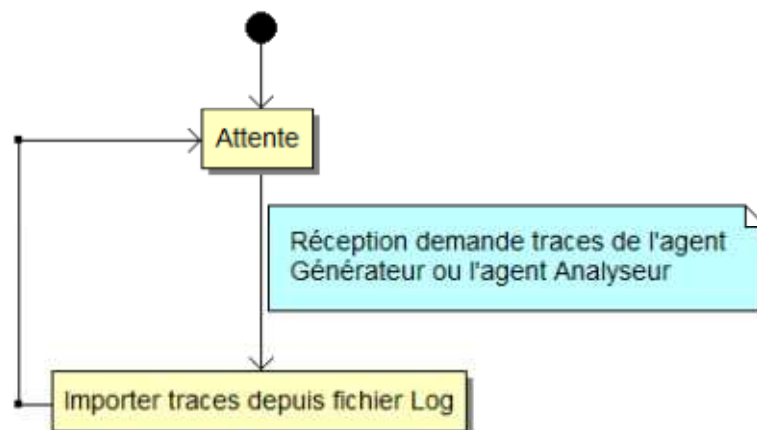


Figure 4.4 : Comportement de l'agent de requête

Au début l'agent de requête est en état d'écoute, il attend les demandes de traces de l'agent générateur ou de l'agent analyseur, ensuite l'agent cherche les traces dans le fichier Log du serveur. Enfin les traces seront envoyées à l'agent générateur.

4.2.3.2 L'agent Générateur

Le but de cet agent est de créer et de mettre à jour les agents de traces existants (un nouvel agent trace est créé si aucun des agents existants ne s'intéresse à la dernière trace interceptée par l'agent générateur, sinon, l'agent qui détient cette trace se met à jour). La figure suivante présente le comportement de l'Agent Générateur (AG).

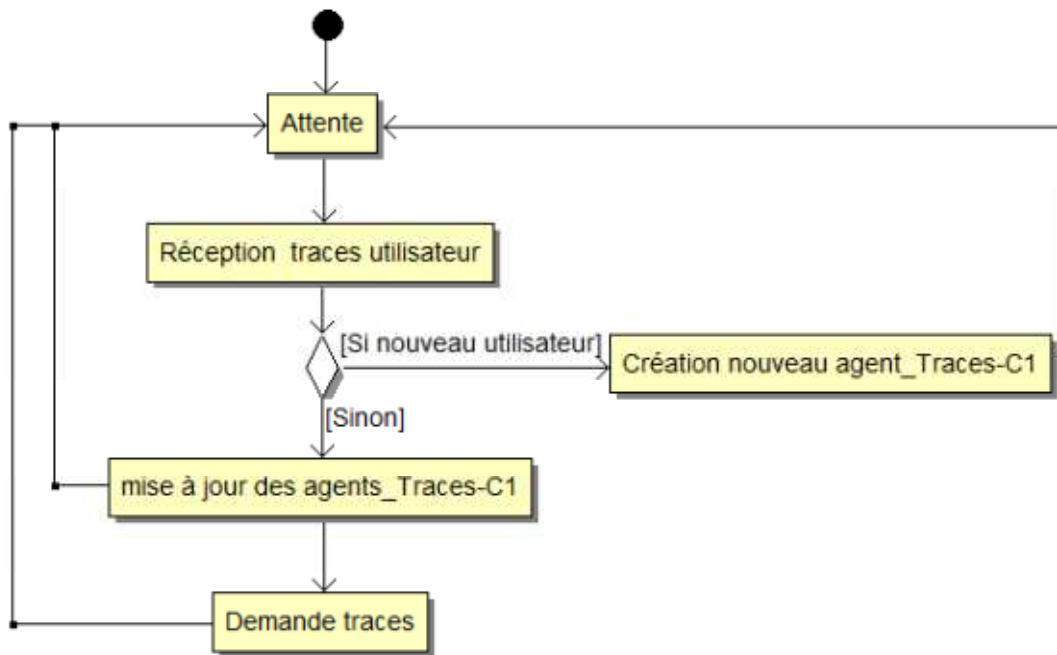


Figure 4.5 : Comportement de l'agent Générateur

Au départ l'agent Générateur est en état d'écoute, il attend les traces de l'agent de requête, ensuite l'agent générateur vérifie s'il s'agit d'une nouvelle trace et dans ce cas l'agent générateur crée un nouvel Agents-Traces-C1. Dans le cas contraire l'agents-traces-C1 concerné par la trace sera mis à jour.

4.2.3.3 L'agent Traces-C1

Les traces transmises au système par l'agent de requête, sont envoyées à l'agent générateur. Ce dernier se charge de la création des Agent-Traces-C1 (Agent traces de la couche 1) (ATC1). Le comportement de cet agent est illustré dans la figure suivante.

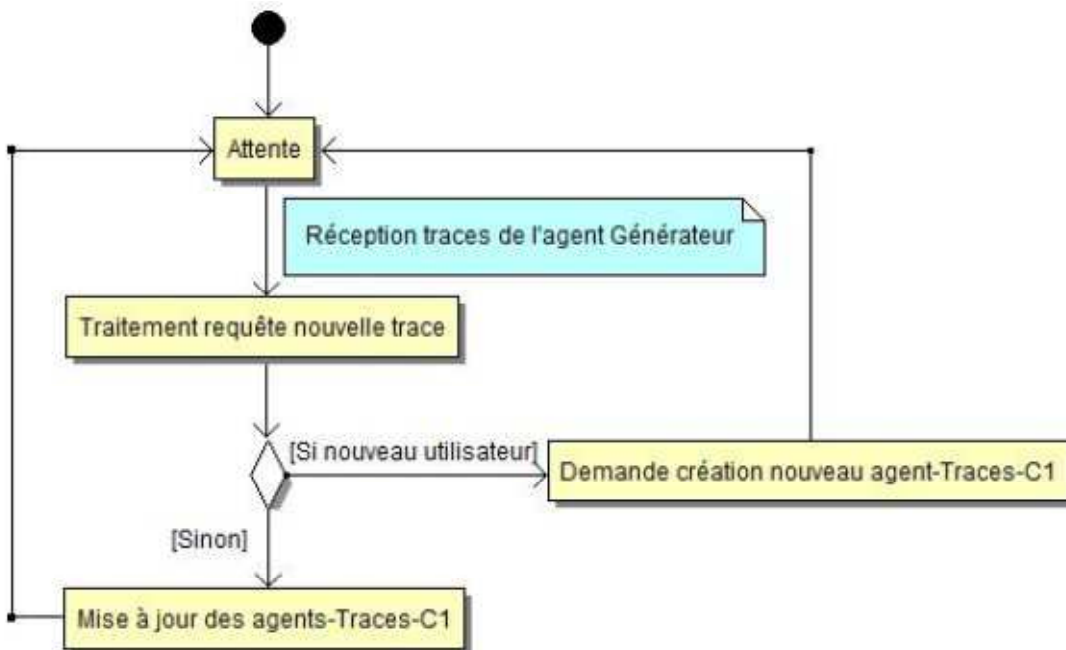


Figure 4.6 : Comportement de l'agent-Traces-C1

Au début l'Agent-Traces-C1 est en état d'écoute, il attend les traces de l'agent générateur : si ces dernières concernent l'un des agents-traces-C1 alors ce dernier sera mis à jour sinon un nouvel agent-Traces-C1 sera créé.

Dans la partie qui suit, nous présentons les différentes interactions entre les agents de la couche d'élaboration.

4.2.3.4 Processus d'élaboration de la situation

L'élaboration de la situation est réalisée par la couche d'élaboration qui contient l'agent de requête, les agent-Traces-C1 et l'agent générateur. L'utilisateur laisse les traces de ses interactions, productions et toutes ses actions sur le serveur dans le fichier Log. L'agent de requête récupère ces traces depuis le serveur et les transmet à l'agent générateur. Ce dernier vérifie si c'est un nouvel utilisateur, et dans ce cas, l'agent générateur crée une instance de l'agent-Traces-C1. Dans le cas contraire, l'agent générateur transmet les nouvelles traces de l'utilisateur à l'agent-Traces-C1 concerné afin de se mettre à jour. L'agent de requête refait ce travail d'une façon périodique et à chaque fois qu'il détecte un changement dans les fichiers traces de l'utilisateur concerné, ceci en accédant au fichier log du serveur des traces et en lisant le contenu de ce dernier. La figure suivante illustre le diagramme de séquence d'élaboration de la situation en cours de progression.

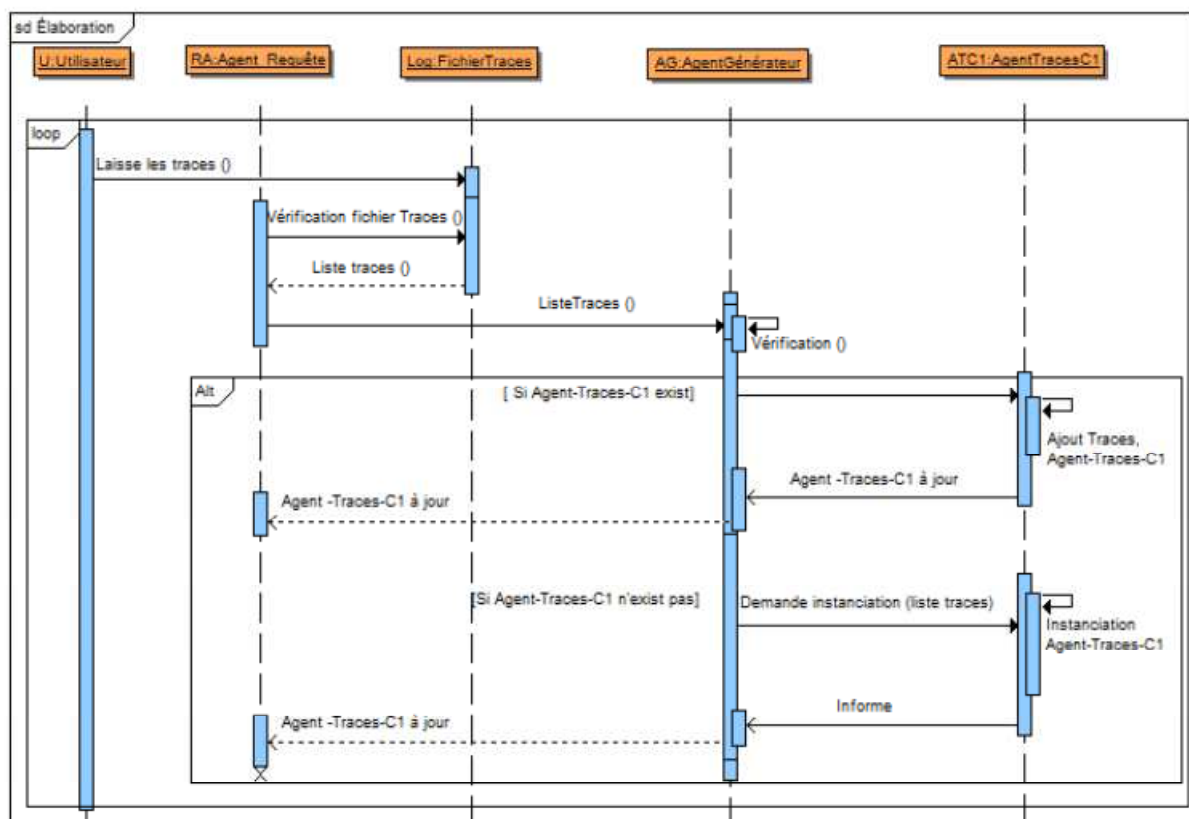


Figure 4.7 : Diagramme de séquence d'élaboration de la situation

4.2.4 Couche de remémoration

La couche de remémoration contient quatre agents : l'agent Analyseur, l'agent ILCSS, l'agent-Traces-C2 et l'agent d'apprentissage.

4.2.4.1 L'agent ILCSS

Le rôle de cet agent est d'évaluer continûment la similarité entre la situation en cours et les différents scénarios stockés dans la mémoire en se basant sur la mesure de similarité ILCSS (cette mesure sera détaillée en fin de ce chapitre). L'agent ILCSS (AILCSS) stocke les valeurs de similarités calculées entre la situation en cours et les scénarios passés dans la table de similarités. Il est chargé de réévaluer ces valeurs à chaque fois qu'il y a changement dans la situation en cours ou suite à la demande des autres agents. La figure suivante décrit le comportement de l'agent ILCSS.

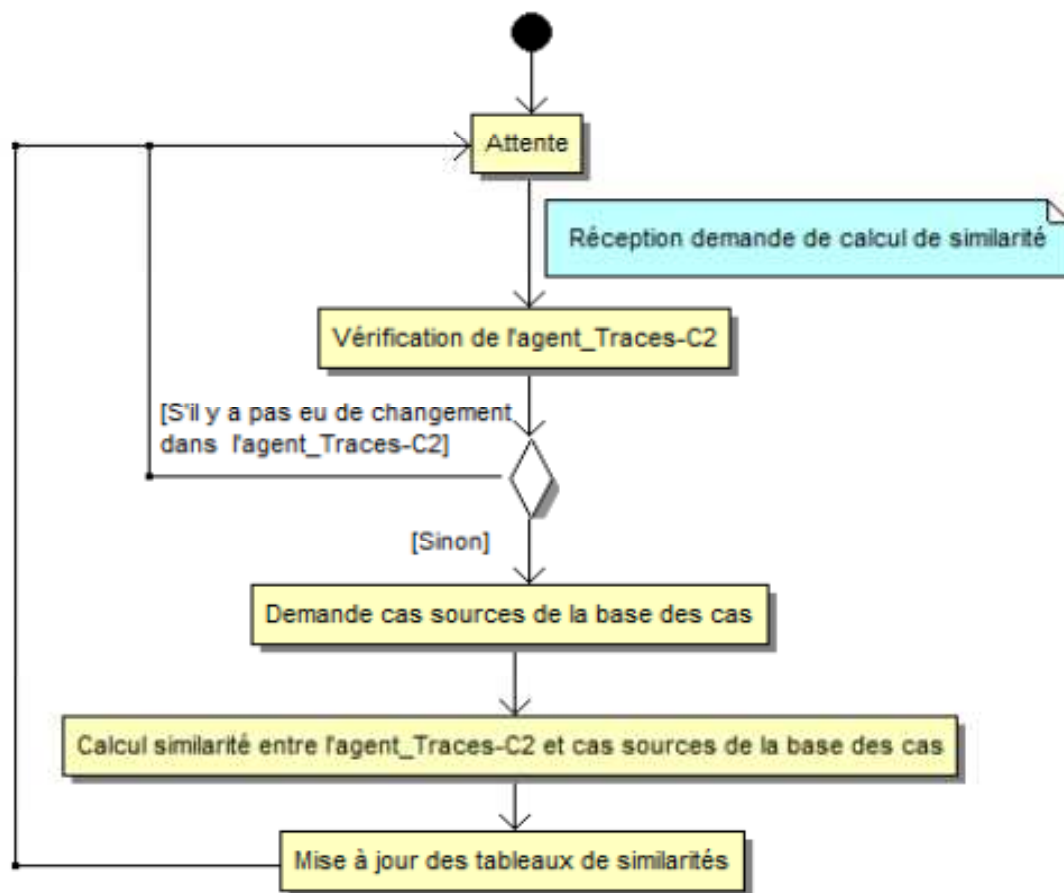


Figure 4.8 : Comportement de l'agent ILCSS

4.2.4.2 L'agent Analyseur

Le but de cet agent est de s'assurer et de vérifier s'il y a changement dans les agents des traces de la deuxième couche suite à l'arrivée de nouvelles informations, ceci d'une manière périodique. L'agent Analyseur (AA) demande à l'agent de requête de la première couche de vérifier s'il y a changement dans le fichier trace. Dans ce cas l'agent analyseur demande à l'agent ILCSS de mettre à jour la table de similarités du cas cible (pour chaque nouveau cas cible le système crée une nouvelle table de

similarités afin de stocker tous les valeurs de similarités entre ce nouveau cas cible et les traces passées). Le comportement de cet agent est défini dans la figure suivante.

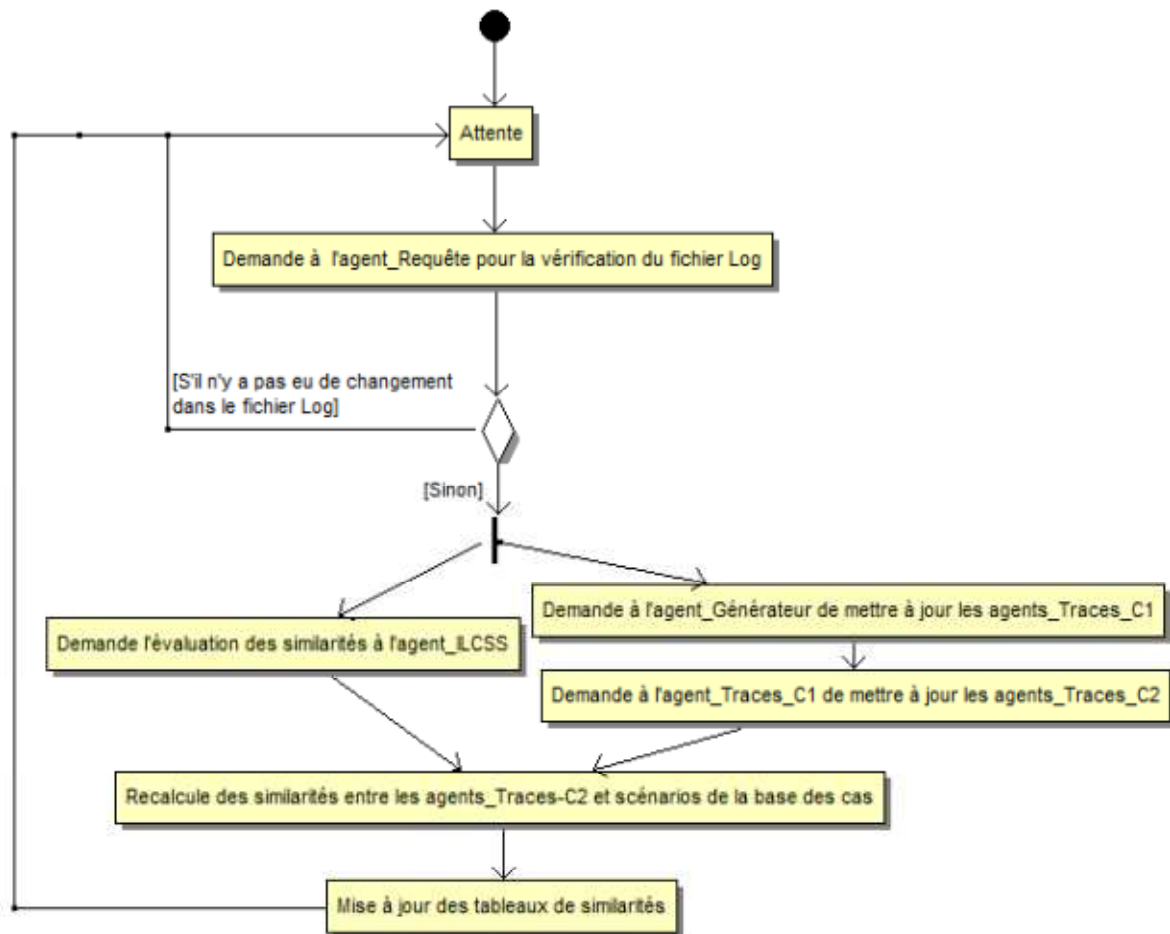


Figure 4.9 : Comportement de l'agent Analyseur

4.2.4.3 L'agent-Traces-C2

Ces agents (un agent par cas cible) contiennent les mêmes informations et données que celles des Agents-Traces-C1 de la première couche. L'agent-Traces-C2 (ATC2) diffère par une abstraction de la trace en cours, décrit originellement par l'Agent-Traces-C1, pour rendre les séquences comparables aux traces passées stockées dans la mémoire. Pour le paramètre temps d'actions, il s'agit d'exprimer les variations entre deux mesures successives plutôt que leurs valeurs absolues. De plus, on applique un filtrage pour éliminer les données supplémentaires telles que l'adresse IP du l'ordinateur de l'utilisateur et nom de l'utilisateur dans les traces initiales de l'Agent-Traces-C1. Nous présentons dans la figure suivante le comportement de cet agent.

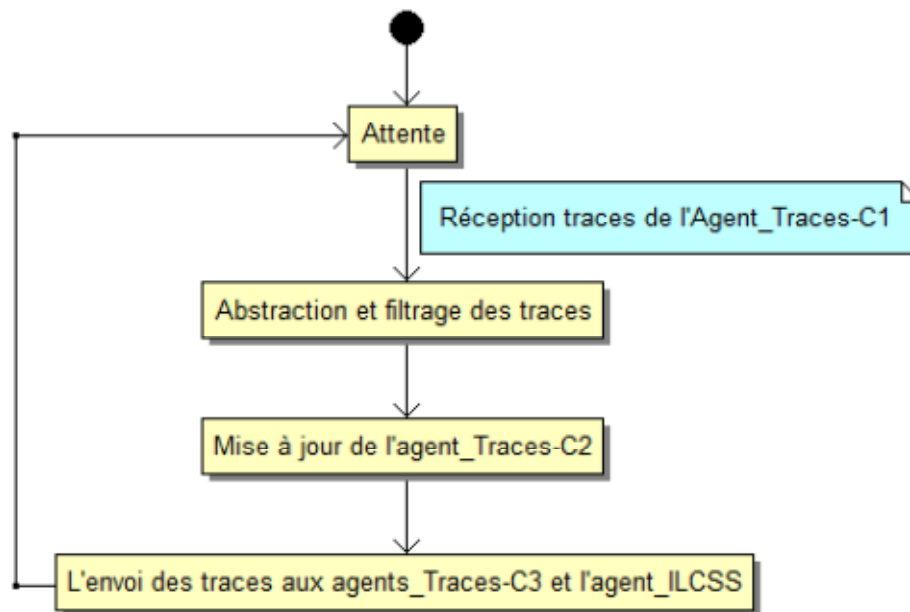


Figure 4.10 : Comportement de l'Agent-Traces-C2

4.2.4.4 L'agent d'apprentissage

Cet agent permet d'ajouter la nouvelle trace dans la base des cas. Le comportement de l'agent d'apprentissage (AA) est illustré dans la figure suivante.

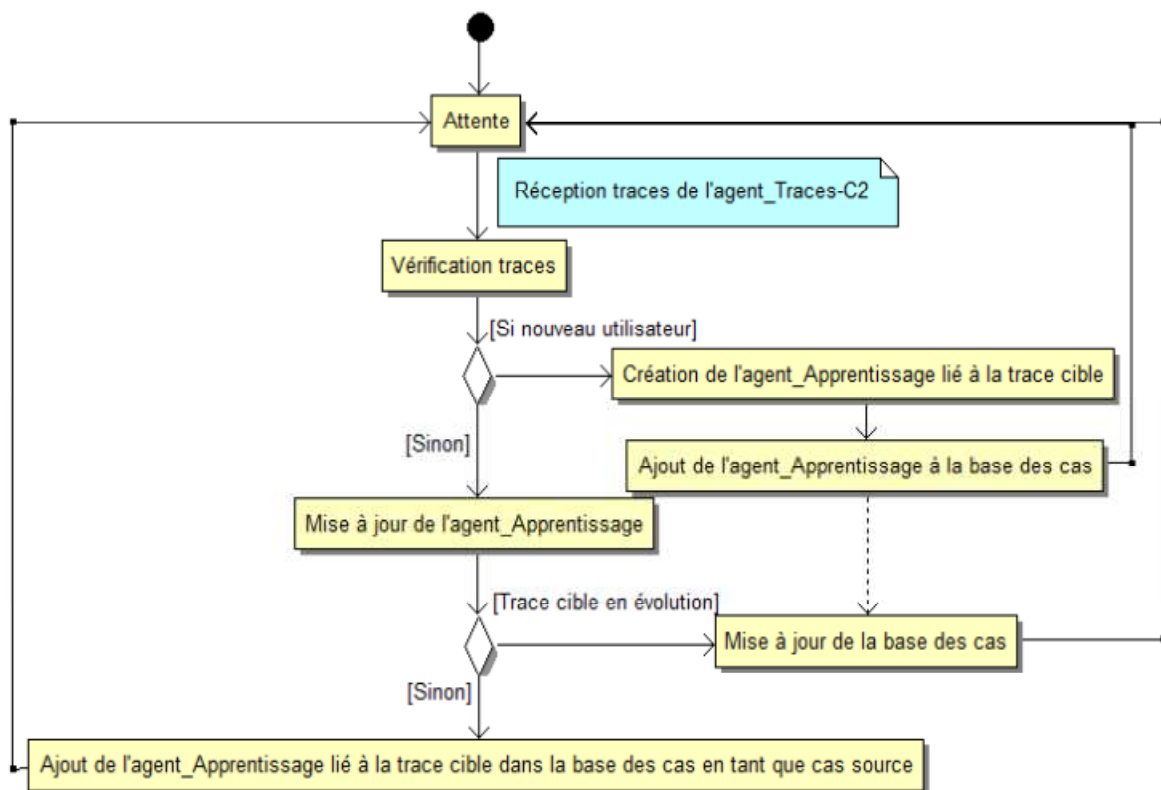


Figure 4.11 : Comportement de l'agent d'apprentissage

4.2.4.5 Processus de remémoration

La phase de remémoration est une phase primordiale dans notre approche. Lors du déclenchement de cette phase, l'agent analyseur initialise la recherche des expériences similaires à la situation en cours. La remémoration de la situation signifie trouver parmi les anciennes traces stockées dans la mémoire celles qui sont très similaires à la situation en cours en se basant sur la mesure de similarité ILCSS. Le diagramme de séquence suivant illustre le processus de remémoration de la situation en cours au sein de la plateforme IDCBR-MAS.

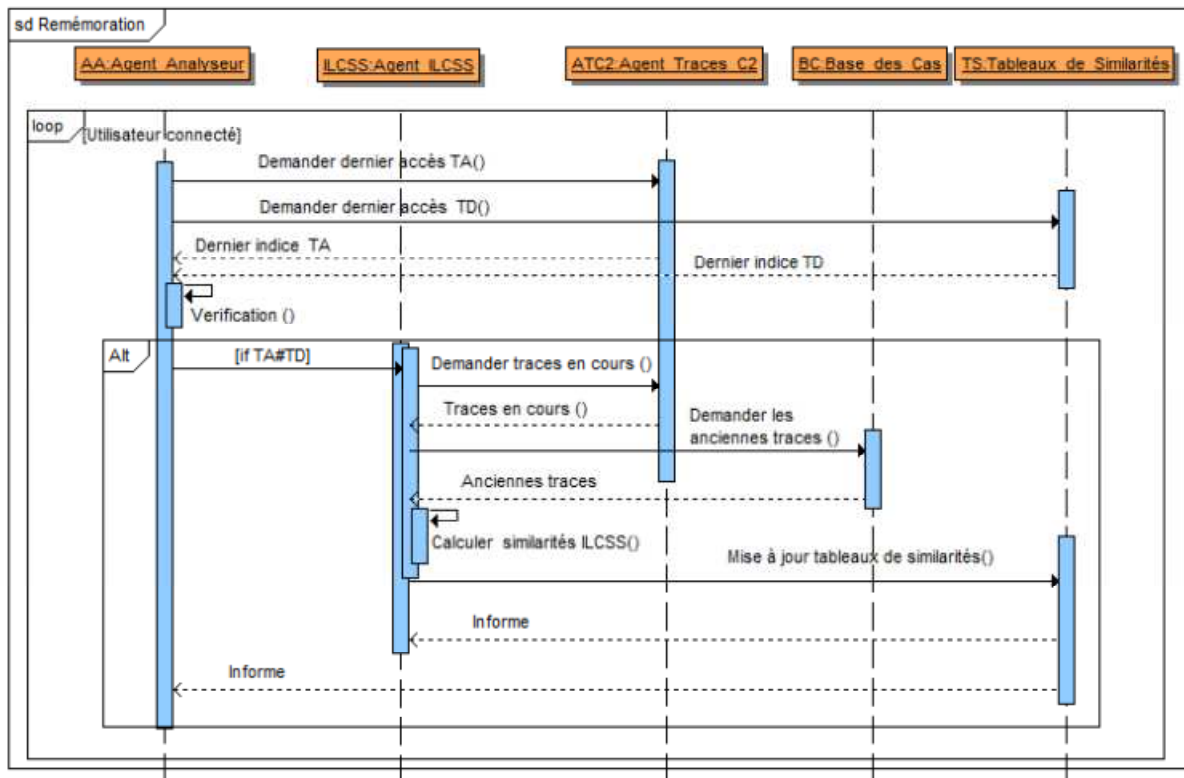


Figure 4.12 : Diagramme de séquence de remémoration de la situation

L'agent analyseur récupère les deux dates chronologiques TA et TD (TA : date de la dernière mise à jour dans le fichier traces, TD : date de la dernière mise à jour dans la table des similarités, au départ ces deux dates sont initialisées à 0). Si la table des similarités entre la situation en cours et les expériences stockées dans la mémoire n'est pas à jour ($TA \neq TD$) alors l'agent Analyseur demande à l'agent de requête de vérifier le fichier des traces en cours et de mettre à jour les agents générateur, agent-Traces-C1 et l'agent-Traces-C2, ensuite il demande à l'agent ILCSS de recalculer les similarités entre la situation en cours (Agents-Traces-C2) et les expériences stockées dans la mémoire. Les valeurs de similarités calculées seront stockées dans la table de similarités (pour chaque itération on enregistre les valeurs de similarités calculées et l'instant de calcul TD).

4.2.5 Couche d'adaptation

Cette couche contient les trois agents suivants : agent Evalueur, Agent-Traces-C3 et l'Agent d'Adaptation.

4.2.5.1 L'agent évaluateur

Le but de cet agent est d'évaluer la solution proposée et de s'assurer que la similarité entre la situation en cours et les expériences passées choisies par la couche de remémoration est suffisante, sinon cet agent demande l'intervention de l'agent ILCSS. La figure ci-après présente le comportement de l'agent Evalueur (AE).

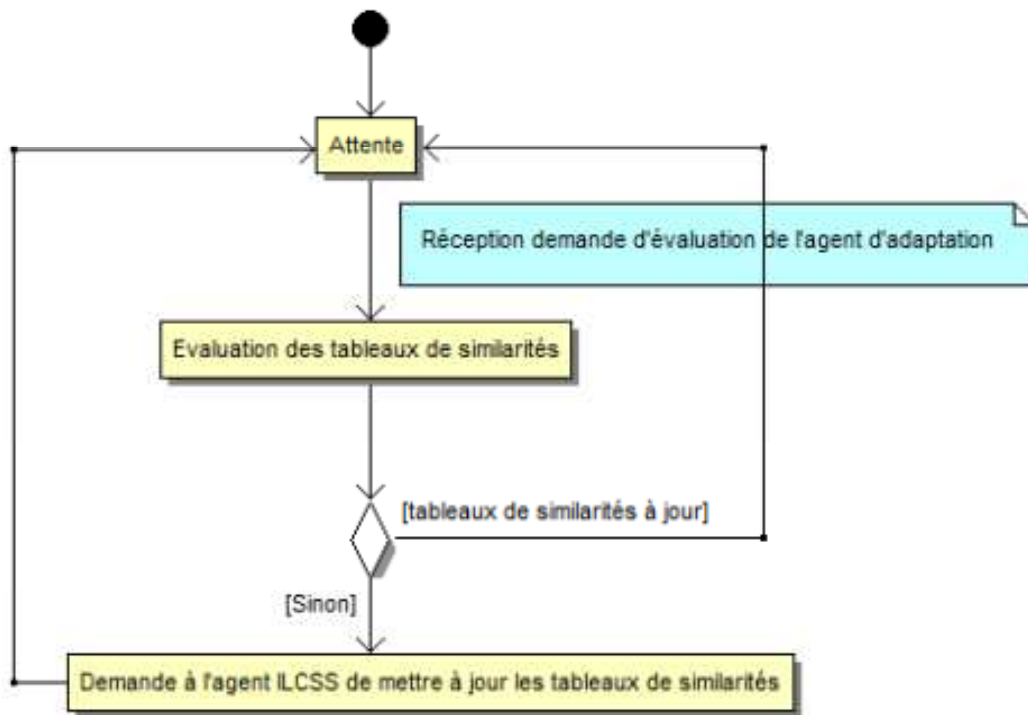


Figure 4.13 : Comportement de l'agent Evalueur

4.2.5.2 L'agent-Traces-C3

Le rôle de l'agent-Traces-C3 (ATC3) est de réaliser une liaison dynamique entre la trace cible et les traces passées similaires. Cette liaison est assurée par un *pointeur* qui contient la liste des expériences passées similaires à la situation en cours à un instant donné. Un pointeur est un objet variable contenant la liste des traces similaires à la trace cible. Il évolue au cours du temps, c'est-à-dire qui permet d'accéder à d'autres objets (les nouvelles traces similaires au cas cible après la ré-exécution de l'étape de remémoration). Chaque agent-Traces-C3 a son propre pointeur qui contient la liste des expériences similaires à ce dernier. L'avantage d'un pointeur c'est que la liste qu'il contient n'est pas statique et elle change d'une manière dynamique suite au changement des agents de traces, ceci en se basant sur la similarité calculée entre la situation en cours et les expériences passées. Donc il se peut qu'une telle expérience passée soit dans la liste à un moment donné et qu'après un certain temps l'expérience

perd sa place dans la liste des cas similaires au cas en cours de progression. Le comportement de cet agent est illustré dans la figure suivante.

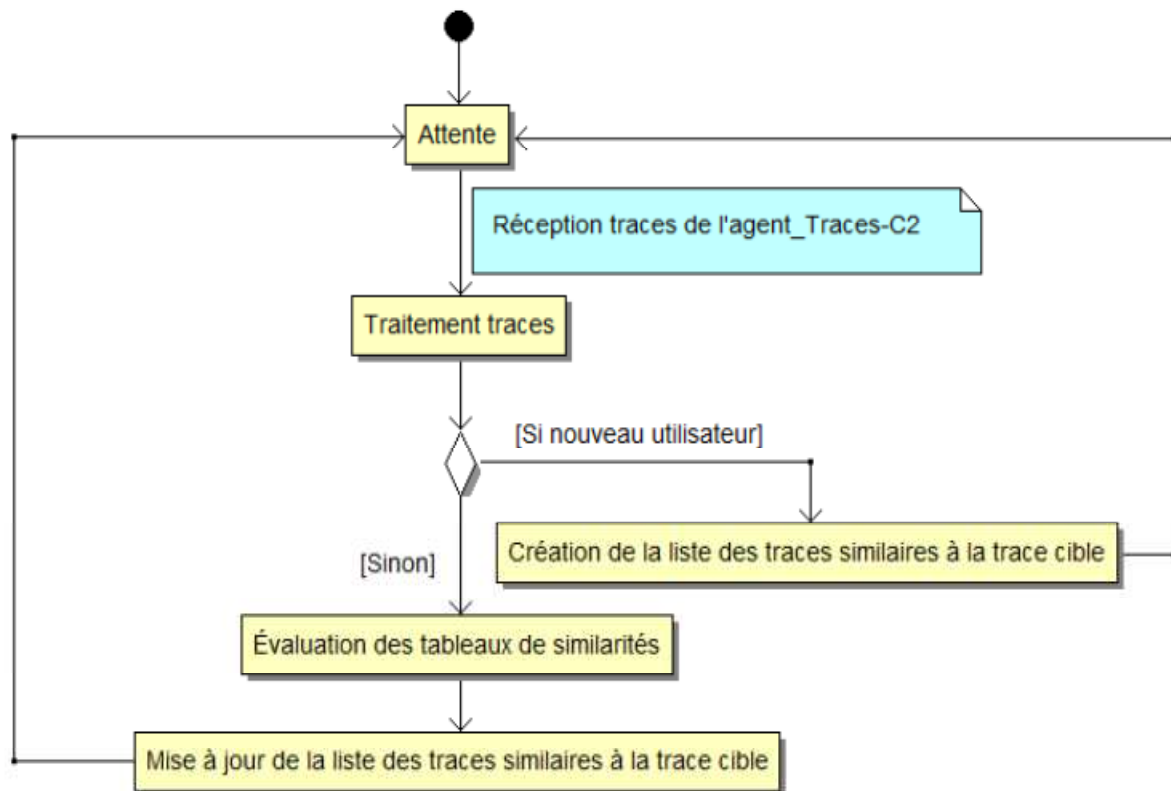


Figure 4.14 : Comportement de l'agent-Traces-C3

4.2.5.3 L'agent d'adaptation

Le rôle de l'agent d'adaptation (AAD) est de sélectionner une ou plusieurs traces similaires à la trace en cours. Ceci en se basant sur la liste des traces similaires de l'agent-Traces-C3. Après le choix d'une ou plusieurs traces similaires, l'agent propose une réponse adaptée à la situation en cours. Le comportement de cet agent est défini dans la figure suivante.

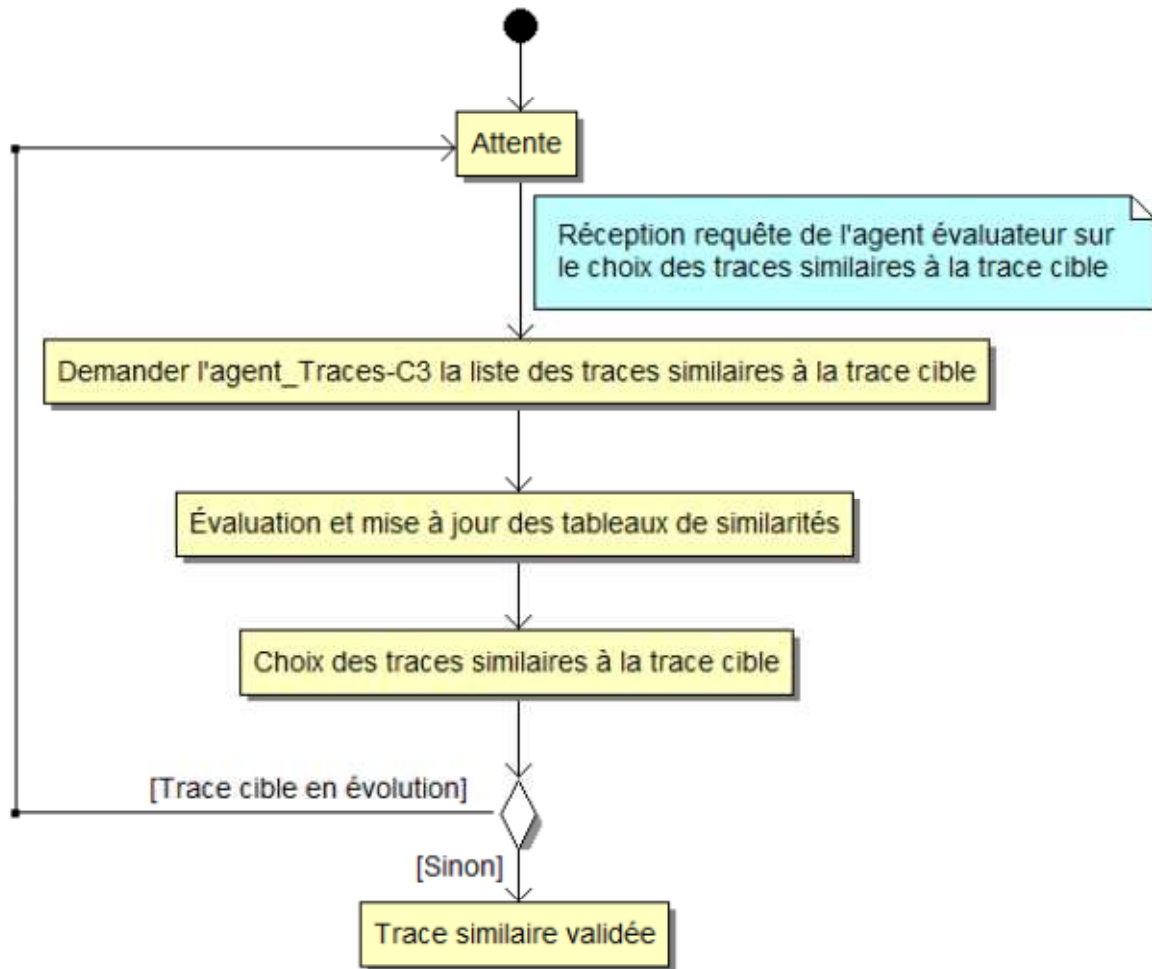


Figure 4.15 : Comportement de l'agent d'adaptation

4.2.5.4 Processus d'adaptation

L'adaptation de la situation consiste à trouver les expériences passées les plus similaires à la situation en cours de progression. L'agent d'adaptation demande à l'agent-Traces-C3 la liste des expériences passées similaires à la trace en cours (on rappelle que cette liste contient les traces jugées très similaires à la trace en évolution, en se basant sur la mesure de similarité ILCSS). Ensuite cette liste sera transmise à l'agent évaluateur afin de vérifier les similarités calculées entre la situation en cours et les traces de la liste, ceci en consultant la table de similarités et en cas de besoin en demandant à l'agent ILCSS de recalculer ces valeurs de similarités. Le résultat de cette étape de remémoration est une liste de traces ordonnées par ordre croissant selon leurs similarités à la trace en cours. Ensuite cette liste sera transmise à l'agent d'adaptation. Ce dernier procède à la validation de la trace future appropriée à la situation en cours de progression. Une fois que la validation est effectuée, l'agent d'adaptation affiche le résultat sur la plateforme IDCBR-MAS. Le diagramme de séquence suivant illustre le processus d'adaptation de la situation en cours de progression au sein de la plateforme IDCBR-MAS.

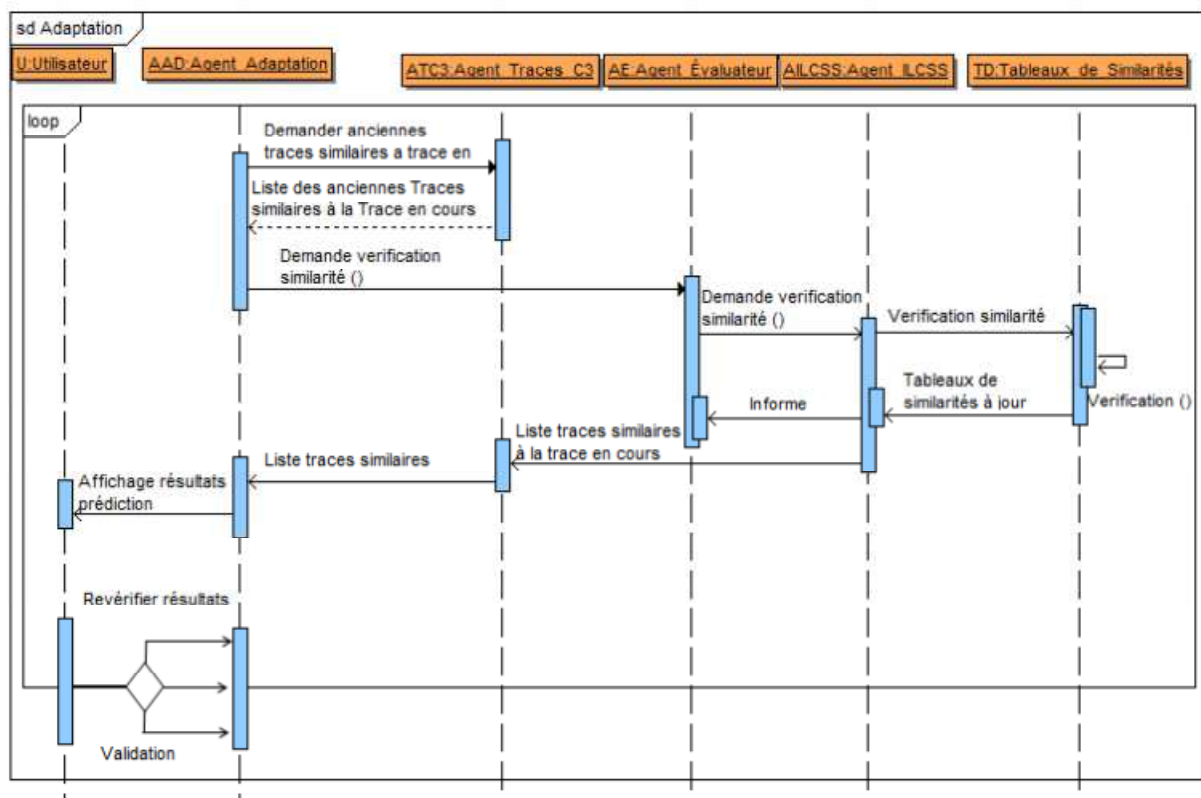


Figure 4.16 : Diagramme de séquence d'adaptation de la situation

4.3 Cas et base de cas

4.3.1 Structure de cas

Les traces des utilisateurs (les cas), désignent les informations qu'un dispositif numérique enregistre sur l'activité, la production, et toute action résultante de l'interaction de ses utilisateurs. Dans notre contexte les traces de l'utilisateur (TA) sont représentées par une collection des traits sémantiques $TA = U \text{ TS}$. Un trait sémantique (TS) constitue les données élémentaires atomiques d'information de la situation [Boukachour, 2002]. Il est représenté par un triplet : $TS = (\text{objet}, (\text{qualification}, \text{valeur}) +)$. On note objet = O, qualification = Q et Valeur = V, $TS = (O, (Q, V) +)$. Les traces de l'utilisateur forment une séquence multidimensionnelle. Donc une trace peut être vue comme une liste de vecteurs ou une matrice multidimensionnelle.

Soit A une trace de dimensions $n \times d$, alors A peut s'écrire comme suit :

$$A = ((O_{A,1}, (Q_{A,1,1}, V_{A,1,1}), \dots, (Q_{A,1,d}, V_{A,1,d}), (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d})), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d}))) \quad (1)$$

A peut s'écrire aussi sous la forme matricielle suivante :

$O_{A,1}$	$O_{A,2}$	...	$O_{A,n}$
$(Q_{A,1,1}, V_{A,1,1})$	$(Q_{A,2,1}, V_{A,2,1})$	...	$(Q_{A,n,1}, V_{A,n,1})$
...	...	...	...
$(Q_{A,1,d}, V_{A,1,d})$	$(Q_{A,2,d}, V_{A,2,d})$	...	$(Q_{A,n,d}, V_{A,n,d})$

Dans notre système un cas cible ou trace cible représente une trace en progression d'un utilisateur. Une trace A à l'instant $t=t_i$ est une trace multidimensionnelle $(A)_{t=t_i}$ de la forme suivante :

$O_{A,1}$	$O_{A,2}$	...	O_{A,t_i}
$(Q_{A,1,1}, V_{A,1,1})$	$(Q_{A,2,1}, V_{A,2,1})$	...	$(Q_{A,n,1}, V_{A,t_i,1})$
...	...	...	...
$(Q_{A,1,d}, V_{A,1,d})$	$(Q_{A,2,d}, V_{A,2,d})$	...	$(Q_{A,t_i,d}, V_{A,t_i,d})$

Les cas sources sont des traces des anciens utilisateurs et peuvent être représentées par la formule (1).

4.3.2 Base de cas

La performance d'un système RàPC est mesurée par son temps de réponse nécessaire pour proposer une solution à un problème donné. Elle dépend de l'organisation de la mémoire et de la performance de la méthode de remémoration. Dans un système à mémoire plate, les cas sont rangés séquentiellement dans une simple liste. L'un des avantages de ce type d'organisation est que le processus d'apprentissage de nouveaux cas est simple, il s'agit d'un simple ajout du nouveau cas résolu à la mémoire. Cependant, sa grande limitation est que le temps de recherche augmente linéairement avec la taille de la base de cas. Pour ces raisons, nous avons opté pour une organisation hiérarchique pour notre base des cas présentée dans la figure 4.17.

Dans le système IDCBR-MAS, les traces passées sont stockées dans une structure hiérarchique contenant plusieurs niveaux. Le bas niveau est une simple mémoire plate organisée en groupes de cas similaires. Le haut niveau contient des index ou des clusters représentant les caractéristiques communes d'un groupe de cas similaires stockés dans le bas niveau. Dans notre base de cas chaque index est composé de deux parties : une partie statique contenant les objets et une autre dynamique contenant les qualifications : la composante comportementale d'un utilisateur à un instant donné. Cette composante comportementale référence un instant précis dans la production des traces de l'utilisateur. Elle regroupe l'ensemble d'activité, de production, et toute action résultante de l'interaction de l'utilisateur avec le serveur à un instant donné.

La base de cas du système IDCBR-MAS est présentée par la figure suivante :

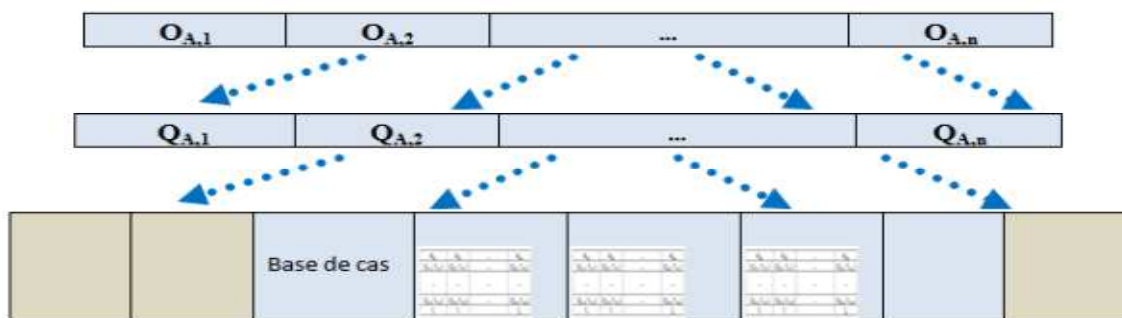


Figure 4.17 : Structure globale de la base de cas en système IDCBR-MAS

Parmi les étapes clef d'un RàPC citons la remémoration des cas similaires au cas cible et la mesure de similarité utilisée. Cependant, ces mesures proposées et utilisées dans les RàPC pour les systèmes dynamiques ne permettent pas de comparer un cas cible en cours de progression avec des cas sources de la base des cas. Nous proposons une mesure de similarité capable de surmonter les limites des autres mesures. Dans la section suivante nous décrivons notre proposition de mesure de similarité, qui est une extension de la mesure de la plus longue sous-séquence commune (*Longest Common SubSequence LCSS*), notre mesure est intitulée ILCSS (*Inverse Longest Common SubSequence LCSS*).

4.4 Mesure et tableaux de similarités

4.4.1 Mesure de similarité dynamique : ILCSS (Inverse Longest Common SubSequence)

Notre objectif est d'étendre l'approche LCSS à la comparaison de séquences temporelles, multidimensionnelles, hétérogènes et dynamiques, c'est-à-dire au moins l'une de ces séquences est en cours de progression. La mesure de similarité $ILCSS_{\delta,N}$ que nous proposons diffère de celle de LCSS [Vlachos et al., 2002] à plusieurs niveaux :

1. La mesure de similarité LCSS permet de comparer deux séquences statiques : elle commence cette comparaison en commençant d'abord par la comparaison des derniers points. Dans notre contexte, nous avons besoin de commencer la comparaison des premiers vecteurs des séquences temporelles et dans le même ordre sur l'axe du temps, afin de comparer la similarité entre une trace en cours de progression et des traces passées ;
2. Notre proposition $ILCSS_{\delta,N}$ mesure de manière continue la similarité entre les cas sources et le cas cible, en utilisant une formule récursive c'est-à-dire on réutilise les résultats trouvés à l'instant $t=t_i$ dans les calculs pour l'instant $t=t_{i+1}$: la mesure entre la trace cible et les traces sources à l'instant $t=t_{i+1}$ dépend des résultats à l'instant $t=t_i$;
3. Contrainte temporelle supplémentaire est nécessaire pour éviter certains cas particuliers, tels que le cas où une grande partie des vecteurs de trace en cours de progression est similaire à un ensemble de ceux d'une trace passée, sans recouvrement, et dans le même ordre sur l'axe du temps.

Dans ce qui suit, nous présentons notre proposition de mesure de similarité dynamique $ILCSS_{\delta,N}$.

Soient A et B deux traces de dimensions $n \times d$ et $m \times d$ respectivement :

$A = ((O_{A,1}, (Q_{A,1,1}, V_{A,1,1}), \dots, (Q_{A,1,d}, V_{A,1,d}), (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d})), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d})))$;

Et

$B = ((O_{B,1}, (Q_{B,1,1}, V_{B,1,1}), \dots, (Q_{B,1,d}, V_{B,1,d}), (O_{B,2}, (Q_{B,2,1}, V_{B,2,1}), \dots, (Q_{B,2,d}, V_{B,2,d})), \dots, (O_{B,m}, (Q_{B,m,1}, V_{B,m,1}), \dots, (Q_{B,m,d}, V_{B,m,d})))$.

Pour la trace A, soit Tail(A) la trace:

$\text{Tail}(A) = (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d})), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d}))$. Tail(A) est égal à la trace A privée de son premier vecteur.

Le but est de compter le nombre de couples de vecteurs considérés comme similaires lorsqu'on parcourt les deux traces comparées.

La similarité entre deux vecteurs $(V_{A,i,1}, V_{A,i,2}, \dots, V_{A,i,d})$ et $(V_{B,j,1}, V_{B,j,2}, \dots, V_{B,j,d})$ est déterminée selon un seuil δ : un vecteur $(V_{A,i,1}, V_{A,i,2}, \dots, V_{A,i,d})$ d'une trace A est considéré similaire à un vecteur $(V_{B,j,1}, V_{B,j,2}, \dots, V_{B,j,d})$ d'une autre trace B si

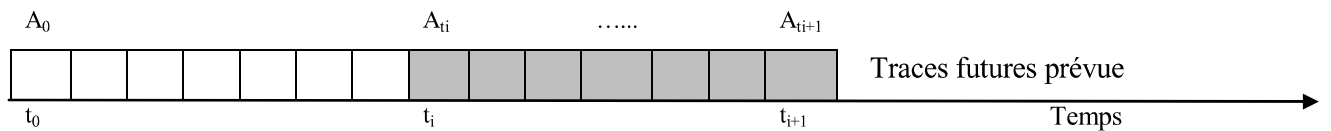
$$|V_{A,i,k} - V_{B,j,k}| \leq \delta \text{ pour chaque } k = 1, \dots, d.$$

On définit aussi un entier N qui sert à contrôler l'écart de temps maximum entre deux vecteurs de chacune des traces pour qu'ils puissent être similaires.

Soient les deux traces (séquences temporelles) (A, B), un réel δ et un nombre entier N. On définit la mesure de similarité entre les deux traces, qu'on note $\text{ILCSS}_{\delta,N}(A, B)$, selon le processus récursif suivant : on initialise le processus par les deux premiers vecteurs des traces (A, B) comparées. Si l'une des deux traces est vide alors la mesure de similarité est nulle et le processus s'arrête. Sinon, si les premiers vecteurs des traces sont similaires, la mesure de similarité vaut 1 plus la similarité calculée entre les deux traces privées de leurs premiers vecteurs ($\text{Tail}(A)$, $\text{Tail}(B)$). Dans le cas où les deux premiers vecteurs des traces A et B ne sont pas similaires, alors la mesure de similarité est égale au maximum de la similarité entre une trace et l'autre privée de son premier vecteur. La formule suivante présente le principe de mesure de similarité entre les deux traces A et B.

$$\text{ILCSS}_{\delta,N}(A, B) = \begin{cases} 0 & \text{si A ou B est vide} \\ 1 + \text{ILCSS}_{\delta,N}(\text{Tail}(A), \text{Tail}(B)) & \text{si } O_{A,i} = O_{B,j}, Q_{A,i,k} = Q_{B,j,k}, \\ & |V_{A,i,k} - V_{B,j,k}| \leq \delta, \text{ avec } 1 \leq i \leq n, 1 \leq j \leq m, |i - j| \leq N \text{ et } k = 1 \dots d. \\ \max\{\text{ILCSS}_{\delta,N}(\text{Tail}(A), B), \text{ILCSS}_{\delta,N}(A, \text{Tail}(B))\} & \text{sinon} \end{cases}$$

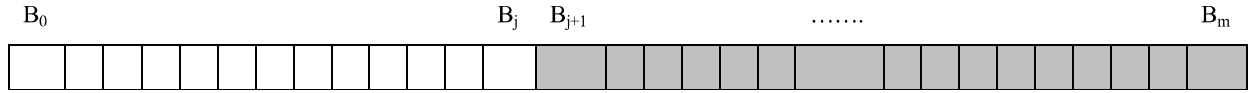
D'une part, à l'instant $t=t_{i+1}$ le système IDCBR-MAS récupère les traces $(A)_{t=t_{i+1}}$ enregistrées dans le serveur entre les deux instants t_i et t_{i+1} et on note $(A)_{t=t_{i+1}} = A_{[t_i, t_{i+1}]}$ voir figure ci-dessous.



Donc $(A)_{t=t_{i+1}}$ est un bloc de traces de n' vecteurs et elle peut s'écrire :

$$A_{[t_i, t_{i+1}]} = ((O_{A[t_{i+1}, t_i]}, (Q_{A[t_{i+1}, t_i]}, 1, 1, V_{A[t_{i+1}, t_i]}, 1, 1), \dots, (Q_{A[t_{i+1}, t_i]}, 1, d, V_{A[t_{i+1}, t_i]}, 1, d), \dots, (O_{A[t_{i+1}, t_i]}, n', (Q_{A[t_{i+1}, t_i]}, n', 1, V_{A[t_{i+1}, t_i]}, n', 1), \dots, (Q_{A[t_{i+1}, t_i]}, n', d, V_{A[t_{i+1}, t_i]}, n', d))))$$

D'autre part, à l'instant $t=t_{i+1}$ il reste que le bloc $B_{[j+1, m]}$ de la trace source B, bloc des traces non encore comparé avec la trace cible, voir figure ci-dessous :



$$B_{[j+1,m]} = ((O_{B[j+1,m],1}, (Q_{B[j+1,m],1,1}, V_{B[j+1,m],1,1}), \dots, (Q_{B[j+1,m],1,d}, V_{B[j+1,m],1,d}), \dots, (O_{B[j+1,m],m',1}, (Q_{B[j+1,m],m',1}, V_{B[j+1,m],m',1}), \dots, (Q_{B[j+1,m],m',d}, V_{B[j+1,m],m',d}))) \text{ avec } m' = m-j.$$

La mesure entre la trace cible A et la trace source B à l'instant $t=t_{i+1}$ dépend des résultats à l'instant $t=t_i$ via la formule récursive suivante :

$$(ILCSS_{\delta,N,t_{i+1}}(A,B))_{t=t_{i+1}} = (ILCSS_{\delta,N,t_i}(A,B))_{t=t_i} + ILCSS_{\delta,N,t_{i+1}}(A_{[t_i,t_{i+1}]}, B_{[j+1,m]})$$

Avec :

$$ILCSS_{\delta,N,t_{i+1}}(A_{[t_i,t_{i+1}]}, B_{[j+1,m]}) = \begin{cases} 0 & \text{si } A_{[t_i,t_{i+1}]} \text{ ou } B_{[j+1,m]} \text{ est vide.} \\ 1 + ILCSS_{\delta,N,t_{i+1}}(\text{Tail}(A_{[t_i,t_{i+1}]}), \text{Tail}(B_{[j+1,m]})) & \text{si } O_{A_{[t_i,t_{i+1}]},i'} = O_{B_{[j+1,m]},j'}, \\ & Q_{A_{[t_i,t_{i+1}]},i',k'} = Q_{B_{[j+1,m]},j',k'}, \left| V_{A_{[t_i,t_{i+1}]},i',k'} - V_{B_{[j+1,m]},j',k'} \right| \leq \delta \\ & , \text{ avec } 1 \leq i' \leq n', 1 \leq j' \leq m', m' = m-j, |i' - j'| \leq N \text{ et } k' = 1 \dots d. \\ \max\{ILCSS_{\delta,N,t_{i+1}}(\text{Tail}(A_{[t_i,t_{i+1}]}), B_{[j+1,m]}), ILCSS_{\delta,N,t_{i+1}}(A_{[t_i,t_{i+1}]}, \text{Tail}(B_{[j+1,m]}))\} & \text{sinon.} \end{cases}$$

L'indice du dernier élément commun entre les deux traces $(A)_{t=t_{i+1}}$ et B à l'instant $t=t_{i+1}$ est obtenu via l'algorithme suivant :

On initialise j à 0, puis on utilise la formule suivante :

$$\{j = j' \text{ si } O_{A_{[t_i,t_{i+1}]},i'} = O_{B_{[j+1,m]},j'}, Q_{A_{[t_i,t_{i+1}]},i',k'} = Q_{B_{[j+1,m]},j',k'}, \left| V_{A_{[t_i,t_{i+1}]},i',k'} - V_{B_{[j+1,m]},j',k'} \right| \leq \delta, \text{ avec } 1 \leq i' \leq n', 1 \leq j' \leq m', |i' - j'| \leq N, j' > j \text{ et } k' = 1 \dots d. \}$$

La similarité $\text{sim}_{\delta,N,t_i}(A,B)$ entre les deux traces A et B à l'instant t_i , est ensuite normalisé par la plus petite longueur des deux traces comparées, de façon à ce que la mesure de similarité varie entre 0 et 1 :

$$\text{Sim}_{\delta,N,t_i}(A,B) = 1 - \frac{ILCSS_{\delta,N,t_i}(A,B)}{\min(n,m)}$$

$A \approx B$ se lit les deux traces A et B sont similaires.

Le point fort de notre système IDCBR-MAS est que le processus de calcul de similarité ne s'arrête pas s'il y a un changement dans la trace de l'utilisateur et on ne refait pas le calcul depuis le début. Les similarités calculées sont sauvegardées dans les tableaux de similarités, détaillé dans la section suivante.

4.4.2 Tableaux des similarités

Notre plateforme IDCBR-MAS récupère les traces via les agents d'interfaces d'une manière permanente, ensuite ces dernières sont traitées par notre système. Le tableau suivant présente les valeurs de similarités calculées par les agents IDCBR-MAS en temps réel, entre la trace cible et les traces passées. Nous signalons que pour chaque nouveau cas cible le système crée une nouvelle table de similarités afin de stocker tous les valeurs de similarités entre ce nouveau cas cible et les traces passées. Les valeurs de similarités sont mises à jour automatiquement chaque fois qu'il y a changement dans le fichier trace.

Similarité ILCSS	Score élaboration	Score remémoration	Score adaptation	TD

Tableau 4.2 : Similarités entre trace cible et traces passées

Les similarités entre la trace cible et les traces passées sont calculées par l'agent ILCSS :

- **Similarité ILCSS** représente la similarité entre trace cible et traces passées ;
- **Score d'élaboration** représente le nombre de fois où la trace a changé de contenu ;
- **Score de remémoration** montre le nombre de fois que l'agent ILCSS a recalculé ces valeurs de similarités suite au changement des données dans la trace cible. Ces similarités serviront comme élément fondamental dans l'adaptation de la situation gérée par l'agent d'adaptation ;
- **Score adaptation** présente le nombre de fois où le système a changé la liste des traces similaires à la trace cible ;
- **TD** est la date de la dernière mise à jour dans le tableau des similarités.

4.5 Conclusion

Nous avons présenté dans ce chapitre un système de raisonnement à partir de cas dynamique multi-agents. L'idée consiste à adapter les traces passées des utilisateurs afin de prédire la suite d'une nouvelle trace en progression. Nous proposons la migration du cycle RàPC d'une architecture linéaire statique à une architecture en étoile dynamique : s'il y a changement dans la trace cible, notre système peut arrêter l'exécution du cycle et retourner à l'étape précédente. De plus, à l'instant t_{i+1} on réutilise les résultats calculés à l'instant t_i . Le cycle de raisonnement proposé est basé sur une mesure de similarité dynamique capable de comparer la similarité entre une trace en cours de progression et des traces passées : il mesure de manière continue la similarité entre les cas sources et le cas cible. La plateforme IDCBR-MAS est totalement indépendante de ses applications. Le chapitre suivant sera consacré au développement de la plateforme et à la présentation d'un exemple du fonctionnement plus complet du système.

Chapitre 5 : Implémentation de IDCBR-MAS : Système Tutoriel Intelligent fondé sur les traces de l'apprenant

5.1 Introduction	154
5.2 E-Learning et les Systèmes Tutoriels Intelligents	155
5.2.1 E-Learning.....	155
5.2.2 Les Systèmes Tutoriels Intelligents.....	156
5.3 Les plateformes de formation à distance	157
5.3.1 Définition.....	157
5.3.2 Moodle.....	159
5.4 Systèmes Tutoriels Intelligents : notre objectif.....	161
5.5 Système Tutoriel Intelligent et RàPCD	162
5.5.1 Création du cours (JADE) sous Moodle.....	163
5.5.2 Interface Moodle/ IDCBR-MAS	164
5.5.3 Trace de l'apprenant.....	166
5.5.3.1 Trait sémantique	166
5.5.3.2 Cas source.....	167
5.5.3.3 Cas cible	170
5.5.4 Base de cas	171
5.6 Développement des agents IDCBR-MAS	173
5.6.1 Aspects techniques	173
5.6.1.1 Motivation à l'utilisation de la plateforme JADE	173
5.6.1.2 Outils de contrôle et de supervision des agents dans IDCBR-MAS	173
5.6.1.3 Communication inter-agents dans IDCBR-MAS	174
5.6.1.4 Introspector agent JADE dans IDCBR-MAS	175
5.6.2 Les agents IDCBR-MAS appliqués au STI.....	176
5.6.2.1 Processus d'élaboration d'une situation en cours de progression	176
5.6.2.2 Processus de remémoration de la situation en cours de progression	177
5.6.2.3 Processus d'adaptation de la situation en cours de progression	178
5.6.2.4 Processus de prédiction d'une situation d'apprentissage en cours de progression.....	179
5.6.3 Cycle dynamique IDCBR-MAS appliqués au STI.....	182
5.7 Application Web.....	188
5.7.1 Scénario d'apprentissage.....	188
5.7.2 Génération de similarités entre le cas cible et cas sources	191
5.7.3 Courbe des similarités	192
5.7.4 Le cas source le plus similaire.....	193
5.8 Conclusion.....	194

5.1 Introduction

L'évolution continue des besoins de formation, vers plus d'efficacité, de flexibilité, d'interaction, et avec l'avènement des nouvelles technologies de l'information et de la communication (NTIC) ont favorisé l'émergence d'outils pédagogiques et informatiques appelés e-Learning, dont l'objectif est de faciliter davantage l'accès à des *formations* et de les rendre plus attractives et donc plus facilement assimilables par les apprenants. Le e-Learning est l'introduction des NTIC dans le domaine de la formation. Tous les publics sont visés, étudiants, employés, techniciens, cadres, et il concerne tous les domaines de formation et apprentissage Humain. Le e-Learning, d'une part permet à l'apprenant de travailler où et quand il le souhaite, il lui permet aussi d'avancer à son rythme, de commencer une formation quand il en a besoin, et d'autre part il permet aux enseignants-tuteurs de faire le suivi d'un ou de plusieurs apprenants, de participer et d'animer des activités en ligne, de communiquer avec les apprenants à tout moment, et enfin de développer un travail collaboratif de groupe entre les apprenants.

Le e-Learning est aujourd'hui en plein essor et des avancées ont été réalisées au cours de ces dernières années dans le domaine. Ceci permet par exemple l'interopérabilité des plateformes d'apprentissage, l'évaluation des apprenants en ligne, le lien entre apprenants dans un environnement non supervisé et dispersé géographiquement, et la possibilité pour l'apprenant de choisir un horaire d'apprentissage souple. L'intégration de diverses composantes du processus d'apprentissage et l'adaptation selon le profil de l'apprenant joue un rôle important et déterminant dans le processus d'apprentissage. Malgré ce rôle déterminant de l'adaptation selon le profil de l'apprenant, il y a lieu de noter que jusqu'à aujourd'hui, les fonctionnalités associées à l'adaptation d'apprentissage selon le niveau et la capacité de l'apprenant sont encore insuffisantes, voire non efficaces. Cette adaptation d'apprentissage est généralement fondée sur les styles d'apprentissage [Felder et Silverman, 1988]. Le style d'apprentissage d'un apprenant c'est son mode personnel et sa façon préférée de percevoir, de traiter et d'organiser l'information, de penser et de résoudre des problèmes en situation d'apprentissage. Par exemple certains apprenants apprennent mieux en regardant (style visuel), d'autres en écoutant (style auditif) et d'autres en touchant des objets et en bougeant (style kinesthésique). Il existe des dizaines de styles qui ont chacun leur façon différente de catégoriser les apprenants : particulièrement sur des aspects perceptuels et de traitement de l'information. Coffield et al. [Coffield et al., 2004], ont recensé 71 styles. L'adaptation d'apprentissage est fondée aussi sur les concepts acquis par l'apprenant [Leshin et al., 1992]. De plus les Systèmes Tutoriels Intelligents (STI) développés jusqu'à présent ne présentent pas une solution globale pour tous les domaines de connaissance. Croock, [Croock et al., 2005] présente un état de l'art des systèmes fréquemment utilisés dans les universités, qui montre la difficulté d'exprimer les méthodes indépendamment des contenus.

Notre contribution se situe dans ce contexte. Elle consiste à concevoir et à mettre en place un système informatique adaptatif capable d'initier l'apprentissage et d'assurer

un suivi individualisé de l'apprenant. Le système permet de représenter, suivre et analyser l'évolution d'une situation d'apprentissage afin d'anticiper un éventuel abandon de l'apprenant ou une démotivation de celui-ci. Ce système s'appuie sur le rapprochement entre le parcours de l'apprenant et des parcours (traces) passés stockés dans la base de cas. L'analyse du parcours de l'apprenant doit se faire de manière continue. Ceci peut être accompli en se basant sur le raisonnement à partir de cas et puisque le parcours de l'apprenant est un tracé qui évolue d'une manière permanente au fil du temps alors il s'agit du raisonnement à partir de cas dynamique pour situations dynamiques.

Dans le cadre de cette thèse, nous avons choisi le e-Learning comme domaine d'application pour valider notre architecture IDCBR-MAS. IDCBR-MAS est générique et indépendante de tout domaine d'application. Le e-Learning n'est qu'un domaine d'application parmi d'autres. Notre approche peut être utilisée dans des domaines variés : e-commerce, finance,...

Dans ce qui suit, et tout au long de ce chapitre, nous mettons en relief les différents concepts, modèles et techniques liés à l'implémentation de notre plateforme IDCBR-MAS. Nous présentons des exemples de trait sémantique, traces, cas cible, cas sources, et un exemple de l'application de notre cycle RàPCD sur un cas réel d'un apprenant en formation à distance. Enfin quelques résultats obtenus lors de la validation de notre approche.

5.2 E-Learning et les Systèmes Tutoriels Intelligents

5.2.1 E-Learning

Le **e-Learning**, e-formation, **e-training**, **formation en ligne**, **téléformation**, **enseignement télématique**, ou encore l'apprentissage en ligne, désignent l'ensemble des outils et moyens permettant l'apprentissage par des moyens électroniques.

Selon l'Union Européenne, le **e-Learning** est : " l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance " [AWT, 2014].

Nous citons aussi la définition suivante : Le e-Learning définit tout dispositif de formation qui utilise un réseau local, étendu ou l'internet pour diffuser, interagir ou communiquer, ce qui inclut l'enseignement à distance, en environnement distribué, l'accès à des sources par téléchargement ou en consultation sur le réseau. Il peut faire intervenir du synchrone ou de l'asynchrone, des systèmes tutorés, des systèmes à base d'autoformation, ou une combinaison des éléments évoqués. Ces systèmes permettent de révolutionner l'approche pédagogique, d'employer des méthodes plus ludiques où l'interactivité joue un grand rôle, de diversifier les outils employés, de s'adapter davantage au processus d'apprentissage de l'apprenant, qui devient le pilote de sa formation. Ce dernier pourra se former à son rythme, en fonction de ses besoins et de ses disponibilités [Kreczanik, 2008].

5.2.2 Les Systèmes Tutoriels Intelligents

Les systèmes tutoriels intelligents (STI) sont des systèmes informatiques fournissant un enseignement et un encadrement personnalisés. Ils contiennent des modèles de contenu d'apprentissage qui précisent quoi enseigner en utilisant des bases de connaissances ainsi que des stratégies d'enseignement qui précisent comment l'enseigner [Wenger, 1987]. Les STI sont des outils conçus pour aider et faciliter la tâche d'apprentissage de l'apprenant. En effet ils sont capables de réaliser des inférences sur l'objet d'apprentissage et les activités qui doivent être réalisées par un apprenant afin d'adapter dynamiquement le contenu ou le style de formation proposée. D'après Komi [Komi, 2008], les STI se distinguent des systèmes classiques d'apprentissage dans la mesure où ils connaissent la matière à enseigner (connaissance du domaine), comment l'enseigner (connaissance pédagogique) et aussi comment acquérir des informations pertinentes sur l'apprenant (représentation de l'apprenant). L'interaction entre tous ces composants, permet au STI de fournir un enseignement adapté à un apprenant.

Depuis quelques années, les travaux sur les Systèmes Tutoriels Intelligents s'intéressent à la conception et à la réalisation de systèmes capables de reproduire le comportement d'un tuteur intelligent capable de permettre un apprentissage adapté au rythme et aux capacités de l'apprenant. Il existe de nombreux travaux de recherche qui s'intéressent à la conception et à la réalisation de systèmes informatiques permettant d'assister l'apprenant dans son apprentissage. On trouve par exemple des agents tuteurs ou pédagogiques qui accompagnent l'apprenant en lui proposant des activités de remédiation [Frasson et al., 1998]. On trouve aussi des agents de support à la collaboration de groupe en apprentissage [Constantino et al., 2001] encourageant, quant à eux, la participation des élèves et facilitent la discussion entre eux. On trouve également des solutions basées sur les agents qui intègrent et cherchent à faire coopérer différents Systèmes Tuteurs Intelligents [Brusilovski 1997]. La plateforme Baghera [Webber et Pesty 2002], est un Environnement Informatique pour l'Apprentissage Humain (EIAH) distant, exploite les concepts et les méthodes de l'approche multi-agents. Baghera assiste les élèves dans leur travail de résolution d'exercice en géométrie, et ils peuvent interagir avec d'autres élèves ou enseignants. Les enseignants peuvent connaître l'état d'avancement des travaux des élèves afin d'intervenir si c'est nécessaire. SAAID [Labouidya et al., 2008], Système d'Aide à l'Apprentissage Individualisé en Autoformation à Distance, est un système qui se base sur l'approche Multi-agents dans le but de produire une assistance pour l'apprentissage dans le domaine des télécommunications et des réseaux. HeapMotiv [Derbali, 2013], est un système qui se base sur des éléments motivationnels issus de jeux vidéo et des stratégies motivationnelles d'un modèle théorique de la motivation dans l'objectif de soutenir les apprenants à atteindre des hauts niveaux de motivation, de persévérance et de performance. Enfin, Le Corre [Le Corre, 2013], propose CHRYSAOR, système tutoriel intelligent pour les environnements virtuels d'apprentissage humain. C'est un couplage entre la formation en réalité virtuelle et les Systèmes Tutoriels Intelligents. Les concepts proposés dans ce modèle permettent au formateur de modifier les rôles et

les affectations (humain ou agent autonome) afin de rendre le système modulaire. CHRYSAOR a été appliqué dans une plateforme de réalité virtuelle à un instrument de diagnostic en hémostase utilisé en milieu hospitalier.

Les STI forment un courant particulier du e-learning. Ils ont pour principe la personnalisation de l'apprentissage en s'intéressant aux besoins spécifiques de l'apprenant. Ils évaluent et diagnostiquent les problèmes et difficultés de l'apprenant en apprentissage, et fournissent l'aide nécessaire. Ainsi, les STI placent l'apprenant au centre de la situation apprentissage. Les STI offrent des capacités de raisonnement et de résolution, passant par le recueil de connaissances sur l'apprenant. Ils évaluent les connaissances acquises par l'apprenant, en comparant ses activités et les informations sur le domaine, ils proposent alors des assistances adaptées.

Le STI permet aux environnements d'apprentissage de s'adapter à la diversité des apprenants. Le e-Learning est le support et environnement d'apprentissage pour les apprenants alors que STI est l'adaptation de l'apprentissage selon la capacité et rythme des apprenants. La figure 5.1 présente le lien STI/E-learning

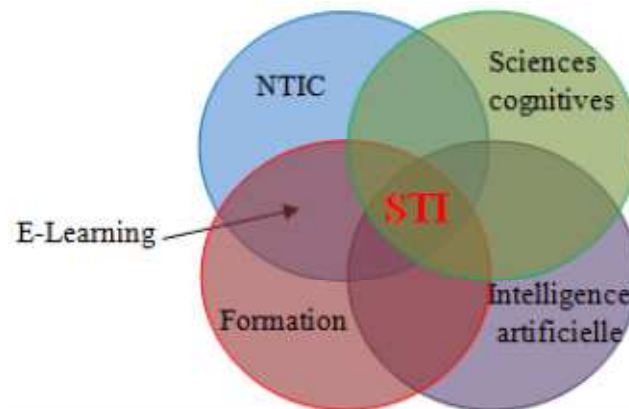


Figure 5.1 : STI/E-learning

5.3 Les plateformes de formation à distance

5.3.1 Définition

Selon Paquette [Paquette, 2002], nous désignerons par plateforme de formation à distance, un système logiciel accessible via un réseau Internet ou Intranet regroupant l'ensemble des outils techniques nécessaires à la création, à la gestion et au suivi d'une formation organisée à distance. On peut citer par exemples les plateformes Moodle [Moodle, 2014], Sakai [Sakai, 2014] et Dokeos [Dokeos, 2014].

Chaque plateforme doit d'une part créer les moyens nécessaires pour faire d'une situation *distancielle* un équivalent acceptable à une situation *présentielle* avec plus de flexibilité par rapport aux contraintes temps et espace et d'autre part proposer les moyens d'accompagnement, d'assistance et d'évaluation de l'apprenant [Durand et al., 2008].

Il existe principalement quatre acteurs (figure 5.2) qui interviennent autour de la plateforme d'apprentissage, chacun avec ces rôles et niveaux d'implication : l'apprenant, le tuteur, le concepteur et les contenus.

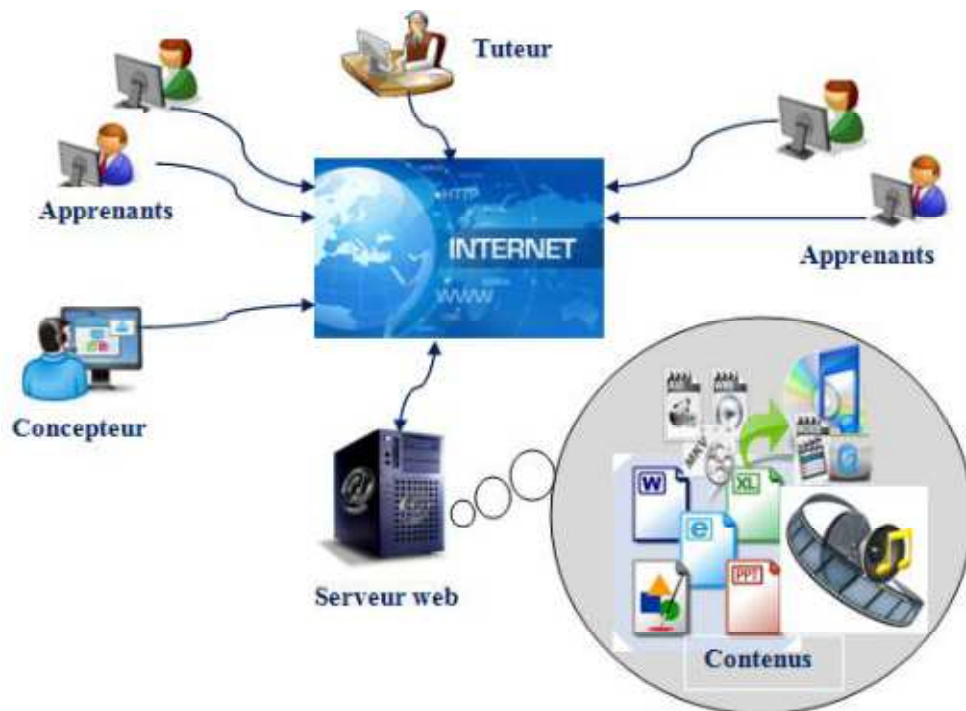


Figure 5.2 : Principaux acteurs d'une plateforme e-learning

L'apprenant, l'acteur principal du système, est au cœur du processus d'apprentissage car il est le seul à pouvoir transformer les informations qu'il reçoit en connaissances appropriées, ce qui est le but de l'apprentissage. L'apprenant suit depuis son poste de travail une formation dont le contenu est organisé selon un parcours pédagogique défini par le formateur / tuteur en fonction des compétences possédées par l'apprenant avant la formation, en fonction des objectifs pédagogiques visés par la formation et en fonction des résultats obtenus lors des évaluations intermédiaires.

Le **tuteur** prend en charge, à la fois, les tâches d'un tuteur de contenu et celle d'un tuteur animateur. Il doit assister les apprenants, superviser leurs travaux, corriger les exercices ne pouvant pas être corrigés par l'ordinateur, met à jour le contenu des cours et/ou exercices [Durand et al., 2008].

Le **concepteur** du cours travaille à partir du scénario pédagogique fourni par l'enseignant. Il a en charge la mise en forme des ressources pédagogiques et leur intégration sur la plateforme.

Les **contenus** : vu le développement des accès distants ces dernières années, les documents pédagogiques ne sont plus limités à des documents textuels, ils peuvent s'enrichir de modules sonores, d'images ou de vidéos, rendant les contenus d'autant plus attractifs. Les standards sont une garantie de pérennité des dispositifs et des contenus et d'interopérabilité des applications. L'intégration de standards comme IMS-LD [IMS-LD, 2014], IMS [MS, 2014] et SCORM [SCORM, 2014], qui travaille au

niveau de l'activité et non plus au niveau du contenu favorisera l'échange et la réutilisabilité d'objets pédagogiques entre les différentes plateformes.

Selon Paquette [Paquette, 2002], une plateforme de formation à distance comporte un ensemble de fonctionnalités techniques qui permettent [Paquette, 2002] :

- A un enseignant concepteur de créer des parcours pédagogiques types regroupant des activités d'apprentissage ainsi que des ressources pédagogiques multimédias et de bâtir des outils de suivi des activités des étudiants ;
- A un apprenant de consulter en ligne ou de télécharger les matières et les ressources pédagogiques qui lui sont recommandées, de les gérer à l'aide de vues sur l'évolution de son travail, d'effectuer des exercices, de s'auto évaluer et de transmettre des travaux à corriger par un enseignant formateur ;
- Aux apprenants et aux personnes ressources qui leur sont affectées (tuteurs, experts du contenu, animateurs, gestionnaires, etc.) de communiquer individuellement ou en groupe, de créer des thèmes de discussion ou de collaborer à des productions communes ;
- A un administrateur de la plateforme d'installer le système d'apprentissage et d'en assurer la maintenance, de gérer les accès et les droits des uns et des autres, de créer des liens avec les systèmes d'information externe (scolarité, catalogues, ressources pédagogiques, etc.).

Les services offerts par ces plateformes mettent à disposition des apprenants un ensemble de cours et de ressources accessibles à travers un réseau local d'entreprise ou à partir d'internet, en utilisant un navigateur. Ces différents outils sont mis à disposition selon les activités pédagogiques : en mode synchrone, apprenants et tuteurs communiquent par *clavardage*, partage d'applications ou vidéo-conférence, les outils asynchrones, messagerie, blogs ou forums, sont utilisés pour des échanges différés [Durand et al., 2008]. Ces systèmes présentent également des générateurs internes de tests d'évaluation que l'on retrouve sous forme de QCM. Ces activités sont soit soumises une validation par l'enseignant soit proposées comme activité de régulation en auto-évaluation. D'une manière générale, elles sont intégrables en tant que ressources pédagogiques dans un parcours d'apprentissage au sein de la plateforme.

Les plateformes offrent aussi des outils de gestion et d'administration tels que la gestion des cours, gestion des apprenants, élaboration de parcours pédagogiques, auto-inscription des apprenants. Des modules complémentaires *plugins* peuvent enrichir les fonctionnalités disponibles : rendu de travaux, visibilité des cours, gestion des notes, moteur de recherche, glossaire intégré, outils collaboratifs Wiki, espace personnel, portfolio, outils de sondage et de statistiques [Durand et al., 2008].

5.3.2 Moodle

Aujourd'hui, Il existe plus de 200 plateformes [Plate-stat, 2014] d'apprentissage en ligne open source ou commerciales. Ainsi, les plateformes open source, de par leur structure, favorisent l'intégration de modules supplémentaires, atout fondamental et très important dans la mise en place de notre système. En effet nous avons besoin

d'accéder au code source de la plateforme afin d'intégrer un module chargé d'intercepter toutes les traces et actions des apprenants. Ce module permet d'alimenter notre système IDCBR-MAS avec les traces des apprenants.

Afin de choisir une plateforme e-Learning, on s'est appuyé sur plusieurs études comparatives [Zouhair, 2009] [Durand et al., 2008], les sites et les documentations des plateformes concernées, et sur des études réalisées dernièrement [Cursus, 2011], [Plume, 2013] et [Mazyad, 2013].

Le choix final s'est donc fait sur la plateforme d'apprentissage en ligne Moodle (*Modular Object-Oriented Dynamic Learning Environment*), qui est l'une des plateformes open source les plus utilisées au monde, avec sa grande communauté francophone et internationale, sa documentation claire, sa bonne intégration de la norme SCORM et ses nombreux forums de discussion. Moodle est né à la *Curtin University of Technology* (Australie), créé par Martin Dougiamas (la première version est sortie au mois d'août 2002), et base son module pédagogique sur le socio-constructivisme.

Moodle ajoute des fonctions pédagogiques pour créer un environnement d'apprentissage en ligne. Moodle permet de créer, par l'intermédiaire du réseau, des interactions entre des pédagogues, des apprenants et des ressources pédagogiques. Elle présente une interface très conviviale avec les formateurs, apprenants et tuteurs. Cette interface se présente sous forme de différents blocs, par exemple le bloc central présente les documents et les activités du cours. Moodle est extrêmement modulaire, la plateforme sait s'adapter à tous les domaines de formation. La figure 5.3 présente la page d'accueil d'un cours Moodle.

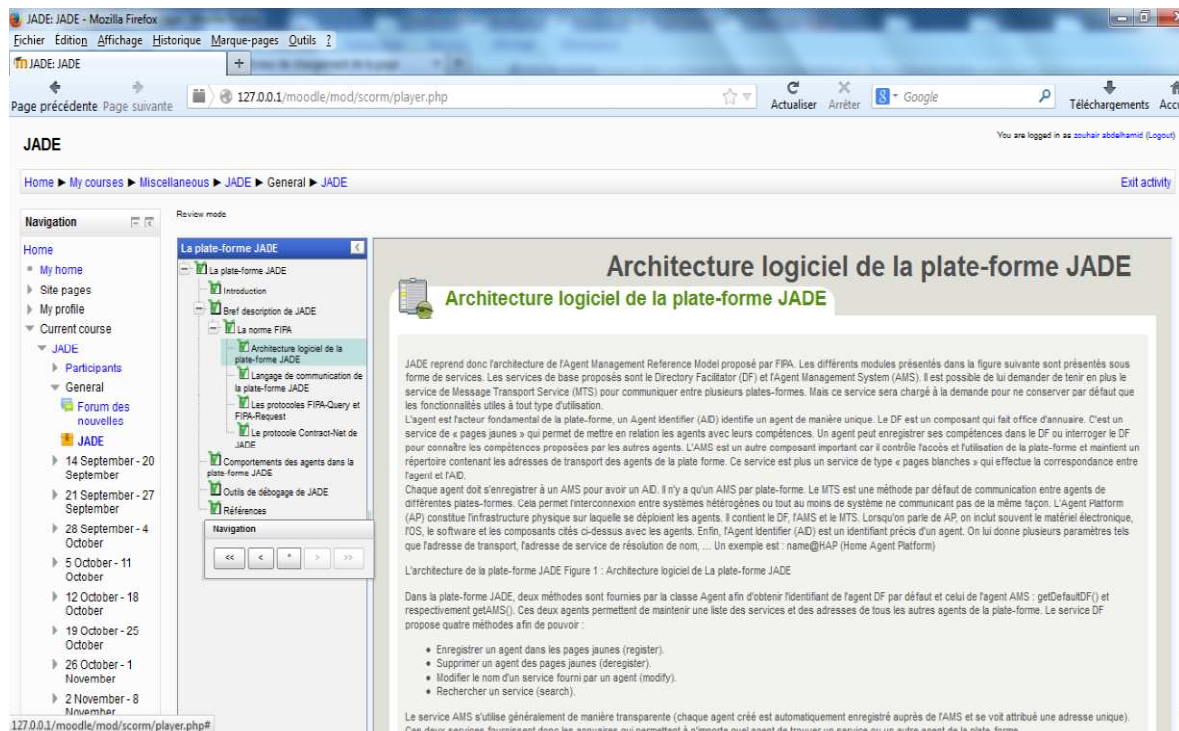


Figure 5.3 : Page d'accueil d'un cours Moodle

Moodle est une plateforme très populaire dans le monde de la formation à distance, avec une estimation de plus de 60 millions d'utilisateurs dans le monde [Moodle-stat, 2014], et avec 1342838 utilisateurs enregistrés sur le site, parlant plus de 78 langues. En plus au mois d'août 2014, plus de 86 000 sites dans 237 pays ont enregistré leur implantation. La plateforme a été traduite dans près de 120 langues. Pour plus de statistique veuillez consulter [Moodle, 2014], [Moodle-stat, 2014].

5.4 Systèmes Tutoriels Intelligents : notre objectif

L'accompagnement et le suivi individualisé de l'apprenant ont pour objectifs l'anticipation d'un éventuel abandon de l'apprenant ou encore l'échec et la démotivation de celui-ci. Le système que nous proposons, permet d'analyser le parcours (trace) de l'apprenant afin d'adapter l'apprentissage selon son rythme et ses capacités.

Les traces des activités passées d'apprentissage seront la source de connaissances pour notre processus d'adaptation d'apprentissage. Elles sont stockées dans la base de cas. Les traces comprennent les productions et actions résultantes de l'interaction de l'apprenant avec la plateforme Moodle. Après le démarrage d'une session d'apprentissage par un apprenant. Ce dernier laisse ses traces sur le serveur d'apprentissage Moodle. Ces dernières peuvent être considérées comme un ensemble d'actions effectuées l'une après l'autre par l'apprenant. Cette suite d'événements peut alors être assimilée à une séquence temporelle d'actions. Donc le problème traité est un cas cible dynamique. Notre objectif sera donc de représenter cette situation (cas cible), prendre en compte le changement dynamique de cette situation, prévoir les évolutions possibles de cette dernière, et réagir en fonction des situations particulières (en fonction du profil des apprenants). Ceci peut être réalisé en s'appuyant sur des situations d'apprentissages antérieures effectuées dans des conditions semblables et dont on connaît les conséquences. Il s'agit alors d'utiliser le raisonnement à partir de cas pour les situations dynamiques.

Les systèmes RàPC classiques souffrent de limites dans la gestion des paramètres dynamiques et ils ne sont pas capables de détecter automatiquement l'évolution de ses paramètres ainsi que s'adapter aux changements de la situation. Ils considèrent chaque changement dans la situation en cours de traitement comme un nouveau cas cible, et par la suite ils exécutent le même cycle statique à nouveau. Or le changement concerne toujours le même cas cible, donc l'utilisation du cycle classique de raisonnement semble non adéquate, par conséquent nous avons besoin d'un cycle dynamique pour traiter un cas cible dynamique. Dans le cadre de cette thèse nous avons proposé un RàPCD avec un cycle RàPC dynamique. En effet plusieurs va-et-vient entre les étapes du cycle RàPC peuvent se produire. Notre système est capable de ré-exécuter une étape antérieure du cycle RàPC suite à un changement dans le cas cible. Nous avons choisi le e-learning comme domaine d'application, plus particulièrement l'apprentissage des apprenants sur la plateforme Moodle et un cours JADE a été installé sur Moodle afin de tester notre approche. Les apprenants laissent leurs traces

en fichier Log de Moodle. Ces traces permettent d'alimenter notre système via des agents d'interface, ensuite notre système procède à l'analyse de l'évolution de ces traces afin de détecter rapidement les situations de réussite ou d'échec dans lesquelles se trouvent les apprenants. Notre système IDCBR-MAS permet à tout moment de prédire les prochaines tâches à exécuter par l'apprenant.

L'approche IDCBR-MAS que nous proposons permet d'apporter des éléments de réponses aux difficultés liées au développement et aux limites des systèmes tutoriels intelligents :

- la difficulté d'acquisition des connaissances du domaine pour les systèmes STI rend le développement de ces derniers coûteux et difficile [Diattara, 2013]. Notre approche, ne nécessite aucunes connaissances du domaine ce qui constitue un atout pour sa mise en pratique.
- selon Le Corre [Le Corre, 2013] l'une des principales limites des STI est le manque de généricité. Par exemple, les modifications de STI seront nécessaires chaque fois que nous changeons le domaine d'apprentissage. De plus, selon Mendelsohn et Dillenbourg [Mendelsohn et Dillenbourg, 1993], cité par [Croset, 2009], les techniques des STI sont difficilement applicables à des domaines moins bien structurés que la programmation, les mathématiques, ... Nous proposons une approche générique indépendante de tout domaine d'apprentissage.
- enfin dans notre approche, on n'a pas besoin d'algorithmes de calcul, en dehors de la mesure de similarité, qui est facile à mettre en application.

Notre approche permet à Moodle de s'adapter à la diversité des apprenants : nous proposons d'utiliser Moodle comme support et environnement d'apprentissage pour les apprenants. La plateforme Moodle permet la mise à disposition des apprenants de ressources pédagogiques (textes, audio, vidéo, évaluations, ...). Elle permet aussi la communication entre les tuteurs et les apprenants (chat, forum). Mais elle n'offre pas une adaptation d'apprentissage selon les besoins des apprenants ce qui présente des faiblesses et des inconvénients majeurs par conséquent une démotivation des apprenants. Nous proposons alors le système IDCBR-MAS, comme complément à Moodle dans le but d'adapter l'apprentissage selon la capacité et rythme des apprenants.

5.5 Système Tutoriel Intelligent et RàPCD

Pour valider notre prototype nous avons réalisé plusieurs expérimentations. Ainsi, nous avons développé notre prototype en suivant les étapes ci-après :

- installation de Moodle ;
- création et ajout du cours jade ;
- programmation du module entre Moodle et IDCBR-MAS ;
- création de la base de cas ;
- développement des agents IDCBR-MAS ;
- développement de l'interface web.

Le prototype est composé de deux grandes parties complémentaires :

- **Application multi-agents** : la programmation du système multi-agents IDCBR-MAS. Nous avons développé aussi des interfaces de visualisation pour qu'on puisse suivre et étudier l'évolution des agents, analyser leurs comportements et apporter les réglages nécessaires.

- **Application Web** : c'est un portail web, interface entre IDCBR-MAS et les différents utilisateurs (apprenants, tuteurs et administrateur).

L'application de notre approche IDCBR-MAS dans le domaine e-learning est un paradigme de raisonnement dans lequel les sources de connaissances principales sont les traces produites par l'apprenant quand il suit une formation sur Moodle. Donc la trace d'un apprenant en cours d'apprentissage (trace cible) sera traitée par notre système IDCBR-MAS en utilisant le cycle RàPCD et en se basant sur la mesure de similarité ILCSS. Le résultat de ce traitement est une proposition par notre système de l'action future appropriée au cas cible en cours de progression.

5.5.1 Création du cours (JADE) sous Moodle

Nous rappelons que le STI proposé est indépendant de l'objet de l'étude. Dans le cadre de cette thèse, nous avons choisi le cours JADE comme objet d'apprentissage sur la plateforme Moodle pour valider notre architecture IDCBR-MAS. IDCBR-MAS est générique et indépendante de tout objet d'apprentissage et donc le cours JADE n'est qu'un objet de formation parmi d'autres. Ainsi la plateforme Moodle contient un cours sur la plateforme de développement des systèmes multi-agents JADE, des apprenants, un tuteur et un administrateur. Chaque apprenant doit d'abord s'inscrire sur Moodle, puis s'inscrire au cours JADE afin de suivre la formation.

Moodle intègre tous les outils nécessaires à l'importation et à l'édition du parcours pédagogiques. Il est important de noter aussi que Moodle intègre un environnement d'exécution qui permet de vérifier la conformité du paquetage importé. Si l'objet n'est pas valide, un message d'erreur sera affiché. En ce qui concerne la conception et la mise en œuvre du cours JADE sur la plateforme d'apprentissage Moodle, nous avons pris en compte les normes et standards de conception des documents pédagogiques, en particulier la norme SCORM (*Sharable Content Object Reference Model*), qui nous garantit le partage et la réutilisation de ce cours sur une autre plateforme d'apprentissage. Nous avons utilisé le logiciel *eXelearning* [eXe, 2014] pour la conception et la création du cours sur JADE. *eXelearning* est un outil open source permettant de créer des contenus structurés et de les exporter dans le format SCORM. La figure 5.4 montre la page d'accueil du cours JADE importée sous Moodle.

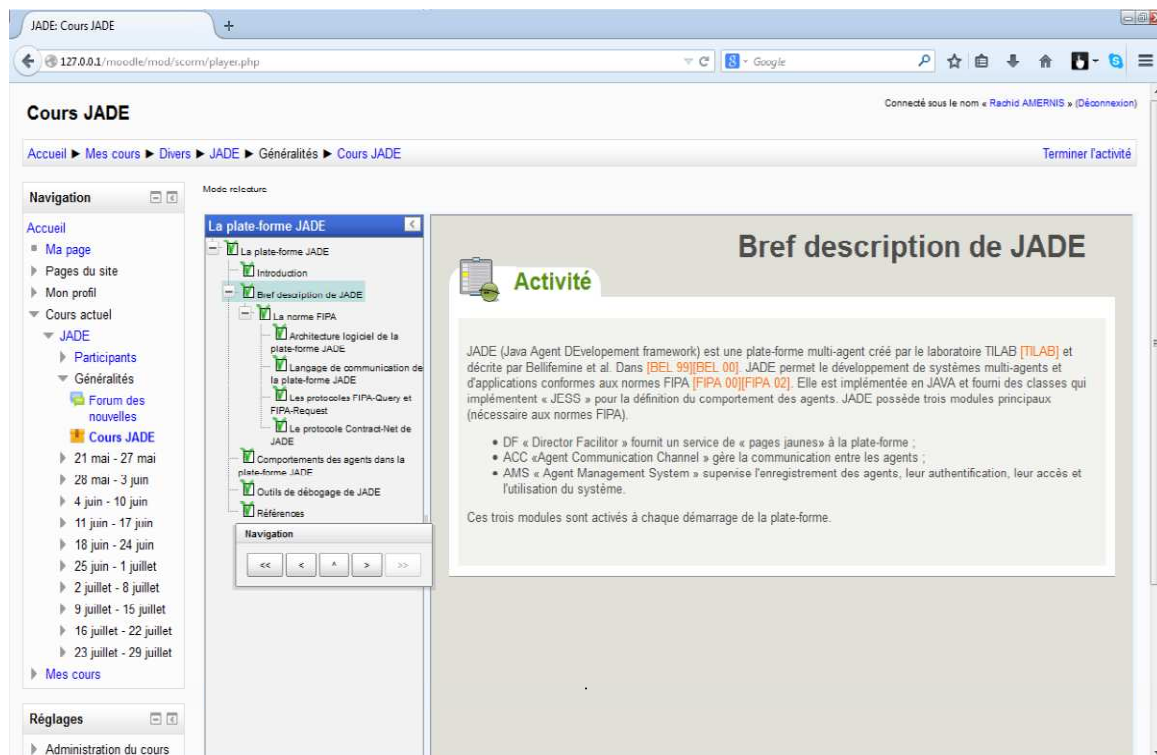


Figure 5.4 : Page d'accueil du cours JADE importée sous Moodle, éditée par eXelearning

La plateforme Moodle est installée sur un serveur web local, ce qui permet à n'importe quel apprenant connecté avec son ordinateur au réseau local de s'inscrire puis de suivre les formations pour lesquelles il est inscrit. Moodle est une plateforme pédagogique web, qui permet aux enseignants/tuteurs d'organiser leurs ressources pédagogiques (cours JADE) et aux apprenants d'y accéder à travers un simple navigateur web. Nous rappelons que les sessions d'apprentissage du cours JADE sur Moodle sont ouvertes seulement aux inscrits. Les apprenants peuvent suivre leurs formations selon leur disponibilité. Les apprenants laissent des traces de leurs activités sur le serveur Moodle et plus exactement sur le fichier Log de Moodle. Les traces alimentent le système IDCBR-MAS en informations dont il a besoin. Afin de collecter, structurer puis présenter ces traces, nous avons développé un module en Moodle qui sert d'interface entre Moodle et IDCBR-MAS.

5.5.2 Interface Moodle/ IDCBR-MAS

Nous avons développé un module dans Moodle, qui sera l'interface entre le serveur Moodle et notre système IDCBR-MAS. Ce module est composé :

1. d'un fichier *datalib.php* version IDCBR-MAS : nous avons modifié *datalib.php* de Moodle, afin de personnaliser la collecte, la structuration et l'enregistrement des traces des apprenants connectés au serveur Moodle ;
2. des traces des apprenants enregistrés sur Moodle : un fichier et un seul pour chaque apprenant ;

3. d'une base de cas, c'est un fichier XML dans lequel la structuration et l'organisation des cas est faite selon des niveaux hiérarchiques, décrits dans la section base de cas.

Le module utilise la même base de données de Moodle. La figure 5.5 présente l'architecture du système IDCBR-MAS et ses liens avec Moodle et l'apprenant.

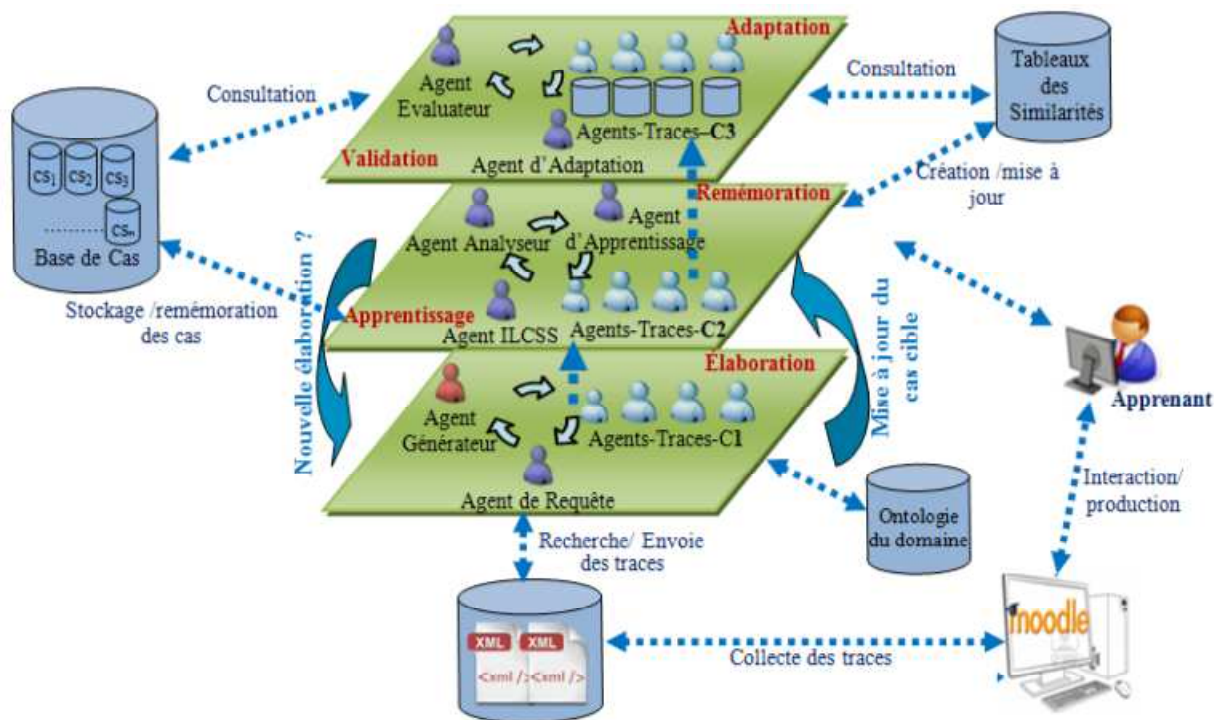


Figure 5.5 : Architecture Moodle-IDCBR-MAS [Zouhair et al., 2014]

La figure 5.6 représente le fichier *datalib.php* qu'on a développé afin d'enregistrer

```

C:\www\www\moodle\lib\datalib.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
base de cas.xml C:\www\www\moodle\lib\datalib.php user.xml C:\www\www\moodle\lib\datalib.php base de cas - Copie.xml C:\www\www\moodle\lib\datalib.php

1857
1858 function insert_xml($log)
1859 {
1860     global $DB, $CFG;
1861     if($log['userid']!=1 && $log['userid']!=0 && $log['userid']!=2)
1862     {
1863         $nomfichier=$CFG->dirroot."&#92;xml&#92;users.xml";
1864         if(file_exists($nomfichier))
1865         {
1866             $user=$log['userid'];
1867             $nomfich=$CFG->dirroot."&#92;xml&#92;user&#92;".$log['userid'].".xml";
1868             $sql_request = "SELECT u.* $ctxselect FROM {user} u WHERE u.id=".$log['userid'];
1869             $result = $DB->get_records_sql($sql_request, null, null);
1870             if($result != null)
1871             foreach($result as $us)
1872             {
1873                 $user=$us->firstname." ".$us->lastname;
1874                 $sql_request = "SELECT MAX(l.id) as max $ctxselect FROM {log} l WHERE l.userid=".$log['userid'];
1875                 $result3 = $DB->get_records_sql($sql_request, null, null);
1876                 if($result3 != null)
1877                 {
1878                     foreach($result3 as $l)
1879                     {
1880                         $id=$l->max;
1881                     }
1882                     $sql_request = "SELECT c.* $ctxselect FROM {course} c WHERE c.id=".$log['course'];

```

Figure 5.6 : Fichier *datalib.php* version IDCBR-MAS [Zouhair et al., 2014]

toutes les actions, interactions et productions des apprenants sur le serveur Moodle. La modification du fichier *datalib.php* nous permet de personnaliser les traces collectées des apprenants selon notre propre structure. Dans la section suivante nous détaillons ces traces ainsi que leurs structures.

5.5.3 Trace de l'apprenant

Dans notre application on distingue deux types de cas : traces en cours de comparaison (cas cible) et traces passées enregistrées dans la base de cas (cas sources). Les traces des apprenants (les cas cible et sources), désignent les informations que Moodle enregistre sur l'activité, la production, et toute action résultante de l'interaction de ses apprenants. Ces traces (TA) sont représentées par une collection des traits sémantiques : triplet : TS = (objet, (qualification, valeur) +). Nous utilisons le langage XML pour formater les TS qui alimentent le système IDCBR-MAS. La figure 5.7 représente un exemple d'un fichier trace (format XML) d'un apprenant en interaction avec Moodle.

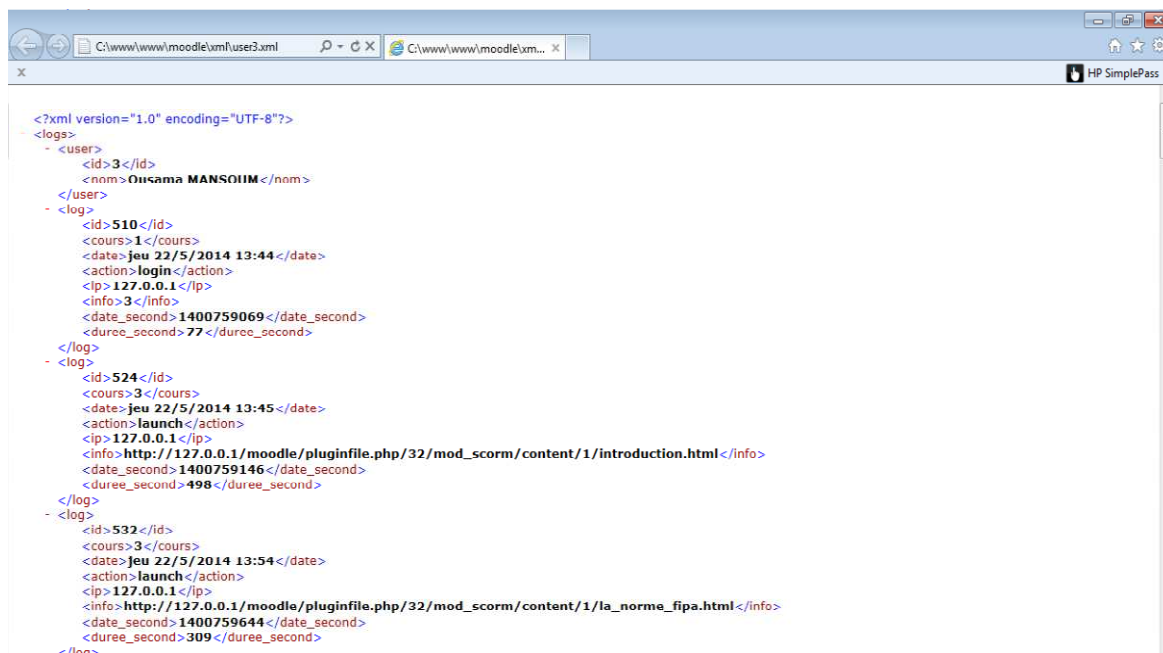


Figure 5.7 : Fichier trace d'un apprenant en interaction avec Moodle et IDCBR-MAS

5.5.3.1 Trait sémantique

Ci-après l'exemple du trait sémantique (TS) d'un apprenant consultant le cours JADE sur la plateforme Moodle :

TS = (A1, ((instant, 12), (cours, JADE), (date, jeu 27/3/2014 15: 47), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/la_norme_fipa.html), (duree, 1800))).

Avec :

- A1 : l'apprenant A1
- l'instant : indique l'ordre chronologique de l'action

- cours : l'objet du cours/activité pédagogique
- date : indique la date et heure d'apprentissage
- action : décrit l'action de l'apprenant (consultation de cours par exemple)
- ip : indique l'adresse réseau de la machine de l'apprenant
- info : indique la cible de l'activité pédagogique de l'apprenant, par exemple s'il s'agit de consultation, cette connaissance indique le paragraphe consulté
- duree : c'est la durée de l'action/production, la durée est en seconde

Ce trait sémantique représente l'information suivante : à l'instant t_{12} , l'apprenant A1 commence la lecture du paragraphe la_norme_fipa du cours JADE. La durée de consultation est de 30 minutes.

Ci-dessous la représentation du même TS en format XML :

```
<user>A1
  <instant>12</instant>
  <cours>JADE</cours>
  <date> jeu 27/3/2014 15:47</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/la_norme_fipa.html </info>
  <duree >1800</duree>
</user>
```

5.5.3.2 Cas source

Les cas sources sont des anciennes traces d'apprenants stockées dans la base de cas du système IDCBR-MAS. Un cas source est une collection de traits sémantiques. Ci-après l'exemple d'un cas source de 35 traits sémantiques en plusieurs formats : liste de TS, tableau de TS et format XML.

1) Cas source en format liste de TS :

Trace-source =

```
(A1, ((instant, 1), (cours, 1), (date, jeu 27/3/2014 15: 07), (action, login), (ip, 127.0.0.1), (info, 7), (duree, 5)),
((instant, 2), (cours, JADE), (date, jeu 27/3/2014 15: 07), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/introduction.html), (duree, 600)),
...,
((instant, 12), (cours, JADE), (date, jeu 27/3/2014 15: 47), (action, launch), (ip, 127.0.0.1), (info,
```


http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/la_norme_fipa.html), (duree, 1800)),

....,

((instant, 34), (cours, JADE), (date, jeu 27/3/2014 16:55), (action, login), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/le_protocole_contractnet_de_jade.html), (duree, 1200)),

((instant, 35), (cours, JADE), (date, jeu 27/3/2014 17: 15), (action, logout), (ip, 127.0.0.1), (info, 7), (duree, 5))).

Remarque :

- les cas sources se terminent toujours par un trait sémantique dans l'action est la fermeture de la session d'apprentissage. Dans notre exemple, il s'agit du trait sémantique 35 avec logout comme action.
- nous n'avons pas détaillé tous les TS du cas source cité ci-dessus, cependant l'annexe A contient une liste complète des cas sources.

2) Cas source en format tableau des traits sémantiques :

Le même cas source sous format de tableau de 35 traits sémantiques :

Numéro du TS	Trait sémantique
TS1	(A1, ((instant, 1), (cours, 1), (date, jeu 27/3/2014 15: 07), (action, login), (ip, 127.0.0.1), (info, 7), (duree, 5)),
TS2	((instant, 2), (cours, JADE), (date, jeu 27/3/2014 15: 07), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/introduction.html), (duree, 600))
TS3 ...TS11	
TS12	((instant, 12), (cours, JADE), (date, jeu 27/3/2014 15: 47), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/la_norme_fipa.html), (duree, 1800))
TS13... TS33	
TS34	((instant, 34), (cours, JADE), (date, jeu 27/3/2014 16:55), (action, login), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/le_protocole_contractnet_de_jade.html), (duree, 1200))
TS35	((instant, 35), (cours, JADE), (date, jeu 27/3/2014 17: 15), (action, logout), (ip, 127.0.0.1), (info, 7), (duree, 5)))

3) Cas source en format XML :

La représentation du même exemple du cas source en format XML :

<user > A1

```
<instant>1</instant>
<cours>1</cours>
<date>jeu 27/3/2014 15: 07</date>
<action>login</action>
<ip>127.0.0.1</ip>
<info>7</info>
<duree >5</duree>
<instant>2</instant>
<cours>JADE</cours>
<date>jeu 27/3/2014 15:07</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/introduct
tion.html</info>
<duree >600</duree>
</log>
</log>
<log >
</log>
<log >
  <instant>12</instant>
  <cours>JADE</cours>
  <date>jeu 27/3/2014 15:47</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/la_norm
e_fipa.html</info>
  <duree >1800</duree>
</log>
<log >
</log>
  <instant>34</instant>
  <cours>JADE</cours>
  <date>jeu 27/3/2014 16:55</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/le_proto
cole_contractnet_de_jade.html</info>
  <duree >1200</duree>
```

```

</log>
<log >
  <instant>35</instant>
  <cours>JADE</cours>
  <date>jeu 27/3/2014 17:15</date>
  <action>logout</action>
  <ip>127.0.0.1</ip>
  <info>7</info>
  <duree >5</duree>
</log>
</user>

```

5.5.3.3 Cas cible

Un cas cible représente une trace d'un apprenant en progression. Ci-après l'exemple d'un cas cible de 5 traits sémantiques en plusieurs formats liste des TS, tableau des TS et format XML :

1) Cas cible en format liste des traits sémantiques:

Trace-cible =

(A10, ((instant, 1), (cours, 1), (date, lu 31/3/2014 11: 02), (action, login), (ip, 127.0.0.1), (info, 7), (duree, 5)),
 ((instant, 2), (cours, JADE), (date, lu 31/3/2014 11: 02), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/introduction.html), (duree, 660))
 ...,
 ((instant, 5), (cours, JADE), (date, lu 31/3/2014 11: 39), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/la_norme_fipa.html), (duree, 1720)).

2) Cas cible en format tableau des traits sémantiques :

Le même cas cible sous format de tableau de 5 traits sémantiques :

Numéro du TS	Trait sémantique
TS1	(A10, ((instant, 1), (cours, 1), (date, lu 31/3/2014 11: 02), (action, login), (ip, 127.0.0.1), (info, 7), (duree, 5)),
TS2	((instant, 2), (cours, JADE), (date, lu 31/3/2014 11: 02), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/i

	ntroduction.html), (duree, 660))
TS3 ...TS4	
TS5	((instant, 12), (cours, JADE), (date, lu 31/3/2014 11: 39), (action, launch), (ip, 127.0.0.1), (info, http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/1a_norme_fipa.html), (duree, 1720)).

3) Cas cible en format XML :

La représentation du même cas cible en format XML :

```

<user > A10
<log >
    <instant>1</instant>
    <cours>1</cours>
    <date> lu 31/3/2014 11: 02</date>
    <action>login</action>
    <ip>127.0.0.1</ip>
    <info>7</info>
    <duree >5</duree>
</log>
<log >
    <instant>2</instant>
    <cours>JADE</cours>
    <date> lu 31/3/2014 11: 02</date>
    <action> launch </action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/introduction.ht
        ml </info>
    <duree >660</duree>
</log>
<log >
</log>
<log >
    <instant>5</instant>
    <cours>JADE</cours>
    <date> lu 31/3/2014 11: 39</date>
    <action> launch </action>
    <ip>127.0.0.1</ip>
    <info>http://127.0.0.1/moodle/pluginfile.php/25/mod_scorm/content/1/la_norm
    e_fipa.html </info>
    <duree >1720</duree>
</log>
</user>

```

5.5.4 Base de cas

La base de cas du système IDCBR-MAS contient les cas sources ou les anciennes traces laissées par les apprenants qui ont suivi la formation sur Moodle. Dans notre application du système IDCBR-MAS en e-Learning, la structure de la base de cas est une structure hiérarchique contenant deux niveaux. Le bas niveau est une simple

mémoire plate organisée en un ensemble de profils d'apprenants où chaque profil contient un groupe de cas sources similaires. Le haut niveau contient des index représentant les caractéristiques communes d'un profil ou d'un groupe de cas similaires stockés dans le bas niveau.

Par exemple un profil contient les traces des apprenants qui ont suivi pratiquement le même ordre chronologique des activités pédagogiques (consultation de cours, réalisation d'un QCM...), mais ils n'ont pas, par exemple, exactement les mêmes durées de réalisation de certaines activités pédagogiques (quelques minutes de différence) et ils n'ont pas exactement la même note en QCM (par exemple tous les apprenants du profil ont une note entre 12 et 14). Un deuxième exemple, le profil des apprenants démotivés : il regroupe les traces des apprenants qui ont sauté des étapes dans leurs processus d'apprentissage, ils ont passé trop ou peu de temps dans des activités pédagogiques importantes. Ils ont commencé leurs processus d'apprentissage par exemple par la réalisation du QCM au lieu de commencer par l'introduction, ensuite définitions, ect.

La base de cas de notre application est présentée par la figure suivante :

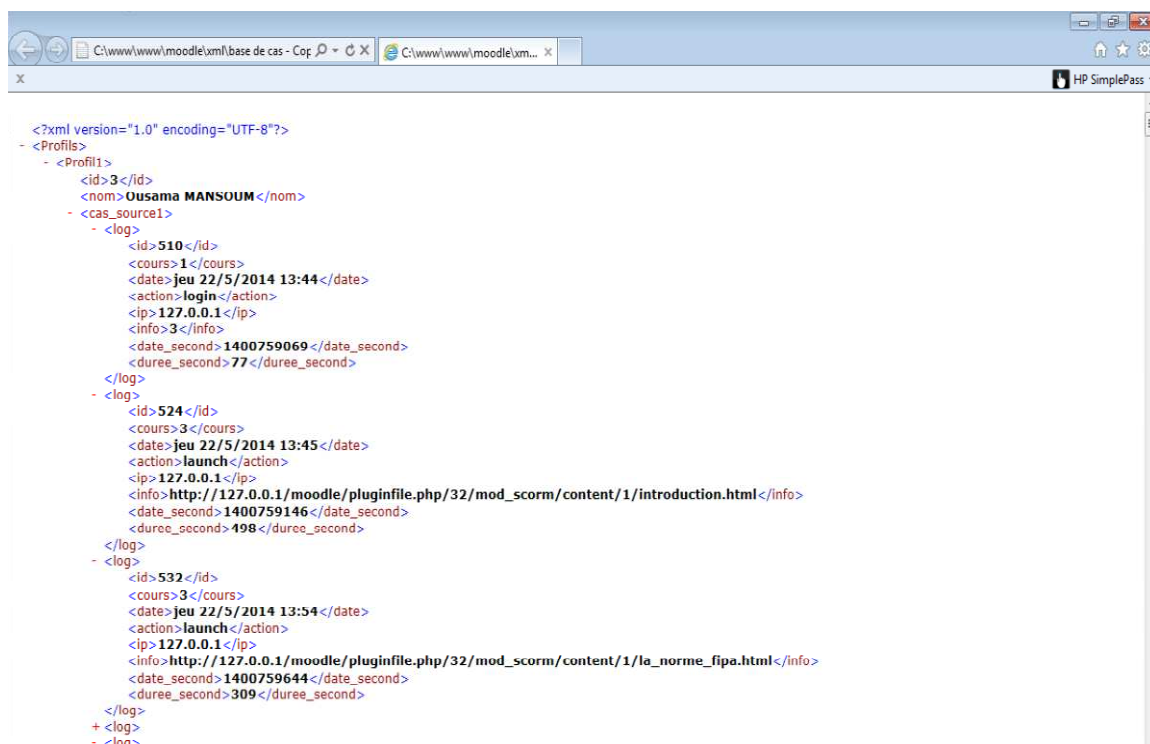


Figure 5.8 : Structure globale de la base de cas en système IDCBR-MAS

Après la description du cas cible, cas sources et la base de cas, nous présentons dans la section suivante le fonctionnement de notre système, particulièrement le cycle dynamique et la mesure de similarité.

5.6 Développement des agents IDCBR-MAS

5.6.1 Aspects techniques

Pour développer notre prototype, nous avons utilisé le langage JAVA et la plateforme JADE de développement des SMA. Nous utilisons aussi des outils et interfaces afin de contrôler et superviser nos agents.

- *JADE GUI* pour contrôler et modifier l'état interne des agents de IDCBR-MAS ;
- *Remote Monitoring Agent (RMA)* pour superviser la communication et le comportement des agents de IDCBR-MAS ;
- *Introspector agent JADE* pour contrôler le cycle de vie d'un agent et les messages reçus ou émis par ce dernier.

5.6.1.1 Motivation à l'utilisation de la plateforme JADE

Il existe actuellement plusieurs plateformes multi-agents pour le développement des SMA : JADE [JADE, 2013], AgentBuilder [AgentBuilder, 2014], Voyager [Voyager 2013], Zeus [Zeus, 2013], Swarm [Swarm, 2013], JACK [JACK, 2013], Madkit [Madkit, 2013]. Ces plateformes peuvent être classées dans les catégories suivantes [Sohail et al., 2005] :

- les plateformes pour agents cognitifs, tels que JADE et AgentBuilder ;
- les plateformes pour agents mobiles tel que Voyager ;
- les plateformes pour agents collaboratifs tel que Zeus ;
- les plateformes pour la simulation multi-agents tels que Swarm et Madkit.

Nous avons choisi JADE pour les raisons suivantes :

1. JADE est conforme aux normes et spécifications de FIPA ;
2. JADE est une plateforme open source programmée en Java ;
3. les agents JADE communiquent en se basant sur le langage ACL, dans le cas où les agents sont hétérogènes, alors ils coopèrent par négociation ;
4. JADE est composée d'un ensemble de bibliothèques (des classes Java) pour la création d'agents. Elle est compatible avec les systèmes d'exploitation Windows et Linux ;
5. la plateforme JADE peut être répartie sur plusieurs serveurs en se basant sur la Machine Virtuelle Java (*Java Virtual Machine - JVM*), ce qui permet à plusieurs agents de s'exécuter en parallèle.

5.6.1.2 Outils de contrôle et de supervision des agents dans IDCBR-MAS

Une plateforme d'agents développée en JADE peut être répartie sur plusieurs serveurs en se basant sur l'exécution d'une machine virtuelle JVM (*Java Virtual Machine*) par serveur. Chaque JVM est un conteneur d'agents capable de fournir un environnement complet pour l'exécution des agents en même temps. L'outil JADE GUI de la figure 5.9 permet de contrôler et superviser les agents de notre plateforme IDCBR-MAS, et en cas de besoin de modifier leur état interne (créer, tuer, suspendre,...) et il permet de lancer les autres outils.

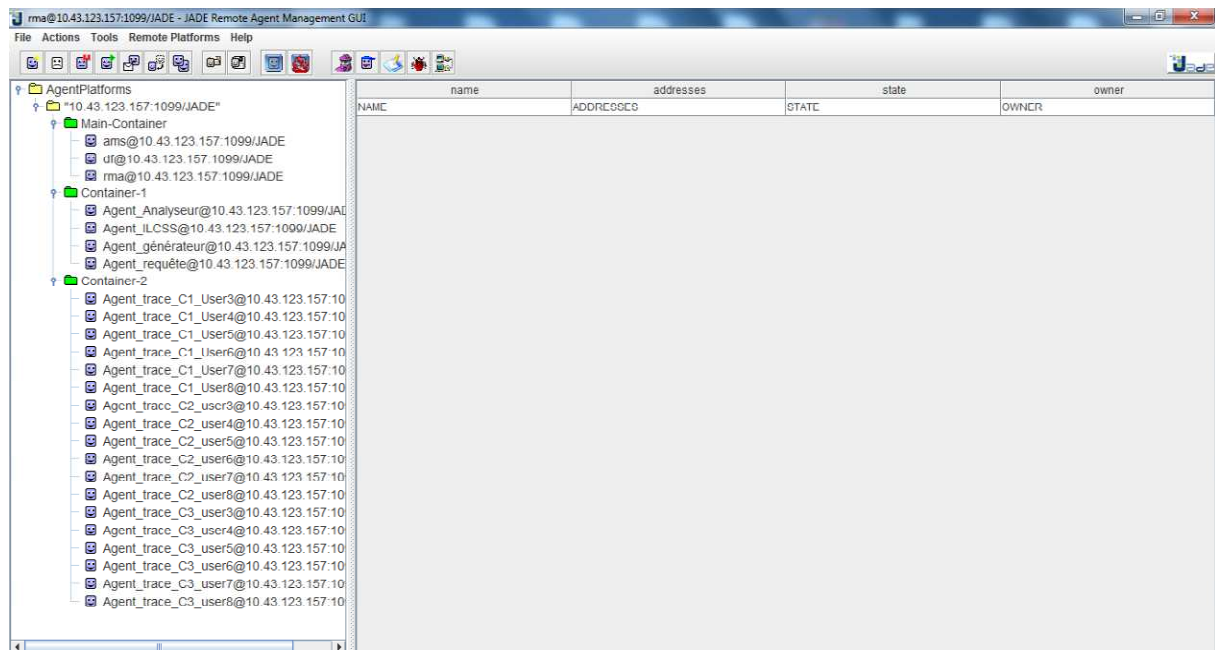


Figure 5.9 : Aperçu de l'interface RMA (Remote Monitoring Agent) dans notre plateforme IDCBR-MAS

5.6.1.3 Communication inter-agents dans IDCBR-MAS

Dans le but de surveiller et contrôler la communication et le comportement des agents au sein de la plateforme IDCBR-MAS, JADE fournit des outils graphiques, tel que l'agent Remote Monitoring Agent (RMA). RMA qui représente l'interface principale de surveillance et du contrôle de la plateforme JADE et de tous ses conteneurs distants. Nous utilisons RMA pour lancer les autres outils graphiques, comme Sniffer. Le but est de surveiller les échanges de messages dans notre plateforme IDCBR-MAS. Il permet d'afficher le flux d'interactions entre les agents sélectionnés, et il permet aussi d'afficher le contenu de chaque message échangé. La figure 5.10 illustre les différentes communication/interactions entre les agents de la plateforme IDCBR-MAS.

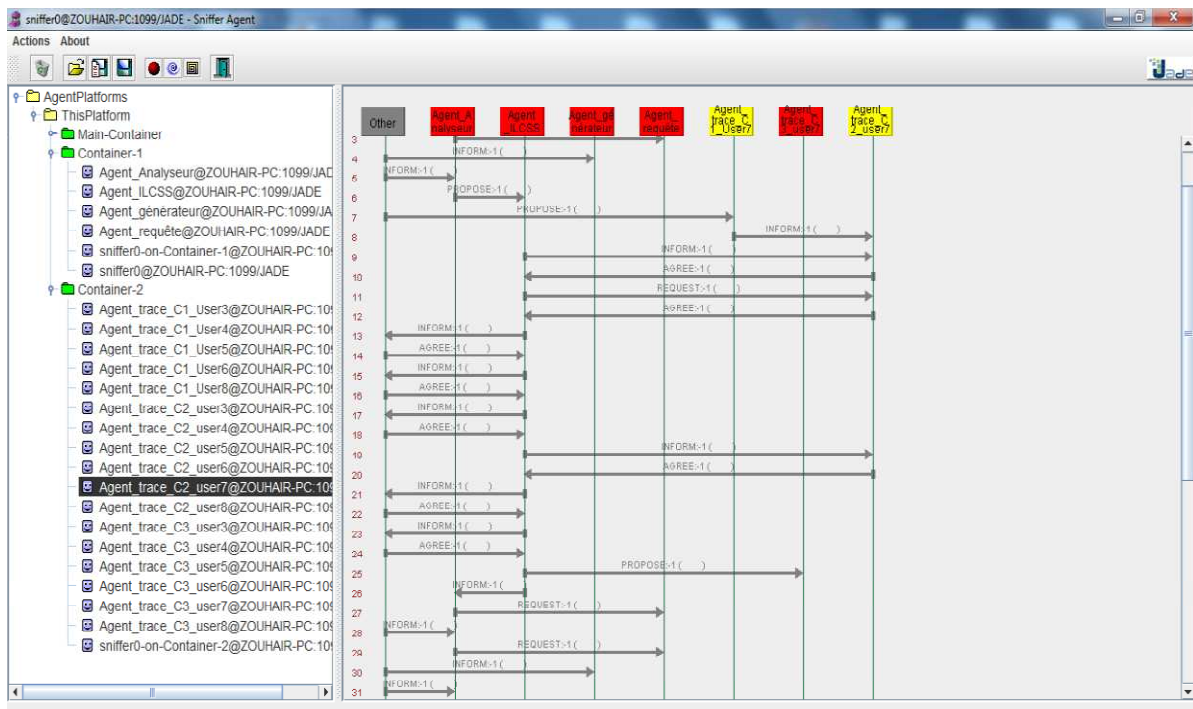


Figure 5.10 : Communication inter-agent dans IDCBR-MAS [Zouhair et al., 2014]

5.6.1.4 Introspector agent JADE dans IDCBR-MAS

Cet outil permet de surveiller et contrôler l'état ou cycle de vie d'un agent dans IDCBR-MAS ainsi que les messages reçus ou émis par cet agent. La figure 5.11 présente l'état ainsi que les messages émis/reçus de l'agent Agent_ILCSS de notre plateforme IDCBR-MAS.

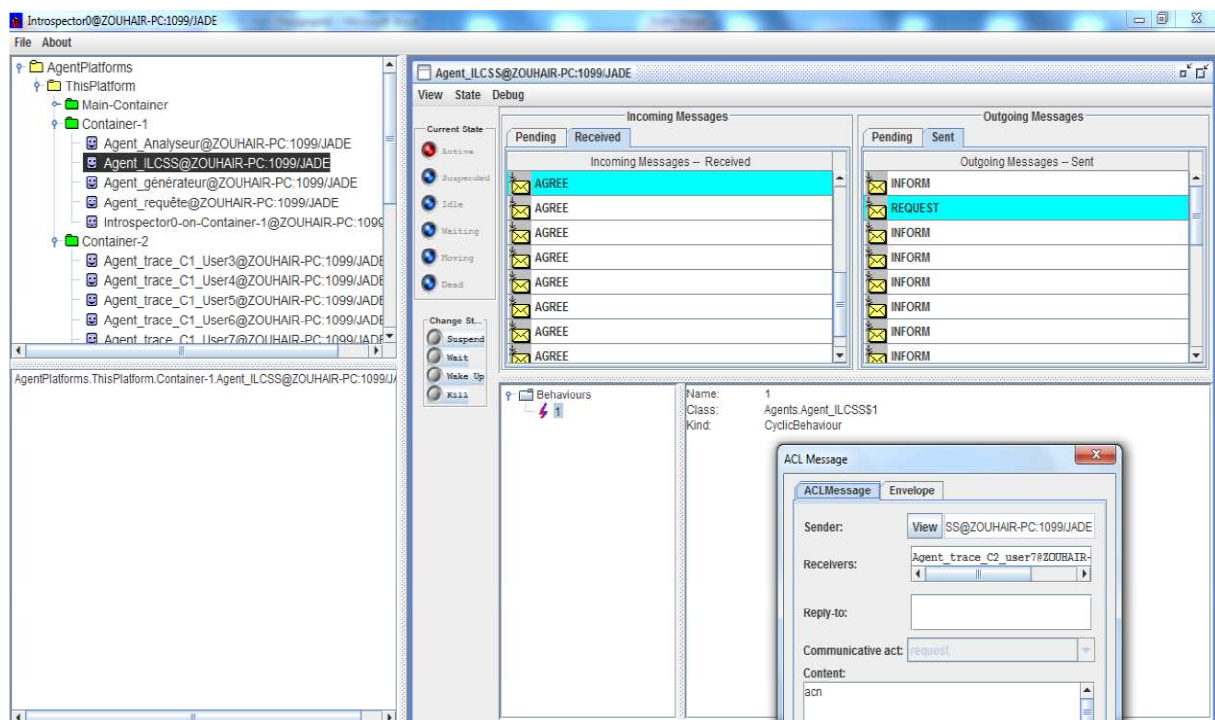


Figure 5.11 : Messages émis/reçus de l'Agent_ILCSS [Zouhair et al., 2014]

La figure 5.12 présente une partie du code de l'agent Agent_ILCSS de notre plateforme IDCBR-MAS.

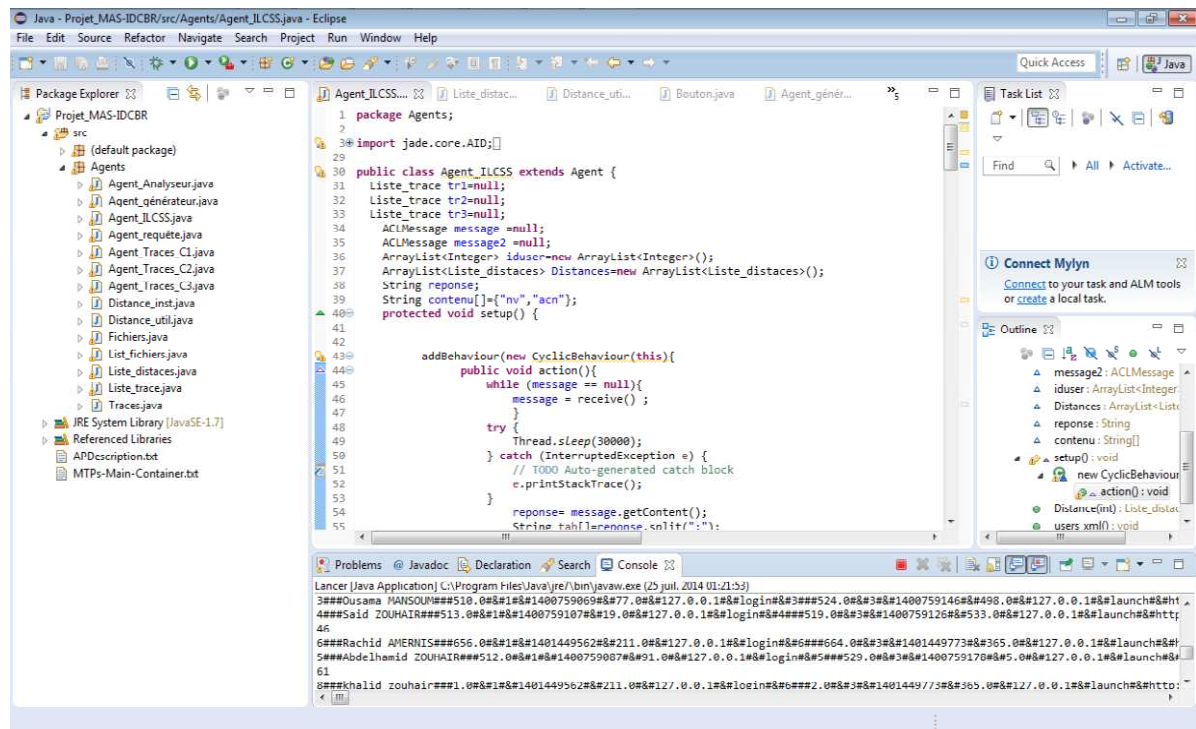


Figure 5.12 : Aperçu de l'Agent_ILCSS de l'IDCBR-MAS

5.6.2 Les agents IDCBR-MAS appliqués au STI

5.6.2.1 Processus d'élaboration d'une situation en cours de progression

Nous rappelons que l'élaboration du cas cible est réalisée par la couche d'élaboration de IDCBR-MAS, qui contient l'agent de requête, les agents-Traces-C1 et l'agent générateur. L'apprenant laisse les traces de ses interactions, productions et toutes ses actions sur le serveur dans le fichier Log. L'agent de requête récupère ces traces depuis le serveur et les transmet à l'agent générateur. Ce dernier vérifie si c'est un nouvel apprenant, et dans ce cas, l'agent générateur crée une instance de l'agent-Traces-C1. Dans le cas contraire, l'agent générateur transmet les nouvelles traces de l'apprenant à l'agent-Traces-C1 concerné afin de se mettre à jour. L'agent de requête refait ce travail d'une façon périodique et à chaque fois qu'il détecte un changement dans les fichiers traces de l'apprenant concerné. La figure suivante illustre le diagramme de séquence d'élaboration du cas cible.

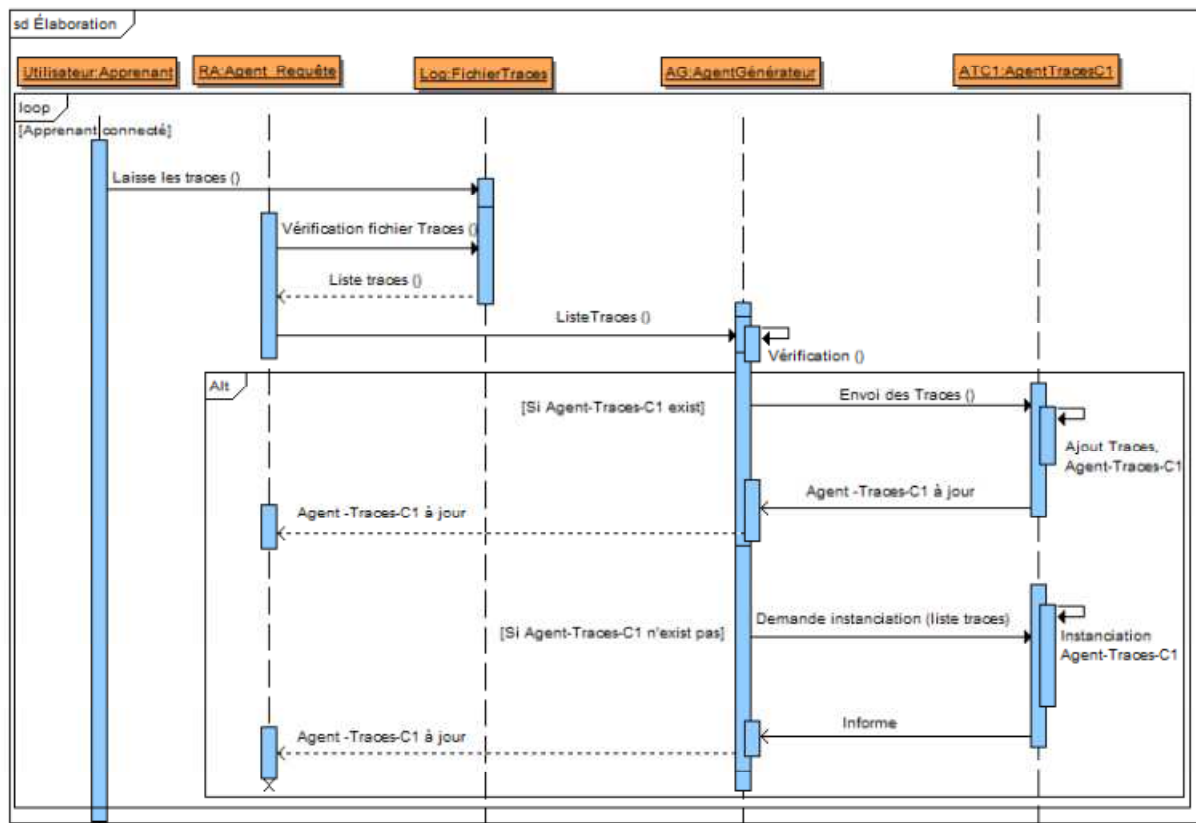


Figure 5.13 : Diagramme de séquence d'élaboration du cas cible

5.6.2.2 Processus de remémoration de la situation en cours de progression

C'est une phase primordiale dans notre approche. Lors du déclenchement de cette phase, l'agent analyseur initialise la recherche des cas sources similaires au cas cible en cours en se basant sur la mesure de similarité ILCSS. Le diagramme de séquence suivant illustre le processus de remémoration.

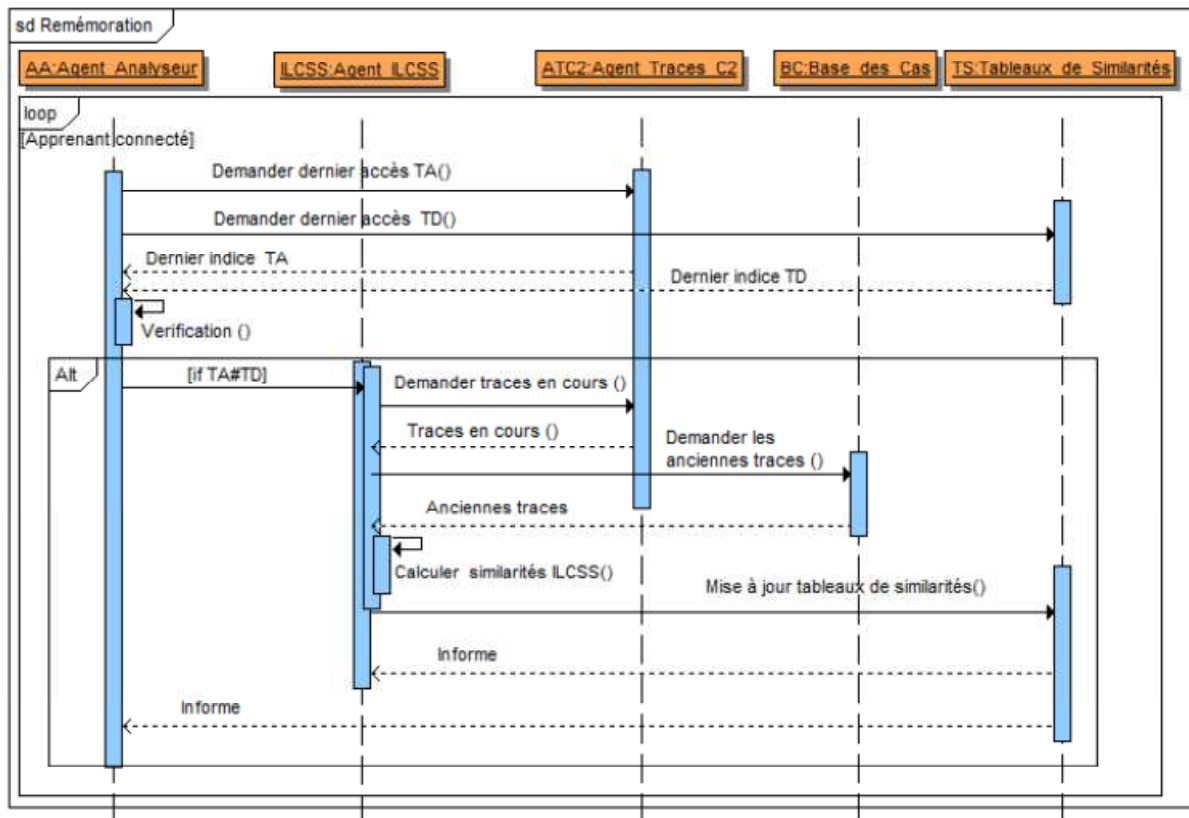


Figure 5.14 : Diagramme de séquence de la phase remémoration

5.6.2.3 Processus d'adaptation de la situation en cours de progression

L'adaptation de la situation consiste à trouver les expériences passées les plus similaires à la situation en cours de progression. L'agent d'adaptation demande à l'agent-Traces-C3 la liste des cas sources similaires. Ensuite cette liste sera transmise à l'agent évaluateur afin de vérifier les similarités calculées entre le cas cible en cours et les cas sources de la liste, ceci en consultant les tableaux de similarités et en cas de besoin en demandant à l'agent ILCSS de recalculer ces similarités. Le résultat de cette étape d'adaptation est une liste des cas sources ordonnés par ordre croissant selon leurs ressemblances au cas cible en cours. Ensuite cette liste sera transmise à l'agent d'adaptation qui procède à la validation de l'action future appropriée au cas cible en cours de progression. Une fois la validation effectuée, l'agent d'adaptation affiche le résultat. Le tuteur valide la proposition du système, sinon il propose sa propre décision. Le diagramme de séquence suivant illustre le processus d'adaptation.

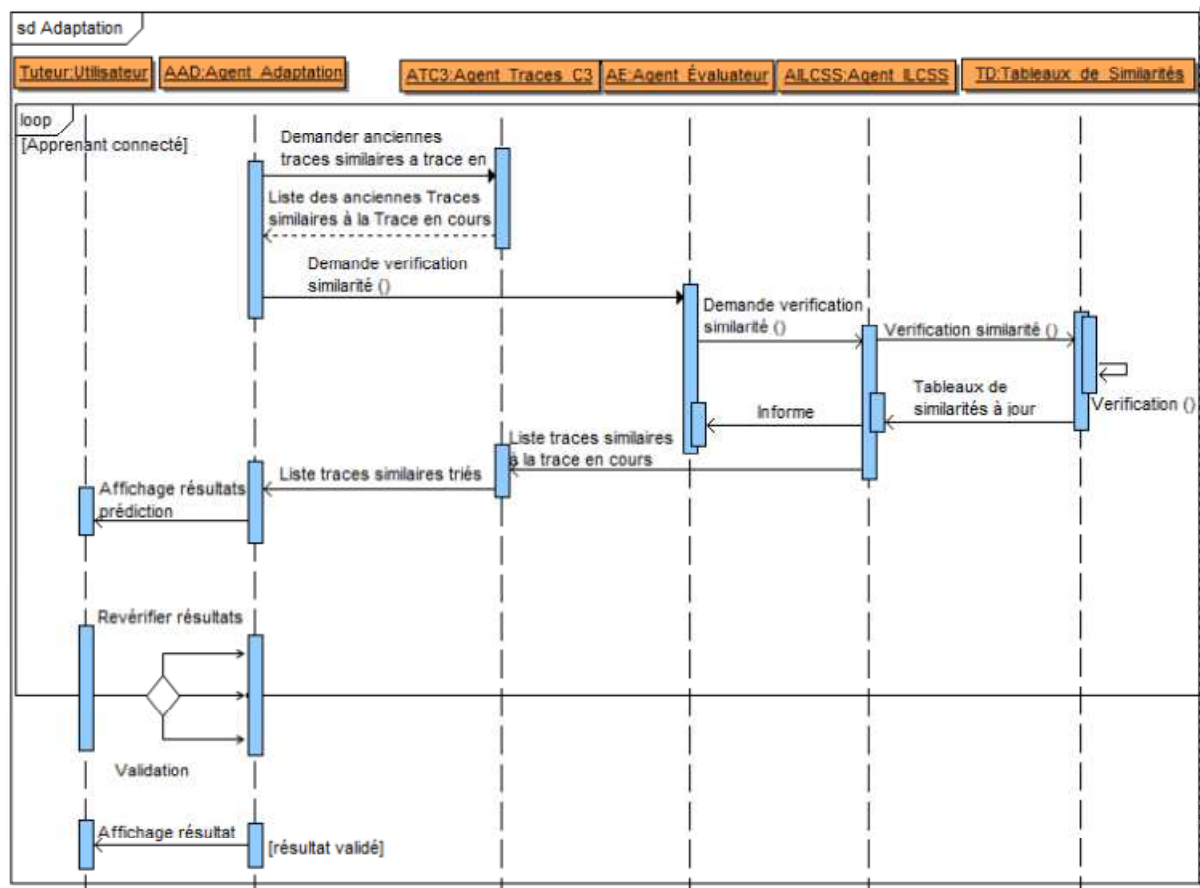


Figure 5.15 : Diagramme de séquence de la phase d'adaptation

5.6.2.4 Processus de prédiction d'une situation d'apprentissage en cours de progression

La prédiction du scénario d'apprentissage d'un apprenant dans notre système passe par plusieurs étapes. Tout d'abord l'apprenant doit s'authentifier pour accéder aux cours JADE disponible au serveur Moodle. Après vérification et validation du login et mot de passe de l'apprenant, le menu principal du serveur Moodle sera affiché. L'apprenant peut commencer à explorer librement le cours JADE. Le module d'interface Moodle/IDCBR-MAS procède à la collecte des traces de l'apprenant d'une manière permanente. Ensuite les agents de la couche élaboration récupèrent ces traces d'une façon permanente aussi et commence l'élaboration du cas cible brute (les tracés récupérés du module d'interface). Cette étape est ré-exécuter chaque fois qu'il y a un changement dans les traces de l'apprenant. Le cas cible sera ensuite envoyer à la couche de remémoration. Les agents de cette dernière procèdent au calcul des similarités entre le cas cible et chaque cas source de la base des cas, par la suite ces agents mettant à jour les tableaux de similarités et envoie la liste des cas similaires triée (ordre croissant) à la couche d'adaptation. Cette étape est ré-exécuter aussi chaque fois qu'il y a un changement dans les traces de l'apprenant. La dernière couche d'adaptation vérifie les similarités calculées précédemment avant d'envoyer au tuteur l'action future de l'apprenant proposé par le système IDCBR-MAS. La ré-exécution de cette étape est conditionnée par le changement des valeurs de similarités calculées par

la remémoration du cas cible et non pas par le changement des traces brutes. Enfin le tuteur aura la possibilité de :

- valider l'action future de l'apprenant proposée par IDCBR-MAS ;
- proposer leur propre action à exécuter par l'apprenant ;
- demander au système IDCBR-MAS de révérifier les résultats.

Le diagramme de séquence suivant illustre le processus de prédiction d'apprentissage d'un apprenant depuis le démarrage de la session d'apprentissage jusqu'à la proposition de l'action future de l'apprenant.

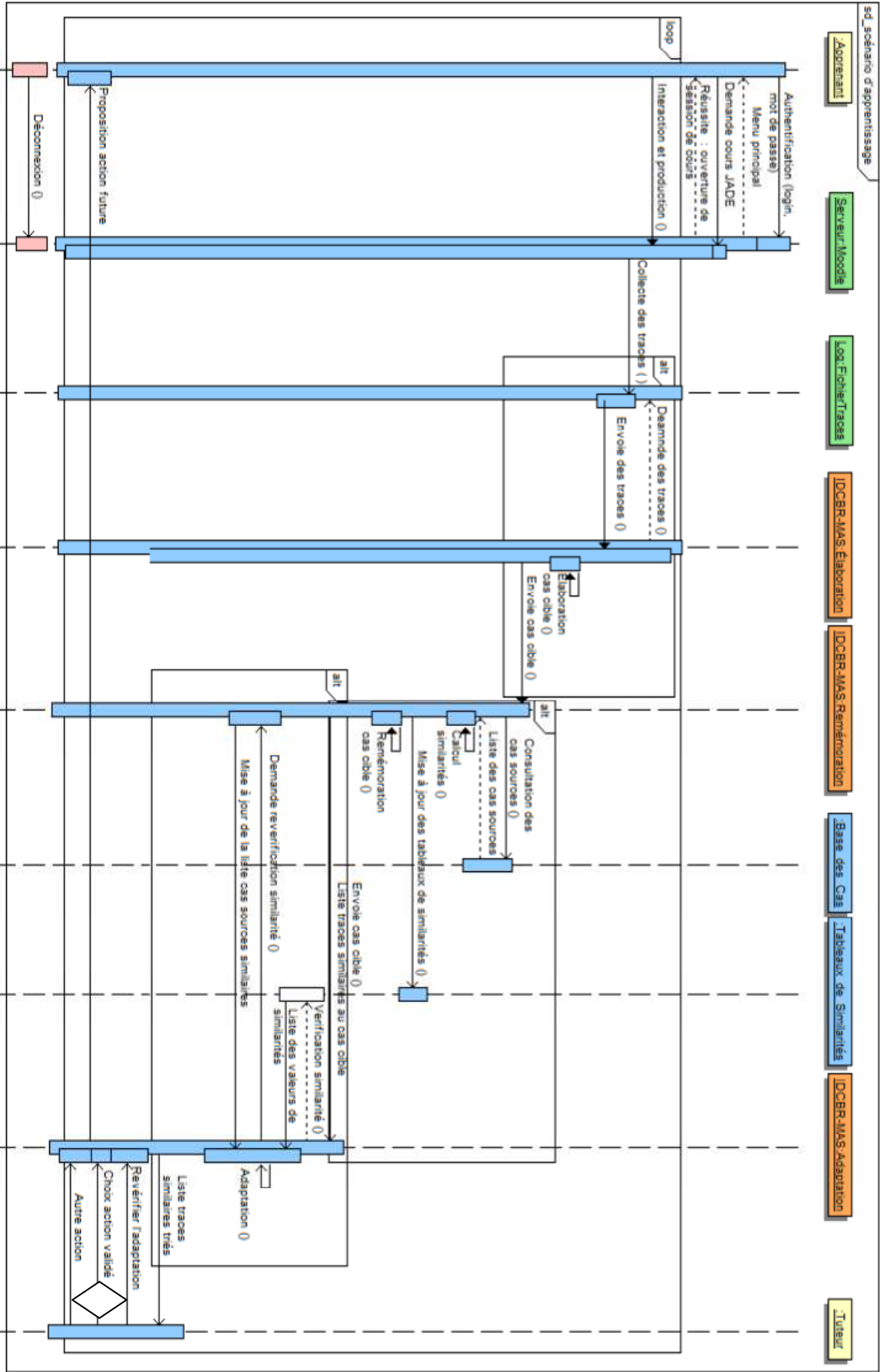


Figure 5.16 : Diagramme de séquence de prédiction d'une situation d'apprentissage

5.6.3 Cycle dynamique IDCBR-MAS appliqués au STI

Dans cette section, nous validons notre cycle RàPC dynamique (détaillé dans le chapitre précédent) et nous appliquons les étapes du cycle sur un exemple réel. Nous montrons ensuite que l'exécution de ce cycle prend en compte d'une manière dynamique le changement de la trace d'un apprenant en interaction avec Moodle. Dans notre exemple d'application, on prend un cas cible de 13 traits sémantiques et une base de cas composée de 4 cas sources (la taille de chaque cas source dépasse 30 traits sémantiques). La liste complète des cas sources sont fournis dans l'annexe A.

Soit le cas cible en format XML suivant :

```
<logs >
<user >
  <id >6</id>
  <nom >Cas cible</nom>
</user>
  <id >1</id>
  <cours >1</cours>
  <date >ven 30/5/2014 13:32</date>
  <action >login </action>
  <ip >127.0.0.1</ip>
  <info >6</info>
  <date_second>1401449562</date_second>
  <duree_second> 211</duree_second>
</log><log>
  <id >2</id>
  <cours >3</cours>
  <date >ven 30/5/2014 13:36</date>
  <action >launch</action>
  <ip >127.0.0.1</ip>
  <info >
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.ht
    ml</info>
  <date_second>1401449773</date_second>
  <duree_second> 365</duree_second>
</log><log>
  <id >3</id>
  <cours >3</cours>
  <date >ven 30/5/2014 13:42</date>
  <action >launch</action>
  <ip >127.0.0.1</ip>
  <info >
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/la_norme_fipa.
    html</info>
  <date_second>1401450138</date_second>
  <duree_second> 261</duree_second>
</log><log>
  <id >4</id>
```

```

<cours >3</cours>
<date >ven 30/5/2014 13:46</date>
<action >launch</action>
<ip >127.0.0.1</ip>
<info >
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_c
ontractnet_de_jade.html</info>
<date_second>1401450399</date_second>
<duree_second> 352</duree_second>
</log><log>
<id >5</id>
<cours >3</cours>
<date >ven 30/5/2014 13:52</date>
<action >launch</action>
<ip >127.0.0.1</ip>
<info >
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbog
age_de_jade.html</info>
<date_second>1401450751</date_second>
<duree_second> 182</duree_second>
</log><log>
<id >6</id>
<cours >3</cours>
<date >ven 30/5/2014 13:55</date>
<action >launch</action>
<ip >127.0.0.1</ip>
<info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comport
ements_des_agents_dans_la_plateforme_jade.html</info>
<date_second>1401450933</date_second>
<duree_second> 196</duree_second>
</log><log>
<id >7</id>
<cours >3</cours>
<date >ven 30/5/2014 13:58</date>
<action >launch</action>
<ip >127.0.0.1</ip>
<info >
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbog
age_de_jade.html</info>
<date_second>1401451129</date_second>
<duree_second> 161</duree_second>
</log><log>
<id >8</id>
<cours >3</cours>
<date >ven 30/5/2014 14:1</date>
<action >launch</action>
<ip >127.0.0.1</ip>
<info >
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comportements
_des_agents_dans_la_plateforme_jade.html</info>

```

```

        <date_second>1401451290</date_second>
        <duree_second> 158</duree_second>
</log><log>
    <id>9</id>
    <cours>3</cours>
    <date>ven 30/5/2014 14:4</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_c
        ontractnet_de_jade.html</info>
    <date_second> 1401451448</date_second>
    <duree_second> 214</duree_second>
</log><log>
    <id>10</id>
    <cours>3</cours>
    <date>ven 30/5/2014 14:7</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.ht
        ml</info>
    <date_second> 1401451662</date_second>
    <duree_second> 361</duree_second>
</log><log>
    <id>11</id>
    <cours>3</cours>
    <date>ven 30/5/2014 14:13</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_c
        ontractnet_de_jade.html</info>
    <date_second> 1401452023</date_second>
    <duree_second> 10</duree_second>
</log><log>
    <id>12</id>
    <cours>3</cours>
    <date>ven 30/5/2014 14:13</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/rfrences.html</
        info>
    <date_second> 1401452033</date_second>
    <duree_second> 13</duree_second>
</log><log>
    <id>13</id>
    <cours>3</cours>
    <date>ven 30/5/2014 14:14</date>

```

```
<action >launch</action>
<ip >127.0.0.1</ip>
<info >
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/bref_descriptio
n_de_jade.html</info>
<date_second> 1401452046</date_second>
<duree_second> 384</duree_second>
</log></logs>
```

Pour les quatre cas sources utilisés dans cet exemple de validation, ils sont fournis dans l'annexe A. Ci-après les deux principales étapes du cycle RàPCD appliquées dans notre exemple :

Étape d'initialisation : cette étape commence juste après l'arrivée du deuxième trait sémantique du cas cible, puisque le premier trait sémantique contient juste l'information qui concerne l'authentification de l'apprenant sur Moodle. Ensuite, le cycle se poursuit par la remémoration des cas sources les plus proches au cas cible. On applique la mesure de similarité ILCSS (détaillée dans le chapitre précédent) sur notre exemple on trouve le tableau de similarités initiales suivant :

Similarité ILCSS (Cas cible, Cas source1)	Similarité ILCSS (Cas cible, Cas source2)	Similarité ILCSS (Cas cible, Cas source3)	Similarité ILCSS (Cas cible, Cas source4)	Score élaboration	Score Remémoration	Score adaptation	Instant TD
0,666666667	0,5	1	0,5	2	1	1	t ₁

Dans cette étape les deux traces sources 2 et 4 sont les plus proches du cas cible.

Étape dynamique : cette étape est constituée de plusieurs sous-processus. L'arrivée du troisième trait sémantique constituant le cas cible provoque le démarrage de cette phase. Nous passons maintenant aux résultats de l'application de ce cycle et à l'analyse de ces résultats.

Le tableau suivant représente les similarités entre le cas cible et les cas sources de l'instant t₁ jusqu'à t₁₂, en se basant sur la mesure de similarité ILCSS.

Similarité ILCSS (Cas cible, Cas source1)	Similarité ILCSS (Cas cible, Cas source2)	Similarité ILCSS (Cas cible, Cas source3)	Similarité ILCSS (Cas cible, Cas source4)	Score élaboration	Score Remémoration	Score adaptation	Instant TD
0,666666667	0,5	1	0,5	2	1	1	t ₁
0,5	0,333333333	1	0,666666667	3	2	2	t ₂
0,4	0,5	1	0,5	4	3	3	t ₃
0,571428571	0,6	1	0,4	5	4	4	t ₄
0,571428571	0,666666667	1	0,5	6	5	4	t ₅
0,625	0,571428571	1	0,571428571	7	6	5	t ₆
0,666666667	0,5	1	0,5	8	7	5	t ₇
0,6	0,444444444	1	0,444444444	9	8	6	t ₈
0,636363636	0,5	1	0,4	10	9	6	t ₉
0,603333333	0,545454545	1	0,363636364	11	10	6	t ₁₀
0,715384615	0,583333333	0,90909091	0,333333333	12	11	6	t ₁₁
0,846153846	0,615384615	0,833333333	0,307692308	13	12	6	t ₁₂

Tableau 5.1 : Similarités du cas cible avec les cas sources

Chaque changement dans le score d'élaboration indique qu'il y a un changement dans le cas cible (l'arrivée de traits sémantiques). Dans ce cas, notre système procède à la remémoration des cas sources similaires au cas cible et le score de remémoration

s'incrémente automatiquement. Après la remémoration, si le système détecte un changement dans la liste des cas similaires alors le score d'adaptation s'incrémente automatiquement aussi. A l'instant t_{12} , le cycle passe à l'état d'attente, en attendant le changement dans la trace cible. Passons maintenant à l'analyse des résultats, le graphe 5.7 présente les similarités calculées entre le cas cible et les cas sources pour 12 instants.

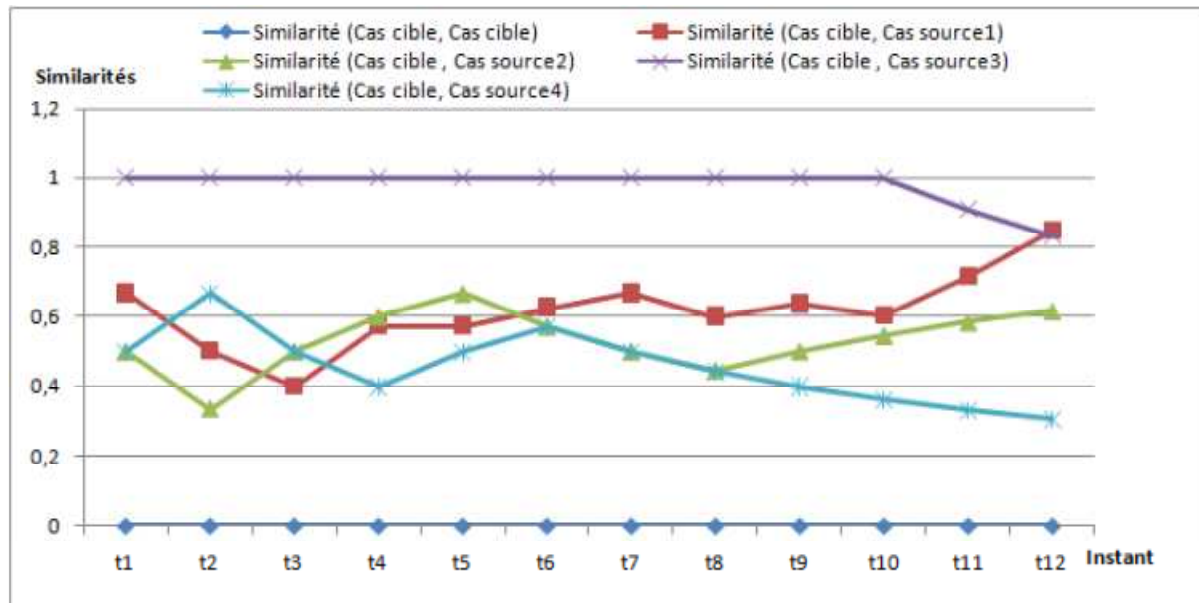


Figure 5.17 : Graphe des similarités entre cas cible et cas sources

D'après le graphe :

- 1) A l'instant t_1 , les deux cas sources 2 et 4 sont très proches du cas cible ;
- 2) A t_3 , le cycle RàPCD retourne de l'étape d'adaptation à l'étape de remémoration suite au changement du cas cible et suite aussi au changement des similarités calculées entre le cas cible et les cas sources. Le cas source1 devient le plus proche du cas cible ;
- 3) A t_4 , le cycle RàPCD retourne de l'étape d'adaptation à l'étape de remémoration suite au changement du cas cible et le cas source 4 devient le plus proche ;
- 4) A t_6 , un autre retour du cycle RàPCD de l'étape d'adaptation à l'étape de remémoration et les deux cas sources 2 et 4 deviennent les plus proche ;
- 5) A t_8 , le cycle retourne de l'adaptation à la remémoration, et le cas source 4 devient le cas source le plus proche du cas cible avec une valeur de similarité égale à 0,3. Cette situation n'a pas changé jusqu'à ce que le système passe en état d'attente à l'instant t_{12} .

Le graphe 5.7 montre le score de chaque étape du cycle RàPCD : le nombre de fois que le cycle a exécuté cette étape.

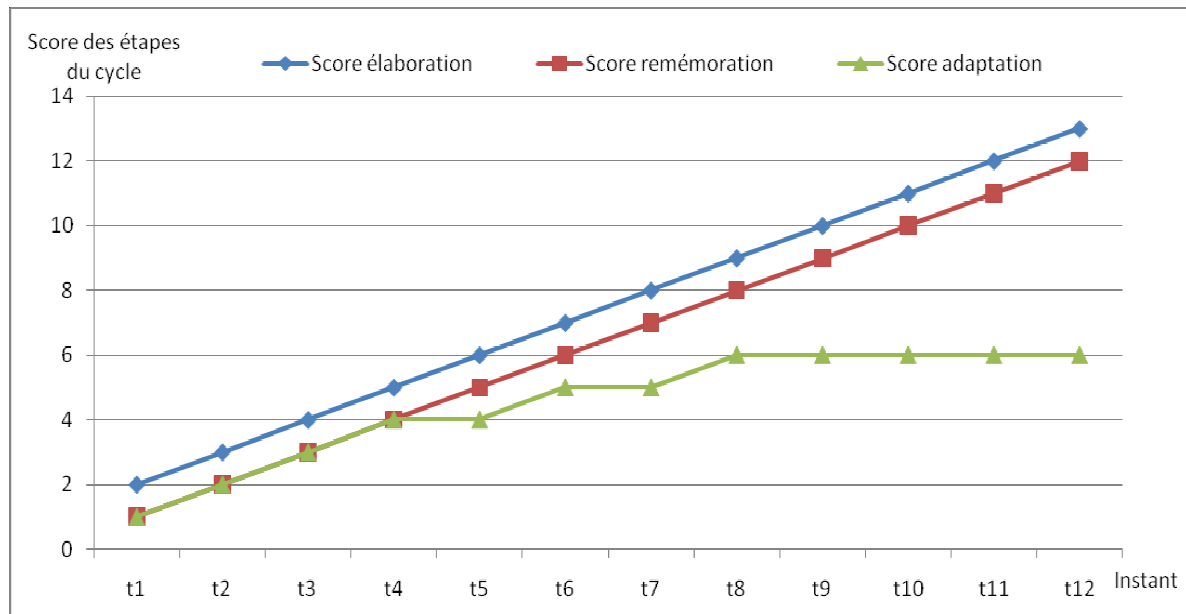


Figure 5.18 : Graphe des scores des étapes du cycle RaPCD

D'après le graphe on remarque que le score d'élaboration qui représente le nombre de fois que le cas cible a changé de valeurs, représente le score le plus élevé suivi du score de la remémoration, ce qui est logique puisque à chaque changement dans la situation le système recalcule les similarités entre le cas cible et les cas sources. Or la dernière valeur du score d'adaptation est égale à six, ceci veut dire que pour douze remémoration du cas cible, l'étape d'adaptation a été exécutée seulement six fois et que la liste des cas sources similaires au cas cible n'a pas subi assez de changement. Si on applique le cycle RaPC classique sur le même exemple alors pour toutes les étapes on aura 12 comme valeur des scores, et on aura aussi pour chaque exécution du cycle une liste des cas similaires différente et donc pour un instant donné on a plusieurs listes de cas sources similaires alors avec notre cycle il y a moins de calcul et pour chaque instant il y a une seule liste des cas similaires. Par conséquent on gagne au niveau du temps de calcul et la qualité de solution et on présente au décideur une seule liste de choix de cas sources similaires.

Enfin l'objectif principal de l'application de notre approche dans le e-Learning est de prédire à un instant donné la prochaine action de l'apprenant. Dans notre exemple et en consultant le cas source4 le plus proche du cas cible à t_{12} alors on peut dire que la prochaine action de notre apprenant est la consultation du paragraphe `outils_de_dbogage_de_jade.html`.

Dans la section suivante, nous exposons les principales interfaces de notre application web et nous procédons à l'analyse des résultats issus de la validation de notre approche.

5.7 Application Web

5.7.1 Scénario d'apprentissage

Notre application web offre les fonctionnalités suivantes :

- **pour l'apprenant** : s'inscrire sur la plateforme IDCBR-MAS puis sur Moodle afin de suivre la formation sur JADE ou un autre cours, et interagir avec le système et le tuteur.
- **pour le tuteur** : s'inscrire sur la plateforme, création et mise à jour du contenu pédagogique sur Moodle, et suivre en temps réel ou en temps différé l'évolution des apprenants. L'application permet au tuteur d'évaluer les résultats des apprenants et de valider les suggestions proposées par le système.
- **pour l'administrateur** : administrer l'application.

La figure ci-dessous illustre les différentes opérations pouvant être réalisées par les tuteurs.

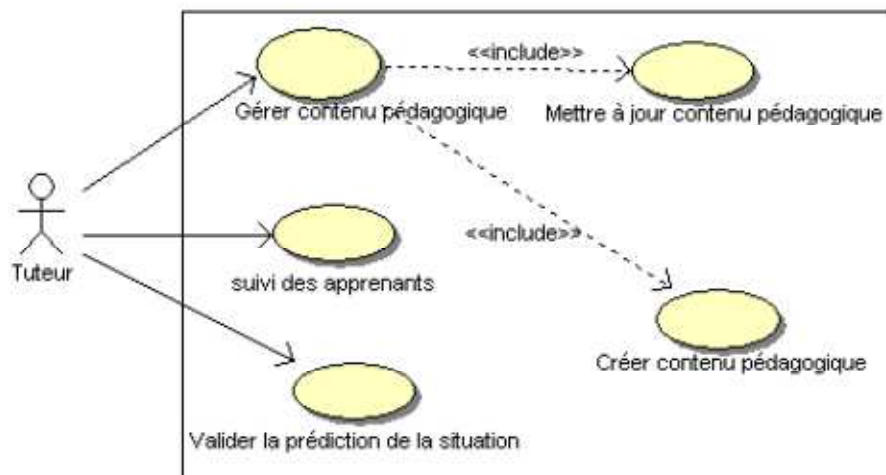


Figure 5.19 : Cas d'utilisation d'un tuteur

Afin de bénéficier des fonctionnalités offertes par IDCBR-MAS, le tuteur est amené à s'inscrire. Une fois inscrit, le tuteur aura la possibilité d'effectuer les tâches suivantes :

- création, et mise à jour du contenu pédagogique sur Moodle ;
- gestion des contenus pédagogiques tels que le dépôt des devoirs, QCM, désactivation, suppression des contenus pédagogiques ;
- le suivi en temps réel de la progression des apprenants ;
- la validation du choix des traces, proposées par IDCBR-MAS, similaires à la trace en progression ;

cas	Gérer contenu pédagogique
résumé	Création, gestion des contenus pédagogiques
Acteur primaire	Tuteur
Acteur secondaire	Apprenant, Administrateur
Pré-conditions	-
Résultats	Système opérationnel

Description	Création et gestion des contenus pédagogiques
Exceptions	

De la même manière les apprenants doivent s'inscrire sur IDCBR-MAS, ensuite s'inscrire sur Moodle pour bénéficier des cours et contenus pédagogiques offerts par les tuteurs. Le diagramme suivant illustre les différentes fonctionnalités destinées aux apprenants.

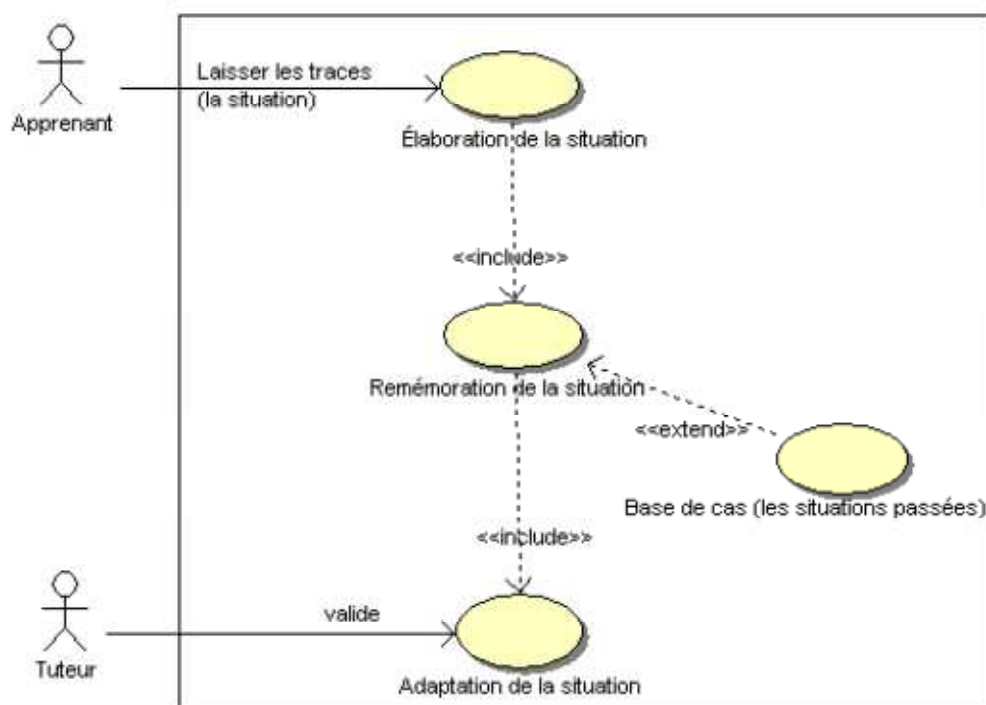


Figure 5.20 : Cas d'utilisation d'un apprenant

cas	Laisser les traces
résumé	Enregistrement des traces de l'apprenant
Acteur primaire	Apprenant
Acteur secondaire	Tuteur, Administrateur
Pré-conditions	-
Résultats	Fichier Traces de l'apprenant
Description	- L'apprenant se connecte à Moodle ; - Interaction de l'apprenant avec Moodle ; - Enregistrement des productions, actions et toutes interactions de l'apprenant avec Moodle dans le fichier Log du serveur (serveur d'hébergement de la Moodle).
Exceptions	-

cas	Élaboration du cas cible
résumé	Elaboration du cas cible en se basant sur les traces

	laissées par l'apprenant (l'interaction de l'apprenant avec Moodle).
Acteur primaire	Apprenant
Acteur secondaire	Tuteur, Administrateur
Pré-conditions	-
Résultats	Trace d'apprenant élaborée
Description	- Récupération des traces de l'apprenant via le module, interface que nous avons ajoutée à Moodle ; - Encapsulation des traces de l'apprenant par les agents traces de IDCBR-MAS.
Exceptions	- Fichiers traces de l'apprenant vide.

cas	Remémoration du cas cible
résumé	Remémoration du cas cible en se basant sur les cas similaires stockés dans la mémoire
Acteur primaire	Apprenant
Acteur secondaire	Tuteur, Administrateur
Pré-conditions	-
Résultats	Trace d'apprenant remémoré
Description	- Calcul des similarités entre le cas cible et les expériences passées stockées dans la mémoire du système ; - Mise à jour des similarités calculées dans la première étape chaque fois qu'il y a un changement du cas cible ; - Mise à jour de la liste des expériences passées similaires au cas cible.
Exceptions	- Fichier traces de l'apprenant vide. - La base de cas vide.

cas	Adaptation du cas cible
résumé	Adaptation du cas cible en se basant sur les situations similaires stockées dans la mémoire
Acteur primaire	Tuteur
Acteur secondaire	Administrateur
Pré-conditions	-
Résultats	Scénario du comportement futur de l'apprenant, profil de l'apprenant
Description	- Mise à jour de la liste des cas sources similaires au cas cible en se basant sur le cycle RàPCD ; - Choix du cas source le plus similaire au cas cible ; - Validation du choix.
Exceptions	Pas de prédiction si la base ne contient pas de cas sources similaires au cas cible, dans ce cas le tuteur intervient.

IDCBR-MAS est géré par un administrateur qui a pour rôle la gestion des comptes tuteurs et apprenants. La figure suivante illustre le diagramme de cas d'utilisation de l'administrateur.

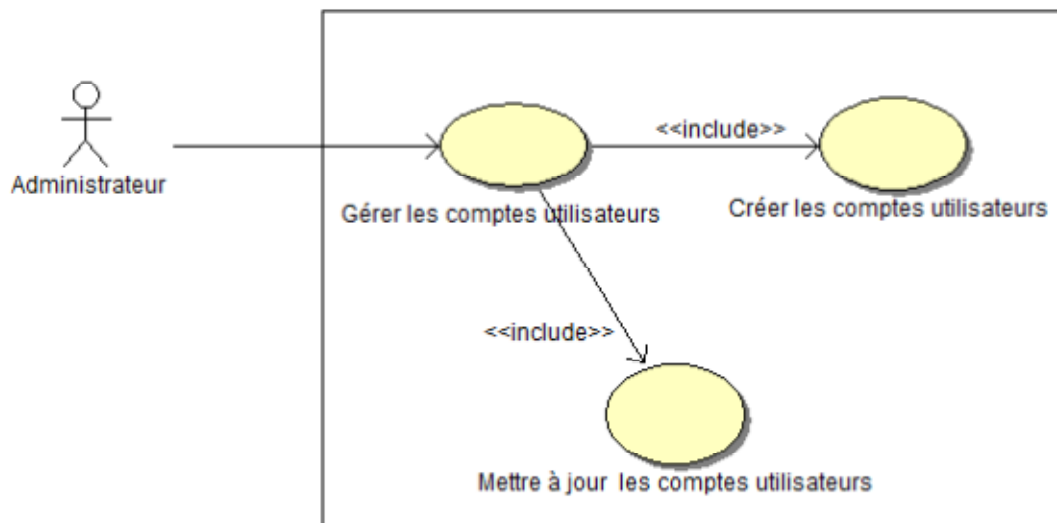


Figure 5.21 : cas d'utilisation de l'administrateur

cas	Gérer compte utilisateur
résumé	Création, mise à jour et suppression des comptes utilisateurs.
Acteur primaire	Administrateur
Acteur secondaire	Apprenant, Tuteur
Pré-conditions	-
Résultats	Système opérationnel
Description	Création et gestion des comptes des apprenants et tuteurs.
Exceptions	-

5.7.2 Génération de similarités entre le cas cible et cas sources

Après l'inscription d'un apprenant sur IDCBR-MAS (même démarche que l'inscription du tuteur), l'apprenant aura la possibilité de lancer Moodle à partir de notre plateforme et lancer par la suite une session d'apprentissage. Le tuteur suit l'apprenant au fur et à mesure de son apprentissage. Toutes les interactions, actions et productions de l'apprenant sont enregistrées sur le fichier Log et dans la base des données Moodle. IDCBR-MAS récupère ces traces via les agents d'interfaces d'une manière permanente, ensuite ces dernières seront traitées par l'agent Agent_ILCSS. La figure 5.22 présente pour chaque apprenant connecté à Moodle (cas cible), les similarités calculées, par les agents IDCBR-MAS, entre le cas cible et les cas sources. Les valeurs de similarités sont mises à jour automatiquement chaque fois qu'il y a changement dans le fichier trace de l'apprenant (fichier Log).

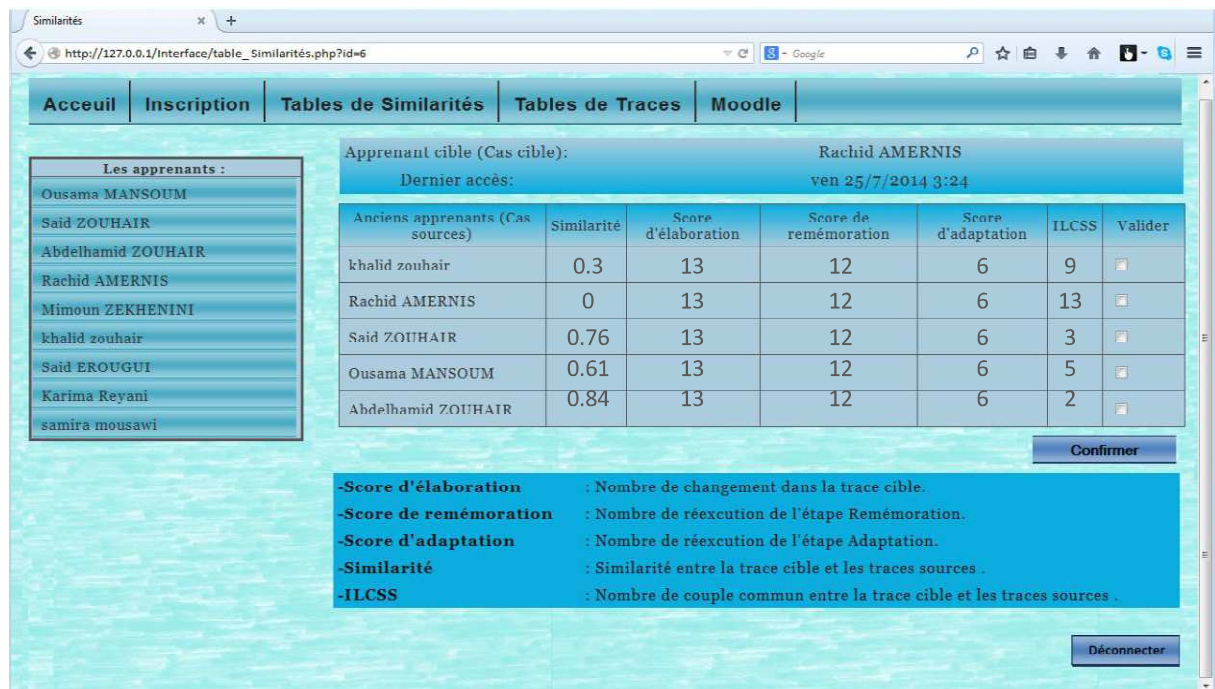


Figure 5.22 : Similarités entre la trace cible et les traces passées [Zouhair et al., 2014]

La colonne score d'élaboration désigne le nombre des changements du cas cible. Les similarités entre le cas cible et les cas sources sont calculées par l'agent ILCSS. La colonne score de remémoration montre le nombre de fois que l'agent ILCSS a recalculé ces similarités suite au changement des données du cas cible. Ces similarités serviront comme éléments fondamentaux dans la prédiction de la situation accomplie par l'agent d'adaptation. Le système propose au tuteur une liste des cas les plus proches au cas cible afin de choisir le meilleur cas similaire. La colonne score d'adaptation présente le nombre de fois que le système a changé la liste proposé au tuteur. Enfin le tuteur peut choisir via le bouton valider un des cas similaires comme il peut demander au système de continuer sa comparaison du cas cible.

Remarque :

Dans la figure 5.22, la table des similarités contient seulement les apprenants qui ont suivi une formation sur Moodle (traces non vides).

5.7.3 Courbe des similarités

La figure 5.23 affiche les courbes des similarités entre le cas cible et les cas sources similaires. Ces courbes sont générées à partir des résultats issus de la phase de remémoration. Le traçage de ces courbes est basé sur les tableaux de similarités remplis au fur et à mesure. Ces courbes affichent l'historique des similarités. Le tuteur peut se baser sur ces courbes afin de prendre sa décision et choisir le cas source le plus similaire au cas cible.

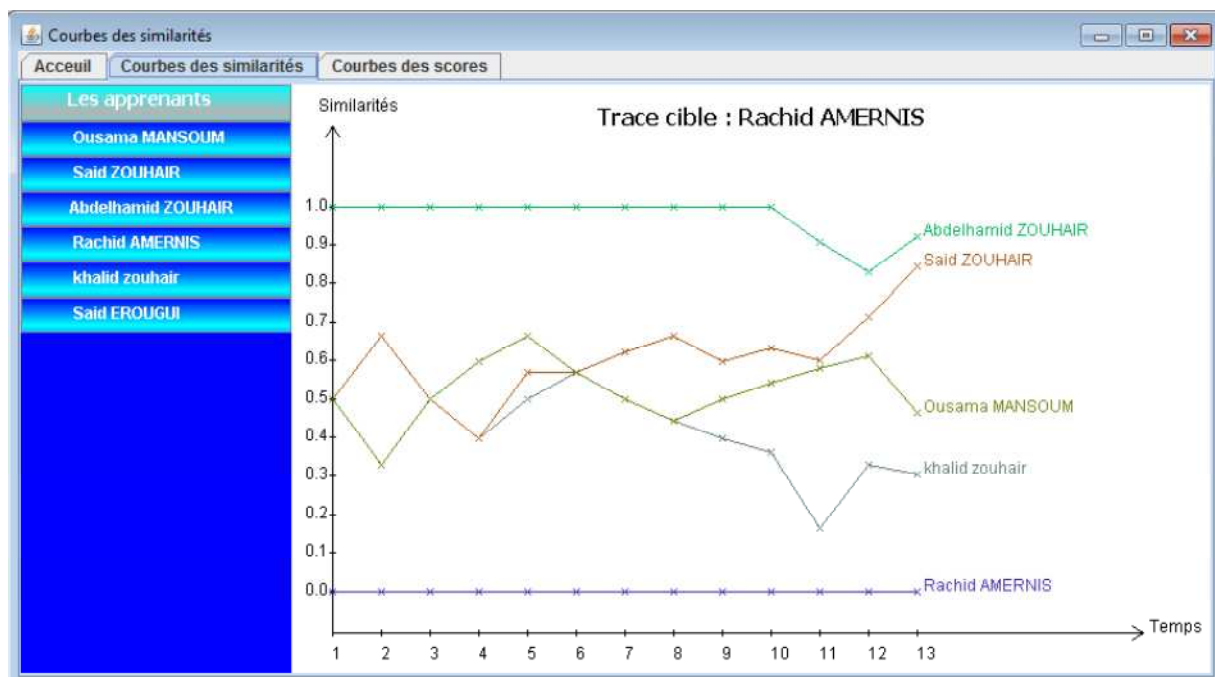


Figure 5.23 : Courbe des similarités [Zouhair et al., 2014]

5.7.4 Le cas source le plus similaire

Le tuteur sélectionne, en cliquant sur le bouton valider, le cas source le plus proche du cas cible en utilisant pour cela la table et les courbes de similarités, le choix du tuteur est transmis à l'agent d'adaptation. Ce dernier effectue l'affichage du cas source validé par le tuteur sur la page principale du cas cible. Le cas source affiché contient deux parties : une partie passée (partie couleur rouge dans la figure 5.24) qui représente les traces passées en commun entre le cas cible et le cas source similaire validé par le tuteur et une partie future (partie couleur vert sur la figure 5.24) qui représente la prochaine action de l'apprenant (l'action prédite par le système IDCBR-MAS et validée par le tuteur). Pour notre exemple la figure 5.24 montre que la prochaine action de l'apprenant est la consultation du paragraphe `outils_de_dbogage_de_jade.html`.

Apprenant cible (Cas cible):		Rachid AMERNIS
Dernier accès:		ven 25/7/2014 3:24
	Action : partie consultée	Durée (S)
Passé	login	7
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html	365
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/la_norme_fipa.html	261
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet_de_jade.html	352
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jade.html	182
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comportements_des_agents_dans_la_plateforme	196
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jade.html	161
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comportements_des_agents_dans_la_plateforme	158
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jade.html	161
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comportements_des_agents_dans_la_plateforme	158
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet_de_jade.html	214
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html	361
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet_de_jade.html	10
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/rfres.html	13
Future	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/bref_description_de_jade.html	384
	http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jade.html	300

Figure 5.24 : Choix du cas source le plus similaire [Zouhair et al., 2014]

5.8 Conclusion

Nous avons présenté dans la première partie de ce chapitre, le domaine d'application de notre approche à savoir e-learning dans l'objectif de proposer un système tutoriel intelligent.

Nous rappelons que le e-learning n'est qu'une application parmi d'autre pour valider notre approche IDCBR-MAS. On peut appliquer notre approche du RàPCD dans d'autres domaines tels que e-commerce, web, etc. La même remarque pour le STI proposé, il est indépendant de l'objet de l'étude. Dans le cadre de cette thèse, nous avons choisi un cours sur JADE comme objet d'apprentissage.

Notre architecture est indépendante du domaine d'application, nous pouvons l'adapter facilement à de nouveaux besoins, sans avoir recours à de grands changements de conception et de développement. Par exemple :

- si on change le cours JADE par un autre cours en mathématique ou cours en marketing mais on travaille toujours avec la plateforme Moodle, alors on change rien dans notre application IDCBR-MAS pour prédire l'apprentissage d'un apprenant ;
- si on change Moodle par une autre plateforme, dans ce cas on doit modifier le module d'interface mais on change rien dans la base des cas et dans les agents IDCBR-MAS ;
- si on change le domaine e-learning par e-commerce par exemple, alors on doit changer le module d'interface entre l'application avec IDCBR-MAS, on doit changer aussi la structure de la base des cas mais rien changer dans les agents IDCBR-MAS

Donc les agents IDCBR-MAS et la mesure de similarité sont indépendants du domaine d'application. Par contre la structure de traces, base des cas et module d'interface dépend du domaine d'application.

Conclusion générale

Les travaux présentés dans ce mémoire de thèse se situent dans le domaine d'aide à la décision, nous avons proposé une approche de raisonnement à partir de cas dynamique à base d'agents pour les situations dynamiques. Cette approche assure la prédiction des situations dynamiques, en se basant sur les expériences passées similaires, tout en prenant en considération le changement dynamique de la situation à prédire d'une manière incrémentale.

En se basant sur le raisonnement à partir de cas et le paradigme multi-agents, nous avons proposé un cycle dynamique fondé sur une mesure de similarité dynamique, capable d'évaluer la similarité entre une situation dynamique en cours de progression et des expériences stockées dans la base des cas. La principale différence entre le cycle RàPC classique et le cycle dynamique que nous avons proposé réside dans la gestion dynamique des étapes en passant d'un cycle linéaire à un cycle en étoile où on peut en cas de changement du cas cible revenir à une étape précédente. Ceci sans refaire le calcul de similarité entre le cas cible et les cas sources. Nous avons proposé des changements dans les processus internes des étapes du cycle RàPC, plus particulièrement nous avons proposé une nouvelle mesure de similarité capable d'évaluer la similarité d'une manière dynamique et incrémentale. Au départ, le système procède à une évaluation complète de la similarité entre le cas cible en progression et les cas sources. En cas de changements du cas cible, le système ne fait qu'une évaluation partielle ne prenant en compte que les changements du cas cible.

Le système Incremental Dynamic Case Based Reasoning- Multi-Agent System (IDCBR-MAS) que nous avons proposé est spécialement prévu pour prédire la suite d'une situation à partir de son commencement observé. Au fil du temps, la situation à prédire se précise plus et donc au lieu d'exécuter le cycle RàPC à chaque fois, la résolution se base sur ce qui avait été calculé à l'instant précédent et exécuter le cycle seulement à partir de l'étape de cycle correspondante en se basant sur la mesure de similarité dynamique. L'efficacité et l'adaptabilité de l'IDCBR-MAS ont été démontrées par une application dans le domaine du e-Learning. Notre approche se base sur des traces de l'apprenant stockées dans le serveur d'apprentissage pour aider un tuteur ou un expert à adapter l'apprentissage à l'apprenant et avant tous de prédire le comportement de ce dernier. Les résultats obtenus lors des expérimentations sont très encourageants au niveau temps de calculs.

Les contributions majeures de nos travaux de recherches dans le cadre de cette thèse de doctorat peuvent être résumées dans les points suivants :

- Proposition d'une approche pour le raisonnement à partir de cas dynamique dédiée à la prédiction des situations dynamiques. On a proposé un cycle capable de gérer les situations dynamiques : modification au niveau de l'ordre des étapes du cycle RàPC ainsi que des changements au niveau des processus internes de ces étapes.

- Proposition d'une approche pour évaluer la similarité entre une situation en cours de développement et des situations passées (mesure dynamique).
- Développement d'un tuteur intelligent basé sur le raisonnement à partir de cas dynamique. Il s'agit d'un système adaptatif capable d'initier l'apprentissage et d'assurer un suivi individualisé de l'apprenant. Notre approche permet de représenter, de suivre et d'analyser l'évolution d'une situation d'apprentissage afin d'anticiper un éventuel abandon de l'apprenant ou une démotivation de celui-ci. Notre architecture est indépendante du domaine d'application, nous pouvons l'adapter facilement à de nouveaux besoins.

Plusieurs pistes de recherche peuvent être envisagées à la suite de ce travail de recherche, dont certaines sont en cours de développement au sein de notre équipe de recherche. Nous citons ci-après quelques-unes :

- Il serait très intéressant d'approfondir le point de plusieurs sources de traces afin de déterminer une méthode de représentation de traces un peu plus générique. Il serait très appréciable que notre système soit capable de traiter des traces de plusieurs sources et de formes différentes (sites hétérogènes de e-commerce par exemple).
- Améliorer notre mesure de similarité en prenant en considération l'ontologie du domaine afin de comparer des traces de sources différentes, par exemple des traces des apprenants enregistrées dans deux plateformes d'apprentissage différentes par exemple Moodle et WebCT [WebCT, 2014].
- Améliorer la phase d'adaptation utilisée dans notre cycle RàPC, en utilisant d'autres formules et règles d'adaptation générative, transformationnelle et hybride.
- Développer l'aspect d'apprentissage pour l'ensemble des agents formant notre plateforme.

Références Bibliographiques

A

[Aamodt et Plaza, 1994] Aamodt A., Plaza E., Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches. AI Communications, pp 39-59, 1994.

[Aamodt, 2004] Aamodt A., Knowledge-Intensive Case-Based Reasoning and Sustained Learning. Proc. of the 9th European Conference on Artificial Intelligence, ECCBR'04, Lecture Notes in Artificial Intelligence, pp.1-15, Springer, 2004.

[Abed et al., 2013], M. Abed, I. Charfeddine, M. Benaïssa et Marta Starostka-Patyk : « Intelligent Traceability System of Containerized Goods». Applied Mechanics and Materials Vol. 309 pp 241-251, 2013.

[Adla, 2010], Abdelkader Adla, "Aide à la Facilitation pour une prise de Décision Collective Proposition d'un Modèle et d'un Outil", Ecole doctorale : Mathématiques, Informatiques et Télécommunications de Toulouse, l'Université Toulouse III - Paul Sabatier, 2010.

[AgentTool, 2013], AgentTool, <http://agenttool.cis.ksu.edu/>, dernière consultation du site 26/08/2013.

[Aich, 2007], Aich A., Reconnaissance et prédiction de situations dynamiques : application à l'assistance de personnes handicapées moteur, Thèse de doctorat, Université de Technologie de Troyes, 2007.

[Althoff et al., 1995] Althoff K.D., Auriol E., Bergmann R., Breen S., Dittrich S., Johnston R., Manago M., Traphöner R., Wess S., Case-based reasoning for decision support and diagnostic problem solving: The INRECA Approach. Proc. of the 3rd German Workshop on CBR, University of Kaiserslautern, 1995.

[Althoff et Bartsch-Spörl, 1996] Althoff K.D., et Bartsch-Spörl B., Decision support for case based application, Wirtschaftsinformatik, ISSN 0937-6429, 1996, vol. 38, no1, pp. 8-16, 1996.

[Althoff, 2001] Althoff K.D., Case-Based Reasoning. S.K. Chang (Ed.), Handbook on Software Engineering and Knowledge Management, pp. 549-588, 2001.

[Azaiez, 2007], Selma AZAIEZ, Approche dirigée par les modèles pour le développement de systèmes multi-agents, Thèse de l'université de Savoie, 2007.

[AFNOR, 1987] AFNOR, Vocabulaire de la documentation, 2e édition, 1987, siteweb consulté le 05 mai 2014, <http://www.afnor.org/>

[AWT, 2014], AWT (Agence Wallonne des Télécommunications) Qu'est-ce que l'e-learning ? [archive], consultée 20-06-2014.

[AgentBuilder, 2014] <http://www.agentbuilder.com/>(consulté le: 01/03/2014)

B

[Bareiss, 1988] Bareiss E.R., « PROTON : A Unified Approach to Concept Representation, Classification and Learning », PhD Thesis, University of Texas, 1988.

- [Bareiss et al., 1993] Bareiss E.R., Porter B. et Wier C., PROTOS: un système apprenti utilisant des cas typiques. *Apprentissage symbolique : une approche de l'intelligence artificielle*, Vol. 1, pp. 105-120, 1993.
- [Baptiste, 2011], Baptiste CABLÉ, « Vers la gestion de scénarios pour la reconnaissance et l'anticipation de situations dynamiques », Université de Technologie de Troyes, France, 2011.
- [Bauer et al., 2001], Bernhard Bauer, Jörg P. Müller, J. Odell, « Agent UML: A Formalism for Specifying Multiagent Software Systems », *International Journal of Software Engineering and Knowledge Engineering*, Vol. 11, No. 3, pp.1-24, 2001.
- [Bauer et Odell, 2005], Bauer, B., Odell, J. UML 2.0 and agents: how to build agent-based systems with the new UML standard. *Engineering Applications of Artificial Intelligence*, vol. 18, pp: 141-157, 2005.
- [Bergmann et al., 2003] Bergmann R., Althoff K.D., Breen S., Göker M., Manago M., Traphöner R. et Weiss S., *Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology. Lecture Notes in Artificial Intelligence, LNAI 1612*, Springer-Verlag, Berlin, 2003.
- [Benjamin et Matthieu, 2004], Benjamin DEVÉZE et Matthieu FOUQUIN, *CASE-BASED REASONING*, EPITF, 14-16 rue Voltaire 94270 Kremlin-Bicêtre, France, 2004.
- [Bellifemmine et al., 1999], F. Bellifemmine, A. Poggi et G. Rimassa: « JADE - A FIPA compliant agent framework ». In 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents, 1999.
- [Bernon et al., 2005], C. Bernon, M. Cossentino et J. Pavon: « An Overview of Current Trends », in *European AOSE Research. Informatica (Slovenia)*, pages 379–390, 2005.
- [Bernon et al., 2009], Carole Bernon, Marie-Pierre Gleizes, Gauthier Picard. Chapitre 2 : Méthodes orientées agent et multi-agent. Dans : *Technologies des systèmes multi-agents et applications industrielles*. Amal El FallahSeghrouchni, Jean-Pierre Briot (Eds.), Lavoisier, p. 44-76, IC2, avril 2009.
- [Bohanec, 2008], Bohanec, M., DEXi: Program for multi-attribute decision making, User's manual, Version 3.03. IJS Report DP-9989, Jožef Stefan Institute, Ljubljana. 2008 Available at: <http://kt.ijs.si/MarkoBohanec/pub/DEXiManual30r.pdf>
- [Boissier et al., 2011], Olivier Boissier, Flavien Balbo, and Fabien Badeig. Controlling multi-party interaction within normative multi-agent organizations. In *Proceedings of the 6th international conference on Coordination, organizations, institutions, and norms in agent systems*, pages 357–376, Berlin, Heidelberg, 2011. Springer-Verlag.
- [Bratman et al., 1988] M.E. Bratman, D.J. Israel, and Martha E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4) :349–355, 1988.
- [Brandau, 1991], Brandau R., Lemmon A., Lafond C.: Experience with extended episodes : cases with complex temporal structure, *Workshop on case-based reasoning*, Washington, DC, San Mateo, pp. 1-12, CA : Morgan Kaufman, may 1991.
- [Bull et al., 1997] Bull M., Kundt G., Gier L.: Case-based risk detection and forecasting in ageographic-medical system, *German workshop on CBR*, pp. 59-64, in R. Bergman and M. Wilke editors, march 1997.
- [Bond, 1988], A. Bond et L. Gasser. *Readings in distributed artificial intelligence*. Morgan Kaufman, 1988.

[Boukachour, 2002], Boukachour H., Système de veille préventive pour la gestion de situations d'urgence: une modélisation par organisations d'agents. Application aux risques industriels. PhD Thesis, University of Le Havre, 2002.

[Brusilovski, 1997], Brusilovski P., Distributed Intelligent Tutoring on the Web. 8th World Conference of Artificial Intelligence in Education. Du Boulay, B.; Mizoguchi, R. (Eds). IOS Press. pp.482-489, 1997.

[Brownell, 2002], Brownell, D. (Ed.), SAX2. O'Reilly & Associates, Inc. 2002.

C

[Charlet et al., 2000] Charlet J., Zacklad M, Kassel G. et Bourigault D., Knowledge engineering: Recent evolutions and new challenges, Editions Eyrolles, Paris, 2000.

[Charras et Lecroq, 1998], Charras C., Lecroq T., Sequence comparison. www.wigm.univ-mlv.fr/lecroq/seqcomp/index.html, 1998.

[Chella et al., 2004], A. Chella, M. Cossentino, et L. Sabatucci: « Tools and patterns in designing multiagent systems with PASSI ». In 6th WSEAS International Conference on Telecommunications and Informatics, Cancun, Mexico, 2004.

[CNTRL, 1979] La Recherche, janv. 1979, N° 96, vol. 10, p. 61, cité par le dictionnaire du CNTRL (Centre National de Ressources Textuelles et Lexicales), 1979.

[Collis et al., 1998], J. C. Collis, D. T. Ndumu, H. S. Nwana, et L.C. Lee: « The ZEUS agent building toolkit ». BT Technology Journal, 16(3), 1998.

[Cordier et al., 2007] Cordier A., Fuchs B., Lieber J. et Mille A., Acquisition interactive des connaissances d'adaptation intégrée aux sessions de raisonnement à partir de cas : Principes, architecture IAKA et prototype KAYAK, 2007.

[Corvaisier et al., 1997] Corvaisier S. F., Mille A. et Pinon J. M., Information retrieval on the world wide web using a decision making system. Actes de la conférence RIAO'97, Centre des hautes études internationales d'Informatique, Montréal, pp. 285-295, 1997.

[Coutinho et al., 2005] L.R. Coutinho, J.S. Sichman, and O. Boissier. Modeling Organization in MAS : A comparison of Models. 2005.

[Corvaisier et al., 1998] Corvaisier F., Mille A., Pinon J.M. Radix 2, assistance à la recherche d'information documentaire sur le web. In IC'98, Ingénierie des Connaissances, Pont-à-Mousson, France, INRIALORIA, Nancy, 1998, p. 153-163.

[Cossentino et Potts, 2001], M. Cossentino et C. Potts: « PASSI a process for specifying and implementing Multi-Agent Systems ». Using UML, 2001.

[Cossentino et al., 2005], M. Cossentino, C. Bernon et J. Pavón: « Modelling and Metamodelling Issues in Agent Oriented Software Engineering: The AgentLink AOSE TFG Approach ». Report of the AOSE TFG Ljubljana meeting, 2005.

[Cristian Pinzón et al., 2011] Pinzón Cristian I., Javier Bajo, Juan F., Juan M. Corchado b, S-MAS: An adaptive hierarchical distributed multi-agent architecture for blocking malicious SOAP messages within Web Services environments, Expert Systems with Applications Volume 38, Issue 5, May 2011, Pages 5486-5499, <http://www.sciencedirect.com/science/article/pii/S0957417410012327>

[Coulon et al., 1990], D. Coulon, J.-F. Boivieux, L. Bourrelly, L. Bruneau, E. Chouraqui, J.-M. David, C. R. Lu, M. Py, J. Savelli, B. Séroussi et C. Vrain. Le raisonnement par analogie en intelligence artificielle. Dans Actes des 3èmes journées nationales du PRC-GDR Intelligence Artificielle, rédacteur B. Bouchon-Meunier, pages 45–88, 1990.

[Cursus, 2011], <http://cursus.edu/>, article disponible sur <http://cursus.edu/article/17799/modalites-formation-tutorat-distance-etude-comparative/#.U9Yr9bHIcq4> (dernière consultation le 28 juillet 2014).

[Croock et al, 2005], De Croock, M., Mofers, F., Van Veen, M., Van Rosmalen, P., Brouns, F., Boticario, J., Barrera, C., Santos, O., Ayala, A., Gaudio, E., Hernández, F., Arana, C., & Trueba, I. (2002). D12-State-of-the-art, Alfabet Project Deliverable, Heerlen: Open University Netherland, October 31, 2005, <http://hdl.handle.net/1820/94>.

[Coffield et al., 2004], Coffield, F., Moseley, D., Hall, E., & Ecclestone, K. (2004). Learning styles and pedagogy in post-16 learning: A systematic and critical review.

[Constantino et al., 2001], Constantino G., Suthers D., Icaza J-I., Designing and Evaluating a Collaboration Coach: Knowledge and Reasoning. Proceedings of the Artificial Intelligence in Education, AI-ED 2001, J.D. Moore et al (Eds). The Netherlands: IOS Press. pp. 176-187.

[Cardelli, 1999], L. Cardelli: « Abstractions for Mobile Computation ». Lecture Notes in Computer Science, vol. 1603. Springer-Verlag, 51–94, 1999.

[Croset, 2009], CROSET, Marie-Caroline. Modélisation des connaissances des élèves au sein d'un logiciel d'algèbre. Études des erreurs stables inter-élèves et intra-élève en termes de praxis-en-acte. 2009. Thèse de doctorat. Université Joseph-Fourier-Grenoble I.

D

[Davenport 2000] T.H. Davenport et L. Prusak. Working knowledge : How organizations manage what they know. Harvard Business School Press, Boston, 2000.

[Danflos, 2000], D. Danflos : « Déploiement National des Systèmes d'Information Multimodale: DELFI, l'exemple Allemand ». Rapport technique, CERTU (Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques), Aout 2000.

[Deloach, 1999], Deloach, S.A., 1999. *Multiagent Systems Engineering : A Methodology And Language for Designing Agent Systems*.

[Deloach, 2005], Deloach S.A., 2005. *Engineering Organizationbased Multiagent Systems*. The 4th International Workshop on Software Engineering for Largescale multiagent Systems (SELMAS'05), vol. 3914, pp. 109–125.

[Demazeau et al., 1990] Y. Demazeau et J.P. Muller, Decentralized artificial intelligence, proceeding of the 1st modeling autonomous agents and multi-agents worlds, Cambridge : North-Holland, 1990.

[Demazeau et Müller, 1991] Demazeau, Y et Müller, J.-P. From Reactive to Intentional Agents. Dans Decentralized A.I. 2, Y.Demazeau & J.P. Müller Eds, Elsevier, 1991.

[Demazeau, 2001], Y. Demazeau et J-P. Briot. Principes et architecture des systèmes multiagents. Hermes, Lavoisier, Paris, France, 2001.

[Denis et Gilleron, 2013], François Denis Rémi Gilleron , Apprentissage à partir d'exemples, Notes de cours <http://www.grappa.univ-lille3.fr/polys/apprentissage/>, consulté le 11/07/2013.

[Durfee, 1989] E.H. Durfee, V. R. Lesser et D. D. Corkill. Trends in Cooperative Distributed Problem Solving. volume 1, pages 63–83, 1989.

[Drogoul, 2003], Drogoul Alexis, Intelligence artificielle distribuée, principes systèmes et applications, MIRIAD/OASIS/LIP6; Université Paris6, 2003.

[Davenport, 2000] Davenport T.H. et Prusak L., Working knowledge : How organizations manage what they know. Harvard Business School Press, Boston, 2000.

[Duvallet, 2001], Claude Duvallet, Des systèmes d'aide à la décision temps réel et distribués : modélisation par agents, Thèse de doctorat, Université du Havre, 2001.

[Dokeos, 2014], <http://www.dokeos.com/fr> (consulté le: 20 mai 2014)

[Durand et al., 2008], Durand A., Dossou Anani Koffi Dogbe-Semanou, Leproust M., Anderstichel H. (2008). étude comparative dans le cadre du Projet @2L organisé au sein du master Ingénierie Pédagogique Multimedia dispensé par le CUEEP de Lille avril 2008.

[Diattara, 2013], Awa Diattara, " Problématique de l'acquisition des connaissances pour des EIAH fortement orientés connaissances : vers un outil auteur pour le projet AMBRE ", (LIRIS, Lyon) et (LIG, Grenoble), 2013.

[Derbali, 2013], Lotfi Derbali, Contribution de la motivation dans les jeux sérieux, Université de Montréal, Thèse présentée au Département d'informatique et de recherche opérationnelle en Mars 2013.

E

[Emmanuel et al., 2001], Emmanuel Adam, Rene Mandiau, and Christophe Kolski. Application of a holo-nic multi-agent system for cooperative work to administrative processes. JASS, 2(1) :100–115, 2001.

[Emmanuel et al., 2003], Emmanuel Adam, Christophe Kolski, René Mandiau, and Emmanuel Vergison. A software engineering workbench for modeling groupware activities. In Proceedings of the Tenth International Conference on Human-Computer Interaction, pages 1499–1503. Lawrence Erlbaum Associates, 2003.

[En-Naimi, 2013], E. M. En-Naimi, "Ingénierie des Systèmes d'Information Intelligents et Distribués & Les Systèmes Multi-Agents (ISIID-SMA) : Notions de Base, Principes Généraux et Applications", HDR, Université Abdelmalek Essaâdi, FST de Tanger, Dept. Génie Informatique, Décembre 2013.

[eXe, 2014], exelearning.org/ (consulté le 9 juillet 2014)

F

[Fadi, 2009] Fadi B., Extraction de connaissances d'adaptation en raisonnement à partir de cas, Thèse de doctorat, Laboratoire Lorrain de Recherche en Informatique et ses Applications, université Henri Poincaré Nancy1, 2009.

[Ferraes, 1986], S. G. Ferraes. Bayes theorem and the probabilistic prediction of inter-arrival times for strong earthquakes felt in mexico city. *Journal of Physics of the Earth*, 34 :71–83, 1986.

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

[Ferber et Gutknecht, 1998], J. Ferber et O. Gutknecht: « A meta-model for the analysis and design of organizations in multi-agent systems ». In In Proceedings of the third international conference on multi-agent systems, 1998.

[Ferber et al., 2004], Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations: an organizational view of multi-agent systems. In James Odell P. Giorgini, Jörg Müller, editor, Agent-Oriented Software Engineering (AOSE), volume LNCS 2935, pages 214–230, Melbourne, july 2004. Berlin Heidelberg. Springer.

- [Finin et al., 1995] T. Finin, Y. Labrou, and J. Mayfield. KQML as an agent communication language. *Jeff Bradshaw (Ed.), "Software Agents", MIT Press, Cambridge, 1995.*
- [Finnie et Sun, 2003], Gavin Finnie and Zhaohao Sun." R5 model for case-based reasoning". *Knowledge-Based Systems*, 16:59–65, 2003.
- [FIPA, 1999] FIPA. FIPA Specification. rapport, 1999, FIPA Association, 1999.
- [Florez, 1999], M. Florez: « Towards a Standardization of Multi-Agent System Frameworks », *ACM Crossroads Student Magazine*. Canada, 1999.
- [Frasson et al., 1998] Frasson C, Martin L, Aïmeur E, and Gouardères G. Lanca: a distance learning architecture based on networked cognitive agents. *International Conference on ITS*, San Antonio, 1998.
- [Fuchs et al., 1995], Fuchs B., Mille A. et Chiron B., Operator decision aiding by adaptation of supervision strategies. *Case-Based Reasoning Research and Development*, proceedings of the 1st International Conference on Case-Based Reasoning (ICCBR'95), LNAI, 1010, 23–32, 1995.
- [Fuchs, 1997] Fuchs B., Représentation des connaissances pour le raisonnement à partir de cas : Le système ROCADE. Thèse de doctorat, Laboratoire Image Signal Acoustique de CPE Lyon, 1997.
- [Fuchs et al., 1999] Fuchs B., Lieber J., Mille A., et Napoli A., Towards a unified theory of adaptation in case-based reasoning. In *Case Based Reasoning Research and Development*, The 3rd International Conference on Case-Based Reasoning (ICCBR'99), volume 1650 de LNAI, pages 104–117, Seon Monastery, Germany, August 25-27 1999. Springer Verlag.
- [Fuchs et al., 2006] Fuchs B., Lieber J., Mille A. et Napoli A., Une première formalisation de la phase d'élaboration du raisonnement à partir de cas. *Actes du 14ième atelier du raisonnement à partir de cas*, Besançon, mars, 2006.
- [Felder et Silverman, 1988], Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education*, 78(7), 674-681.

G

- [Georgé et al., 2005], J-P. Georgé, M-P. Gleizes, and P. Glize. Basic approach to emergent programming - feasibility study for engineering adaptive systems using self-organizing instruction-agents. In *LNCS 3910 Springer Verlag*, editor, *The Third International Workshop on Engineering Self-Organising Applications (ESOA'05) at the Fourth International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS05)*, pages 16–30, Utrecht, Netherlands, July 2005.
- [Goglin, 2001], Goglin, J.F. : « Construction du datawarehouse », Hermès, 2001.
- [Graff et Giardina, 2005], C.J. Graff and C. Giardina. Dynamic petri-nets : a new modeling technique for sensor networks and distributed concurrent systems. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, volume 1, pages 506–512, 2005.
- [Gutknecht et Ferber, 2000], Olivier Gutknecht and Jacques Ferber. Madkit : a generic multi-agent platform. In *Proceedings of the fourth international conference on Autonomous agents*, AGENTS '00, pages 78–79, New York, NY, USA, 2000. ACM.
- [GARLATTI, 1996], GARLATTI Serge, Les systèmes interactifs d'aide à la décision en situations complexes. 40èmes Journées de l'APHO (Association de Pharmacie Hospitalière de l'Ouest) : Pharmacien hospitalier et troisième millénaire, Avril, Perros-Guirec, 1996.

H

- [Hammond, 1986] Hammond K.J., CHEF: a model of case-based planning. Proc. Of AAAI'86, Morgan Kaufman, pp. 267-271, 1986.
- [Hammond, 1989] Hammond K. J., Case-Based Planning: Viewing Planning as a Memory Task. Academic Press, San Diego, 1989.
- [Haouchine et al., 2006] Haouchine K. M., Chebel-Morello B. et Zerhouni N., « Méthode de Suppression de Cas pour une Maintenance de Base de Cas », Laboratoire d'Automatique de Besançon, Besançon, 14e Atelier de Raisonnement à Partir de cas mars 2006 – Besançon.
- [Haouchine, 2009] Haouchine K. M., Remémoration guidée par l'adaptation et maintenance des systèmes de diagnostic industriel par l'approche du raisonnement à partir de cas, Thèse de doctorat, Laboratoire d'automatique de Besançon, université de Franche-Comté, 2009.
- [Hattenschwiler, 1998], Haettenschwiler, P., M. Moresino & A. Schroff : "Rapid prototyping of decision support systems", ICSC Symposium, Tenerife, 1998.
- [Holtzman, 1989] Holtzman, S. (1989): Intelligent decision systems, Addison Wesley.
- [Holland, 1975], J. H. Holland, *Adaptation In Natural And Artificial Systems*, University of Michigan Press, 1975.
- [Hsairi, 2010] L. Hsairi : « Résolution des conflits par négociation à base d'argumentation dans le cadre de l'entreprise étendue ». Thèse de doctorat, 2010.
- [Hübner et al., 2002], Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. Spécification structurelle, fonctionnelle et déontique d'organisations dans les systèmes multi-agents. In P. Mathieu and J.P. Müller, editors, Actes des 10èmes Journées Francophones Intelligence Artificielle Distribuée & Systèmes Multi-Agents (JFIADSMA'02), pages 205–216, 2002.
- [Haynes et al., 1998], T. Haynes, K. Lau, and S. Sen, Learning cases to compliment rules for conflict resolution in multi-agents systems. In Sandip Sen, editor, AAAI Symposium on Adaptation, Co-evolution and Learning in Multi-agents Systems, pages 51–56, 1998.

I

- [Inmon, 1996], Inmon W.H., Building the Data Warehouse, New York: John Wiley & Sons, 1996.
- [Ivana, 2006] Ivana R., Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas : Application au diagnostic dans une plateforme d'e-maintenance, Laboratoire d'Automatique de Besançon, université de Franche-Comté, 2009.
- [IMS-LD, 2014], http://en.wikipedia.org/wiki/IMS_Learning_Design (consulté le: 20 mai 2014)
- [MS, 2014], <http://fr.wikipedia.org/wiki/IMS> (consulté le: 20 mai 2014)

J

- [JACK, 2004], Agent Oriented Software Group. *JACK Intelligent Agents JACK Manual Release 4.1*, Mai 2004.
- [Jaczynski, 1998], Jaczynski M., Modèle et plate-forme à objets pour l'indexation par situations comportementales : application à l'assistance à la navigation sur le Web. Thèse de doctorat, Université de Nice-Sophia Antipolis, 1998.

[Jaczynski et Trousse, 1998], Jaczynski M. et Trousse B., assisted browsing by reusing past navigations of a group of users. Proceedings of EWCBR'98, LNAI, 1488, 160-171, 1998.

[Jaere et al., 2002], Jaere M. D., Aamodt A. et Skalle P., Representing temporal knowledge for case-based prediction, ECCBR'02, 2002.

[Jennings et al., 1998] N. R. Jennings, K. Sycara, and M. Wooldridge. A Roadmap of Agent Research and Development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1998.

[Juan et al., 2003], T. Juan, L. Sterling, M. Martelli et V. Mascardi: « Customizing aose methodologies by reusing aose features ». Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS-2003), pp.113–120, 2003.

[James, 2005], James O'Brien « les systèmes d'informations de gestion », Editions du Renouveau pédagogique Inc., 1995, Montréal, Canada.

[JADE, 2013], JADE : <http://jade.tilab.com/> (consulté le: 24/09/2013) (consulté le: 24/09/2013)

[JACK, 2013], JACK: <http://aosgrp.com/products/jack/index.html#.UXVGMcqaV5o> (consulté le: 24/09/2013).

K

[Kadous et Sammut, 2002] Mohammed Waleed Kadous, and Claude Sammut. Temporal classification : Extending the classification paradigm to multivariate time series. Technical report, 2002.

[Kanawati et al., 1999], Kanawati R., Jaczynski M., Trousse B. et Andreoli J.M., Applying the Broadway recommendation computation approach for implementing a query refinement service in the meta search engine. Actes de RàPC'99 Raisonement à Partir de Cas, Plaiseau, p, 17–26. 1999.

[Kahn, 1993] Kahn C.J., Case-based selection of diagnostic imaging procedure, In: Leake DB(ed). Case-Based Reasoning: Papers from the 1993 Workshop. Menlo Park : AAAI Press, 1993.

[Karoui et al., 2006] Karoui H., Kanawati R. et Petrucci L., COBRAS: Cooperative CBR system for bibliographical reference recommendation. In the 8th European Conference on Case-Based Reasoning (ECCBR'06), éditeurs Thomas Roth-Berghofer, Mehmet H. Göker, et H. Altay Güvenir, volume 4106 de LNCS, pages 76–90, Ölüdeniz/Fethiye, Turkey, September 2006, Springer.

[Keen et Scott-Morton, 1978] Keen, P., & M. Scott-Morton (1978): "Decision Support Systems: an organizational perspective", Addison-Wesley Publishing.

[Kersten, 1999] Kersten, G. & S. Noronha (1999): "WWW-based negotiation support: design, implementation, and use", decision Support Systems, Newport Beach, Vol. 25, N°2, pp. 135-154.

[Kolodner, 1988] Kolodner, J., Workshop on case-based Reasoning, editor (1988) DARPA 88, Clearwater, Florida. Morgan Kaufmann, San Mateo.

[Kolodner, 1993] Kolodner J., Case-Based Reasoning. Morgan Kaufmann, San Mateo, UCA, 1993.

[Kolodner, 1996] Kolodner J., Making the implicit explicit: Clarifying the principles of case-based reasoning. In Case-Based Reasoning: Experiences, Lessons and Future.

[Komi, 2008], Komi S., «Conception et implémentation d'une plate-forme d'évaluation adaptative des apprentissages », Université du Québec À Montréal, 2008.

[Koton, 1988] Koton P., Using experience in learning and problem solving, PhD Thesis, Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge Massachusetts, USA 1988.

[Kreczanik, 2008], Kreczanik T. . "Conception et appropriation des dispositifs d'information pédagogiques hypertextuels". Université Jean-Moulin Lyon, thèse soutenue le 4 novembre 2008.

L

[Lawrence et Rabiner, 1990], Lawrence R. Rabiner. Readings in speech recognition. chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[Lazkano et al., 2007], E. Lazkano, B. Sierra, A. Astigarraga, and J.M. Martínez-Otzeta. On the use of bayesian networks to develop behaviours for mobile robots. *Robotics and Autonomous Systems*, 55(3):253 – 265, 2007.

[Labouidya et al., 2008] O. Labouidya, N. Elkamoun, H. Benabdillah and A. Dahbi, “Berraissoul A., Conception d'un module positionnement de l'apprenant pour les systèmes tuteurs intelligents : modèle et expérience”, ISDM n° 34, page 34, 2008.

[Labidi 1993], S. Labidi et W. Lejouad. De l'intelligence artificielle distribuée aux systèmes multiagents, 1993. Rapport de recherche INRIA SOPHIA-ANTIPOLIS 1993.

[Lebrun, 2012], Yoann, LEBRUN, "Architecture multi-agent pour la gestion d'objets tangibles et virtuels sur Tables Interactives", Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH), l'Université de VALENCIENNES ET DU HAINAUT-CAMBRESIS, 2012.

[Lesire et Tessier, 2005], Charles Lesire and Catherine Tessier. Particle petri nets for aircraft procedure monitoring under uncertainty. In *Applications and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 857–870. 2005.

[Leake et Wilson, 1998] Leake D.B. et Wilson D.C., Categorizing case-base maintenance: dimensions and directions. *Advances in Case-Based Reasoning: 4th European Workshop, EWCBR 98. Proceedings*, 1998: pp.196-207, 1998. Springer- Verlag, Berlin, Germany.

[Leake et Wilson, 2000] Leake D.B. et Wilson D.C., Remembering why to remember: Performance-guided case-base maintenance. *Advances in Case-Based Reasoning, Proc. Of EWCBR-2K*, Springer-Verlag, 2000.

[Leake et Sooriamurthi, 2001], D. Leake and R. Sooriamurthi. When two case bases are better than one: Exploiting multiple case bases. In *ICCBR*, pages 321–335, 2001.

[Leake et Sooriamurthi, 2002], D. Leake and R. Sooriamurthi. “Managing Multiple Case Bases: Dimensions and Issues”, *FLAIRS-02 Proceedings*. AAAI (www.aaai.org).pp 106-110, 2002.

[Leake et Sooriamurthi, 2004], D. Leake and R. Sooriamurthi, “ Case dispatching versus case-base merging: when MCBR matters”. *International Journal on Artificial Intelligence Tools* 13(1): 237-254 (2004)

[Lévine et Pomerol, 1989], Lévine P. & J. Pomerol : « Systèmes interactifs d'aide à la décision et systèmes experts », Editions Hermès, 1989.

[Lieber et al., 1999], Lieber J., Aquin M., Brachais S., and Napoli A., Une étude comparative de quelques travaux sur l'acquisition de connaissances d'adaptation pour le raisonnement à partir de cas. 12ième atelier de raisonnement à partir de cas, Villetaneuse, Paris, 1999.

[Lieber et al., 2004] Lieber J., Aquin M., Brachais S. et Napoli A., Une étude comparative de quelques travaux sur l'acquisition des connaissances d'adaptation pour le raisonnement à partir de cas. 12ème Atelier de Raisonnement à Partir de Cas (RàPC'04), pages 53–60, Villetaneuse, France, Mars 2004.

[Lieber, 2007] Lieber J., Application of the Revision Theory to Adaptation in Case- Based Reasoning: the Conservative Adaptation. In *7th International Conference on Case- Based Reasoning - ICCBR'07* 4626 (2007) 239-253.

[Likhachev et Arkin, 2001], Likhachev M., and Arkin A.C., Spatio-Temporal Case-Based Reasoning for Behavioral Selection. The IEEE International Conference on Robotics and Automation (ICRA2001), 2:1627-1634, 2001.

[Likhachev et al., 2002], Likhachev M., Kaess M., and Arkin R.C., Learning behavioral parameterization using spatiotemporal case-based reasoning. The IEEE international conference on robotics and automation (ICRA02), Washington, D.C, 2002.

[Longueville, 2003] Longueville, B. (2003) : « Capitalisation des processus de décision dans les projets d'innovation : Application à l'automobile », Thèse de doctorat, Ecole Centrale de Paris, Décembre 2003.

[Leshin et al., 1992], Leshin, C., Pollock, J., & Reigeluth, C. (1992). Instructional design strategies and tactics. Englewood Cliffs, NJ: Educational Technology Publications.

[Le Corre, 2013], Le Corre, Frédéric. CHRYSAOR: un système tutoriel intelligent pour les environnements virtuels d'apprentissage humain. Application à la formation au matériel de laboratoire en hémotase. 2013. Thèse de doctorat. Université de Bretagne occidentale-Brest.

M

[Main et al., 2000] Main J., Dillon T.S. et Shiu S.C.K., A tutorial on case-based reasoning. Soft Computing in Case Based Reasoning, pages : 1-28, 2000.

[Malek , 1996], Malek M., Un modèle hybride de mémoire pour le raisonnement à partir de cas. Thèse de doctorat, Université de Grenoble 1, 1996.

[Malek, 2000] Malek M., Hybrid approaches for integrating neural networks and casebased reasoning : From loosely coupled to tightly coupled models. In Soft Computing in Case Based Reasoning, éditeurs Tharam S. Dillon Sankar K. Pal et Daniel S. Yeung, pages 73-94, Mars 2000.

[Malek et Kanawati, 2001] Malek M., et Kanawati R., Cobra : A cbr-based approach for predicting users actions in web site, International Conference on Case-based Resoning ICCBR'01., p. 336-346, 2001.

[Malek et Kanawati, 2004] Malek M. et Kanawati R., A data driven CBR approach: A continuous maintenance strategy. In the International Conference on Advances in Intelligent Systems: Theory and Applications, pages 281–290, Luxemburg, November 2004.

[Marling et al., 2003], C. Marling, M. Tomko, M. Gillen, D. Alexander, and D. Chelberg, "Case-Based Reasoning for Planning and World Modeling in the RoboCup Small Size League", School of Electrical Engineering and Computer Science, Ohio University, Athens, Ohio 45701, USA.

[Marakas, 2003] Marakas, G. (2003): "Decision Support Systems In the 21st Century", Second Edition, Prentice Hall.

[Matthew et al., 2009], Matthew D. Coles, Djamel Azzi, Barry P. Haynes, and Alan Hewitt. A Bayesian network approach to a biologically inspired motion strategy for mobile wireless sensor networks. Ad Hoc Networks, 7(6) :1217 – 1228, 2009.

[Marcenac et Calderoni, 1997] P. Marcenac and S. Calderoni. Self-organisation in agent-based simulation. In *Proceedings of MAAMW' 97*, Ronnedby, Sweden, 1997.

[Mbala, 2003], Aloys MBALA HIKOLO, Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance, Université de Franche-Comté, 2003.

- [McGinty et Smyth, 2001], L. McGinty and B. Smyth, Collaborative case-based reasoning: Applications in personalized route planning. In ICCBR, pages 362–376, 2001.
- [Mendelsohn et Dillenbourg, 1991] P. Mendelsohn, P. Dillenbourg, “Le développement de l’enseignement intelligemment assisté par ordinateur”, *Symposium Intelligence Naturelle et Intelligence Artificielle*. 1991.
- [Melfort, 2010], Melfort Cindy, « SIMSERV » un système expert d’aide à la sélection de plantes de service pour différents contextes agro-écologiques et socio-économiques, INRA (Institut National de la recherche Agronomique), UR1321, ASTRO Agrosystèmes tropicaux, F-97170, Petit-Bourg, France, 2010.
- [Mesa, 1997], MESA International: « MES Functionalities & MRP to MES Data Flow Possibilities » White Paper 2 (Pittsburgh: Manufacturing Execution Systems Assoc., 1997).
- [Mille, 1995] Mille A., Raisonnement basé sur l’expérience pour coopérer à la prise de décision, un nouveau paradigme en supervision industrielle. Thèse de doctorat, Université de Saint Etienne, 1995.
- [Mille et al., 1996] Mille A., Fuchs B. et Herbeaux O., A unifying Framework for Adaptation in Case-Based Reasoning. Workshop on Adaptation in Case-Based Reasoning, ECAI-96, pp. 22-28, Budapest, Hungary, August, 1996.
- [Mille, 1999] Mill. A., Tutorial CBR : Etat de l’art de raisonnement à partir de cas. Plateforme AFIA’99, Palaiseau, 1999.
- [Mille, 2006] Mille A., Traces based reasoning (TBR) definition, illustration and echoes with story telling. Rapport Technique RR-LIRIS-2006-002, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon, January 2006.
- [Müller, 2002], J.P. Müller. Des systèmes autonomes aux systèmes multiagents : Interaction, émergence et systèmes complexes. Thèse, Université Montpellier II, 2002.
- [Minsky, 1974], Marvin Minsky, A Framework for Representing Knowledge, MIT-AI Laboratory Memo 306, June, 1974. Reprinted in Cognitive Science, Collins, Allan and Edward E. Smith (eds.) Morgan-Kaufmann, 1992 ISBN 55860-013-2.
- [Minsky, 1988], Marvin Minsky,
http://theses.univ-lyon2.fr/documents/getpart.php?id=lyon2.2004.bailly_b&part=87691.
- [Moodle, 2014], <https://moodle.org/> (consulté le : 20 mai 2014)
- [Moodle-stat, 2014], <http://en.wikipedia.org/wiki/Moodle> (consulté le: 20 mai 2014)
- [MADKIT, 2013], MADKIT : <http://www.madkit.net/> (consulté le: 24/09/2013)
- [Culloch et Pitts , 1943], McCulloch, W. S., Pitts, W., A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.
- [Mazyad, 2013], Hanaa Mazyad, "Une Approche Multi-agents à Architecture P2P pour l’Apprentissage Collaboratif", Université du Littoral Cote d’Opale, Thèse soutenue le 31 janvier 2013.

N

- [Navarro et al., 2010], Navarro M., Heras S., Julian V., Botti V., Incorporating temporal-bounded CBR techniques in real-time agents, Expert Systems with Applications 38, pp. 2783-2796, ScienceDirect, 2010.

[NFAOUI, 2008], El Habib NFAOUI, "Architecture d'aide a la décision distribuée et de simulation proactive dans les chaînes logistiques : une approche multi agent", Université Lumière Lyon 2, Université Sidi Mohamed Ben Abdallah de Fès, 2008.

[Nwana, 1996], A. Nwana: « Software Agents: An Overview », knowledge Engineering Review, Vol. 11, N° 3, pp. 1-40, 1996.

[Needleman et Wunsch, 1970], NEEDLEMAN, Saul B. et WUNSCH, Christian D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 1970, vol. 48, no 3, p. 443-453.

O

[Ontañón et Plaza, 2007] Ontañón et E. Plaza, Learning and Joint Deliberation through Argumentation in Multi-Agents Systems, in AAMAS'07 Proceedings of the 6th international joint conference on Autonomous agents and multi-agents systems, Honolulu, HI, USA, 2007.

[Ontañón et Plaza, 2011], S. Ontañón et E. Plaza, "An argumentation framework for learning, information exchange, and joint-deliberation in multi-agent systems." *Multiagent and Grid Systems*, vol. 7, no 2, p. 95-108, 2011.

[Odell et al., 2001], Odell, J., Van Dyke Parunak, H. et Bauer, B. Representing agent interaction protocols in UML. En P. Ciancarini and M. Wooldridge (eds.), *Agent-Oriented Software Engineering - Proceedings of the First International Workshop (AOSE-2000)*. Springer- Verlag, 2001.

[Olivier, 2006], François Olivier, "De l'identification de structure de réseaux bayésiens à la reconnaissance de formes à partir d'informations complètes ou incomplètes", Thèse de Doctorat de l'Institut National des Sciences Appliquées de Rouen, Laboratoire LITIS, 2006.

P

[Paquette, 2002], Paquette, G., L'ingénierie pédagogique, pour construire l'apprentissage en réseau. Québec: Presses de l'Université du Québec, 2002.

[Pavon et Gomez, 2003], J. Pavon et J. Gomez: « Agent-oriented Software Engineering with INGENIAS ». In 3rd Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'03), Prague, Czech Republic., 2003.

[Plaza et al., 1997], E. Plaza, J. Llu, and F. Martin, Cooperative casebased reasoning. In Gerhard Weiss, editor, *Distributed Artificial Intelligence Meets Machine Learning. Learning in Multi-Agents Environments*, number 1221 in *Lecture Notes in Artificial Intelligence*, pages 180–201.

[Piotr et al., 2003], Piotr Chrzastowski-Wachtel, Boualem Benatallah, Rachid Hamadi, Milton O'Dell, and Adi Susanto. A top-down petri net-based approach for dynamic workflow modeling. In Mathias Weske, editor, *Business Process Management*, volume 2678 of *Lecture Notes in Computer Science*, pages 1019–1019. Springer Berlin / Heidelberg, 2003.

[Pomerol, 1997] Pomerol J.C. (1997). "Artificial Intelligence and Human Decision Making", *European Journal of Operational Research*, Vol. 99, pp. 3-25.

[Power, 2002], Power, D.J.: "Decision support systems: concepts and resources for managers", Westport, Conn., Quorum Books, 2002.

[Pokahr et al., 2005], Pokahr, L. Braubach et W. Lamersdorf: « Jadex: A bdi reasoning engine, chapter of multi-agent programming ». KluwerBook, Editors: R. Bordini, M. Dastani, J. Dix and A. Seghrouchni, 2005.

[Prasad et al., 1995], M. V. Nagendra Prasad, Victor R. Lesser and Susan E. Lander “Retrieval and Reasoning in Distributed Case Bases”, Appeared in the Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries, Vol 7, No. 1, March 1996, pp 74 - 87. Also as UMASS CS Technical Report 95-27, 1995.

[Pétri, 2014], http://www.scholarpedia.org/article/Petri_net (consulté le 27 janvier 2014).

[Perrin, 2004], Perrin, A. (2004). La valorisation des pratiques de management des connaissances, outils de mesure et mesure des outils. FRE 2780 CNRS-UNSA, France : Université de Nice Sophia Antipolis.

[Prax, 2007], Jean-Yves Prax, Le Manuel du Knowledge Management, mettre en réseau les hommes et les savoirs pour créer de la valeur, Dunod 2007.

[Plume, 2013], <https://www.projet-plume.org/>, article disponible sur : <https://www.projet-plume.org/fr/ressource/comparaison-de-lms> (dernière consultation le 28 juillet 2014).

[Plate-stat, 2014],
http://fr.wikipedia.org/wiki/Liste_de_plateformes_p%C3%A9dagogiques_sous_licence_libre
(consulté le 10 juillet 2014)

Q

[Quillian, 1968] Quillian M.R., Semantic memory. In Minsky M. (éd.), Actes Semantic information processing. Cambridge, Mass : MIT press. 227-270.

R

[Ram et Santamaria, 1993], Ram A., and Santamaria J.C., Continuous case based reasoning. The AAAI case based reasoning workshop, pages 86–93, Washington, 1993.

[Ram et Santamaria, 1997], Ram A., and Santamaria J.C., Continuous case-based reasoning, Artificial intelligence, 90 : 25-27, 1997.

[Richter et Wess, 1991] Richter M.M. et Wess S., Similarity, Uncertainty and Case- Based Reasoning in PATDEX. Automated reasoning, essays in honour of Woody Bledsoe, Kluwer, 1991, pp. 249-265.

[Richter et al., 1993] Richter M.M., Wess S., Althoff K.D. et Maurer F., First European Workshop on Case-Based Reasoning, University of Kaiserslautern, Germany, Lecture Notes in Artificial Intelligence, vol 837, Springer Verlag, Berlin, 1993.

[Richter, 1998] Richter M.M., Introduction. In Case-Based Reasoning Technology: From Foundations to Applications, Edited by M. Lena, B. Bartsc-Sporl, H. D. Burkhard, et S. Wess. Springer-Verlag, Berlin, pp. 1-15, 1998.

[Rissland et al., 1995] Rissland, E. L., Daniels, J. J. Using CBR to Drive IR, [en ligne]. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada. SanFrancisco: Morgan Kaufmann, 1995, p. 400-407.

[Rougegrez-Loriette, 1994], Rougegrez-Loriette S., Prédiction de processus à partir de comportements observés : le Système REBECAS, Thèse de doctorat, Université Paris 6, 1994.

[Rougegrez-Loriette, 1998], Rougegrez-Loriette S., Raisonnement à partir de cas pour des évolutions spatio-temporelles de processus, revue internationale de géomatique, journées Cassini, vol. 8, n 1-2, pages 207-227, 26-27, 1998.

[Sophie Lorette, 2006], Sophie Lorette. Théorie de l'évidence et logique floue pour la détermination du stade de réalisation d'un scénario. In 12ième rencontre francophone sur la logique floue et ses applications (LFA2006), 2006.

[Roy, 1993] Roy, B. & D. Bouyssou (1993) : "Aide multicritère à la décision : méthodes et cas". Economica, Paris (France).

[Richter et Wess, 1991] Richter M.M. et Wess S., Similarity, Uncertainty and Case- Based Reasoning in PATDEX. Automated reasoning, essays in honour of Woody Bledsoe, Kluwer, 1991, pp. 249-265.

S

[Sabouret, 2009], Nicolas Sabouret, HDR, "Interactions sur le fonctionnement dans les systèmes multi-agents ouverts et hétérogènes", université Pierre et Marie Curie, 2009.

[Saleh et El-Morr, 2004] Saleh, K., El-Morr, C. *M-UML: an extension to UML for the modeling of mobile agent-based software systems*. Information and Software Technology, vol. 46, pp: 219–227, 2004.

[Saxena, 2007], Saxena A., Knowledge-Based Architecture for Integrated Condition Based Maintenance of Engineering Systems, PhD in Electrical and Computer Engineering from Georgia Institute of Technology, 2007.

[Schank, 1982] Schank R., Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Cambridge University Press, New York, NY, 1982.

[Schmidt et al., 1998], Schmidt R., Pollwein B., Gierl L. , Medical multiparametric time course prognoses applied to kidney function assessments, forthcoming in International journal on medical informatics, 1998.

[Schneider, 1994] Schneider, D. K. (1994). Modélisation de la démarche du décideur politique dans la perspective de l'intelligence artificielle. Thèse de l'Université de Genève, Suisse.

[Scott-Morton, 1971] Scott Morton M. (1971): "Management decision systems, computer based support for decision making", Harvard University, Boston (MA, USA).

[Sean, 1996], Sean R Eddy. Hidden markov models. Current Opinion in Structural Biology, 6(3):361–365, 1996.

[Simon, 1977] Simon H. (1977): "The New science of management decision", Prentice hall, Englewood-Cliffs.

[Smyth et Keane, 1995] Smyth B., Keane M. T., « Experiments on Adaptation-Guided Retrieval in a Case-Based Design System », Proceedings of ICCBR-95, LNAI 1010, Springer-Verlag, p. 313-324, 1995.

[Smyth et McKenna, 1998], Smyth B. et McKenna E., Modelling the Competence of Case-Bases. Advances in case-based reasoning. Lecture-notes-in-computer-science, 1488 : 208-220, Dublin, 23-25, 1998.

[Smyth et McKenna, 1999] Smyth B. et McKenna E., Building compact competent case-bases. Proc. of the 3rd International Conference on Case-Based Reasoning, pp. 329-342, 1999.

[Soh et Luo, 2003], L. k. Soh and J. Luo, Combining individual and cooperative learning for multiagent negotiations. In Int. Conf. Autonomous Agents and Multiagent Systems AAMAS'03, pages 686–692, 2003.

[SOHAIL et al. , 2005], Muhammad SOHAIL, Muhammad Afan SHAH, Said SAIDNA, Abbas SLIMANI, Plates-formes des systèmes multi-agents, Master recherche en Informatique, Université Paris Sud – Orsay, 2005.

[Sprague et Carlson, 1982], Sprague, R. & E. Carlson (1982): “Building Effective Decision Support Systems”, Prentice-Hall, Inc, Englewood Cliffs.

[Strugeon, 1995] E. Le Strugeon. Une méthodologie d’auto-adaptation d’un système multi-agents cognitifs. *Thèse de doctorat, Université de Valenciennes*, 1995.

[Sudeikat et al., 2004], J. Sudeikat, L. Braubach, A. Pokahr et W. Lamersdorf: « Evaluation of agentoriented software methodologies – examination of the gap between modeling and platform». In AOSE 2004, volume 3382 of LNCS, pages 126–141. Springer, 2004.

[SCORM, 2014], http://fr.wikipedia.org/wiki/Sharable_Content_Object_Reference_Model (consulté le: 20 mai 2014)

[Sakai, 2014], <http://www.sakaiproject.org/> (consulté le: 20 mai 2014)

[SWARM, 2013], SWARM : <http://savannah.nongnu.org/projects/swarm> (consulté le: 24/09/2013)

T

[Tarek, 2006], Tarek Jarraya, Réutilisation des protocoles d’interaction et Démarche orientée modèles pour le développement multi-agents, l’Université de Reims Champagne Ardenne, 2006.

[Taggart, 1981] Taggart W. et Robey D. Minds and Managers : On the Dual Nature Of Human Information Processing And Management. *Academy of Management Review*, 6(2) :187-195, 1981.

[Thomas et Keiji, 1989], Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Comput. Intell.*, 5 :142–150, December 1989.

[Tolchinsky et al., 2006], P. Tolchinsky, S. Modgil, U. Cortes, and M. Sanchez-marre, Cbr and argument schemes for collaborative decision making. In *Conference on computational models of argument*, COMMA-06 (Vol. 144, pp. 71–82), 2006.

[Tolchinsky et al., 2012], Tolchinsky, P., Modgil, S., Atkinson, K., McBurney, P., & Cortés, U., “Deliberation dialogues for reasoning about safety critical actions”. *Autonomous Agents and Multi-Agent Systems*, 2012, vol. 25, no 2, p. 209-259.

[Trentesaux, 1996] Trentesaux, D. (1996) : « Conception d’un système de pilotage distribué, supervisé et multicritère pour les systèmes automatisés de production », Thèse de doctorat, Institut National Polytechnique de Grenoble (France).

[Turban, 1995] Turban, E. (1995): “Decision Support and Expert Systems”, Macmillan, New York (USA).

[Turban, 1998], Turban, E. & J. Aronson : “Decision Support and Intelligent systems”, Prentice-Hall International, Upper Saddle River (N. J., USA), 1998.

[Turing , 1950], Turing, A.M. (1950). Computing machinery and intelligence. *Mind*, 59, 433-460.

[Turing, 2002], Cours en ligne, Politechnica University of Bucharest http://turing.cs.pub.ro/auf2/html/chapters/chapter2/chapter_2_2_2.html, (consulté le 05 mai 2014).

V

[Valdes, 2004], Valdes J., *Les systèmes experts en détail*, <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/jvaldes/index.html> [en ligne] consulté le 11/07/2013.

[Vance, 1997] Vance D.M., Information, Knowledge and Wisdom : The Epistemic Hierarchy and Computer-Based Information System. In *Proceedings of the Third Americas Conference on Information Systems*, B. Perkins and I. Vessey (eds.), 1997.

[Varma, 1999] Varma A., ICARUS: Design and Deployment of a Case-Based Reasoning System for Locomotive Diagnostics. In *3rd international conference on case-based reasoning (ICCBR-99)*, vol. 1650, pp. 581-595.

[Vazquez-Salceda et al., 2004] J. Vazquez-Salceda, V. Dignum, , and F. Dignum. Organizing multiagent systems. Technical Report UU-CS-2004-015, Institute of Information and Computing Sciences, Utrecht University, 2004.

[Veloso et al., 1995] Veloso M., Carbonell J., Pérez A., Borrajo, D., Fink, E., et Blythe, J., Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 7(1):81–120, 1995.

[Vlachos et al., 2002], Vlachos M., Kollios K., and Gunopulos G., Discovery similar multidimensional trajectories. *The 18th International Conference on Data Engineering (ICDE02)*, pages 673–684, San Jose, CA, 2002.

[VOYAGER, 2013], VOYAGER : <http://www.recursionsw.com/products/voyager/voyager-intro.html> (consulté le: 24/09/2013).

W

[Watson et Marir, 1994] Watson I. et Marir F., Case-Based Reasoning: A Review. *The Knowledge Engineering Review*, 1994.

[Wagner et Tulba, 2003], G. Wagner and F. Tulba. Agent-Oriented Modeling and Agent-Based Simulation. In *Proceedings of 5th Int. Workshop on Agent-Oriented Information Systems (AOIS-2003)*, ER2003 Workshops, Springer-Verlag, LNCS. In P. Giorgini and B. Henderson-Sellers (Eds.), 2003.

[Wenger, 1987], Wenger, E., *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*: Morgan Kaufman Press, 1987.

[Wikipédia, 2009] Disponible sur: <http://fr.wikipedia.org>.

[Wilke et al., 1997] Wilke W., Vollrath I., Altho K-D., et Bergmann R., Aframework for learning adaptation knowledge based on knowledge light approaches. Dans *Proceedings of the Fifth German Workshop on Case-Based Reasoning*, p. 235–242. 1997.

[Wilson, 2001] Wilson D.C., Case-base maintenance: the husbandandry of experience. Thèse de doctorat, Department of Computer Science Indiana University, Juillet 2001.

[Wilke et Bergmann, 1998] Wilke W. et Bergmann R., Techniques and Knowledge Used for Adaptation During Case-Based Problem Solving. *Proc. of IEA-98-AIE, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1998.

[Wooldridge et Jennings, 1995], M. Wooldridge et N. R. Jennings: “Intelligent agents: Theory and practice “. *Knowledge Engineering Review*, 10:115–152, 1995.

[Wooldridge et al., 2000], M. Wooldridge, N. Jennings et D. Kinny " The GAIA methodology for agent-oriented analysis and design ". Autonomous Agents and Multi-Agent Systems, 3(3), 2000.

[Webber et Pesty, 2002], Webber, C., Pesty, S., Emergence de diagnostic par formation de coalitions - Application au diagnostic des conceptions d'un apprenant. Journées Francophones pour l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents J.P.Muller, 2002, Hermes, Lille, pp.45-57.

Y

[Yang et Zhu, 2001] Yang Q. et Zhu J. A case addition policy for case-based maintenance. Computational Intelligence Journal, A special Issue on Case-Base Maintenance, Blackwell Publishers, Boston MA UK, 17(2), pp. 250-262, 2001.

[Yang et Wu, 2000], Yang Q. et Wu J., Keep it simple : A case-base maintenance policy based on clustering and information theory. 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2000 : advances in artificial intelligence. Proceedings Lecture Notes in Artificial Intelligence, Vol.1822. 2000: 102-114. Montréal PQ, 14-17 May 2000. Springer-Verlag, Berlin, Germany, 2000.

[Yehia et al., 1998], Yehia H.C., Rubin P.E., and Vatikiotis-Bateson E., Quantitative association of vocal-tract and facial behavior. Speech Communication, 26 :23-34, 1998.

[Yu et al., 2003], R. Yu, B. Iunga, H. Panetto, A multi-agents based E-maintenance system with case-based reasoning decision support, Faculté des Sciences, CRAN-Universit de Nancy I, and Faculty of Hydroelectric Power, Huazhong University of Science and Technology, Wuhan, China, in Engineering Applications of Artificial Intelligence 16 (2003) 321–333.

Z

[Zadeh, 1968], Zadeh, L. A., Fuzzy algorithms. Information and control, 12, 94-102, 1968.

[Zadeh, 1973], Zadeh, L. A., Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Systems, Man and Cybernetics, 3, 28-44, 1973.

[Zaraté, 2005], Pascale Zaraté, "Des Systèmes Interactifs d'Aide à la Décision AuxSystèmes Coopératifs d'Aide à la Décision : Contributions conceptuelles et fonctionnelles", Habilitation à Diriger des Recherches, Spécialité Informatique, INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE (INPT), 2005.

[Zehraoui, 2003], Zehraoui F., Cbr system for sequence prediction "casep", In Workshop on Applying case-based reasoning to time series prediction, p. 260–269, Trondheim, Norway, 2003.

[Zehraoui et al., 2003], Zehraoui F., Kanawati R. et Salotti S., Case base maintenance for improving prediction quality. In The International Conference on Case-Based Reasoning (ICCBR-2003), p. 703–717, Trondheim, Norway, 2003.

[Zehraoui, 2004], Zehraoui F., Systèmes d'apprentissage connexionnistes et raisonnement à partir de cas pour la classification et le classement de séquences. Thèse de doctorat, Université Paris13, 2004.

[Zehraoui et Bennani, 2004], Zehraoui F., and Bennani Y., M-SOM-ART : Growing Self Organizing Map for sequence clustering and classification. The European Conference on Artificial Intelligence, Valencia, 2004.

[ZEUS, 2013], ZEUS : <http://www.labs.bt.com/projects/agents/zeus> (consulté le 24/09/2013).

[Zouhair, 2009], Abdelhamid ZOUHAIR, "E-learning : Conception et modélisation du parcours de l'apprenant", Rapport de stage de Master. Master Informatique : Mathématiques et Informatique pour les Systèmes Complexes et Distribués, Université du Havre, 2009.

[Zouhair et al., 2014], Abdelhamid ZOUHAIR, El Mokhtar EN-NAIMI, Benaïssa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, "Our System IDCBR-MAS: from the Modelisation by AUML to the Implementation under JADE Platform", Published in The International Journal of Artificial Intelligence and Interactive Multimedia (IJIMAI), Vol. 2, Regular Issue, No. 6 pp. 48-57, 06/2014.

Publications

I) Publications/Articles dans des Revues & Journaux Internationaux de Spécialité:

1. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, “*Individualized Follow-up of Learners based on Multi-Agents Case-Based Reasoning in Distance Learning*”, **Published in** The International Journal on Information and Communication Technologies, International Sciences Press, India, IJICT, Vol. 4, No. 4, pp. 141-150, November 2011.
2. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, “*Multi-Agents Dynamic Case Based Reasoning and The Inverse Longest Common Sub-Sequence And Individualized Follow-up of Learners in The CEHL*”, **Published in** The International Journal of Computer Science Issues (IJCSI), Volume 9, Issue 4, pp. 25-31, July 2012;
3. El Mokhtar EN-NAIMI, **Abdelhamid ZOUHAIR**, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, “*Intelligent Tutoring Systems Based on the Multi-Agent Systems (ITS-MAS):The Dynamic and Incremental Case Based Reasoning (DICBR) Paradigm and the Inverse Longest Common Sub-Sequence (ILCSS) in the CEHL*”, **Published in** The International Journal of Computer Science Issues (IJCSI), Volume 9, Issue 6, No. 3, pp. 112 - 121, November 2012;
4. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, “*Incremental Dynamic Case Based Reasoning and Multi-Agent Systems (IDCBR-MAS) for Intelligent Tutoring System*”, **Published in** The International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Volume 3, Issue 12, pp. 48-56, December 2013.
5. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, “*Our System IDCBR-MAS: from the Modelisation by AUML to the Implementation under JADE Platform*”, **Published in** The International Journal of Artificial Intelligence and Interactive Multimedia (IJIMAI), Vol. 2, Regular Issue, No. 6 pp. 48-57, 06/2014.
6. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, “*Modelisation and Implementation of our System Incremental Dynamic Case Based Reasoning founded In the MAS under JADE Plate-Form*”, **Submitted in** The International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS), July 2014.

II) Chapitres d'Ouvrage & Communications Publiées en IEEE:

1. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, "*Multi-Agents Case-Based Reasoning and Individualized follow-up of Learner in Remote Learning*", International Conference on Multimedia Computing and Systems (ICMCS), 2011,
Communication Published in The IEEE Xplore, Digital Library,
DOI: 10.1109/ICMCS.2011.5945644, Publication Year: 2011, Page(s): 1-6
(<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5945644&queryText%3D%22En-Naimi%22>);
2. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, "*Dynamic Case-Based Reasoning Based on the Multi-Agent Systems: Individualized*",
Chapter Published in Springer Berlin Heidelberg, Intelligent Distributed Computing VI, Studies in Computational Intelligence, Volume 446, 2013, pp 51-57, Print ISBN : 978-3-642-32523-6; DOI: 10.1007/978-3-642-32524-3_8,
(http://link.springer.com/chapter/10.1007/978-3-642-32524-3_8);
3. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, "*Intelligent Tutoring Systems founded on the Multi-Agents Incremental Dynamic Case Based Reasoning*", Colloquium in Information Science and Technology (CIST), 2012,
Communication Published in The IEEE Xplore, Digital Library,
DOI: 10.1109/CIST.2012.6388066, Publication Year: 2012, Page(s): 74-79
(<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6388066&queryText%3D%22En-Naimi%22>);
4. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, "*Intelligent Tutoring Systems Founded of Incremental Dynamic Case Based Reasoning and Multi-Agents Systems (ITS-IDCBR-MAS)*", International Conference on Advanced Logistics and Transport (ICALT), 2013,
Communication Published in The IEEE Xplore, Digital Library,
DOI: 10.1109/ICAAdLT.2013.6568482, Publication Year: 2013, Page(s): 341-346
(<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6568482&queryText%3D%22En-Naimi%22>);

III) Communications dans des Conférences internationales avec comité de lecture :

1. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, "*E-learning : Adapting pedagogical learning to distance based on the analysis traces of the learner*", The 5th International Conference ECDC2010, Organized by Univ. of Isfahan (Iran) & University of The Havre (France); 30 September - 01October 2010, Kish Island, Iran;

2. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *"Raisonnement à partir de cas et suivi individualisé de l'apprenant en apprentissage à distance"*, 7ème JFMMA, 16-18 Mars 2011, Tanger, Maroc ;
3. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *"MultiAgent Case-Based Reasoning and Individualized Follow-up of Learner in Remote Learning"*, The 2nd International Conference on Multimedia Computing and Systems, ICMCS'11, 7-9 April 2011, Ouarzazate , Morocco;
4. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *"Dynamic Case-Based Reasoning Based on the Multi-Agent Systems: Individualized Follow-up of Learners in Distance Learning"*, 6th International Symposium on Intelligent Distributed Computing - IDC 2012, 24-26 September 2012, Calabria, Italy;
5. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *"Intelligent Tutoring Systems Founded on The Multi-Agent Incremental Dynamic Case Based Reasoning"*, IEEE CIST'12, The 2nd IEEE international Colloquium in Information Science and Technology, 22-24 October 2012, Fez, Morocco;
6. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *"Intelligent Tutoring Systems Founded of Incremental Dynamic Case Based Reasoning and Multi-Agents Systems (ITS-IDCBR-MAS)"*, International Conference on Advanced Logistics and Transport (ICALT'2013), 29-31 May 2013, Sousse, Tunis;
7. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *"Modelisation and Implementation of our System Incremental Dynamic Case Based Reasoning founded In the MAS under JADE Plate-Form"*, International Conference on Multimedia Computing and Systems, ICMCS'14, April 14-16, 2014, Marrakesh, Morocco.
8. **Abdelhamid ZOUHAIR**, El Mokhtar EN-NAIMI, Benaissa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *"The Impact of The Implementation of Our System IDCBR-MAS"*, IEEE International Colloquium in Information Science and Technology (CIST'14), October 2014; Tétouan, Morocco.

Annexe A

Dans cette annexe, nous présentons les cas sources utilisés dans la validation de notre approche.

Cas Source 1

```

-<logs>
-<user>
  <id>4</id>
  <nom>Cas source1</nom>
</user>
-<log>
  <id>513</id>
  <cours>1</cours>
  <date>jeu 22/5/2014 13:45</date>
  <action>login</action>
  <ip>127.0.0.1</ip>
  <info>4</info>
  <date_second>1400759107</date_second>
  <duree_second>19</duree_second>
</log>
-<log>
  <id>519</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:45</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html</info>
  <date_second>1400759126</date_second>
  <duree_second>533</duree_second>
</log>
-<log>
  <id>533</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:54</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/architecture_logiciel_d
    e_la_plateforme_jade.html </info>
  <date_second>1400759659</date_second>
  <duree_second>454</duree_second>
</log>
-<log>
  <id>541</id>
  <cours>3</cours>

```

```

<date>jeu 22/5/2014 14:1</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/le_protocole_contractn
    et_de_jade.html </info>
<date_second>1400760113</date_second>
<duree_second>448</duree_second>
</log>
-<log>
  <id>547</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:9</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/outils_de_dbogage_de
    _jade.html </info>
  <date_second>1400760561</date_second>
  <duree_second>189</duree_second>
</log>
-<log>
  <id>550</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:12</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/comportements_des_agents_d
    ans_la_plateforme_jade.html </info>
  <date_second>1400760750</date_second>
  <duree_second>191</duree_second>
</log>
-<log>
  <id>552</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:15</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/les_protocoles_fipaquery_
    et_fiparequest.html </info>
  <date_second>1400760941</date_second>
  <duree_second>337</duree_second>
</log>
-<log>
  <id>559</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:21</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/langage_de_communic

```

```

    ation_de_la_plateforme_jade.html</info>
    <date_second>1400761278</date_second>
    <duree_second>568</duree_second>
  </log>
-<log>
  <id>566</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:30</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html
  </info>
  <date_second>1400761846</date_second>
  <duree_second>12</duree_second>
</log>
-<log>
  <id>568</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:30</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html
  </info>
  <date_second>1400761858</date_second>
  <duree_second>308</duree_second>
</log>
-<log>
  <id>576</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:36</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/les_protocoles_fipaquery_et
    _fiparequest.html </info>
  <date_second>1400762166</date_second>
  <duree_second>21</duree_second>
</log>
-<log>
  <id>579</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:36</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/rfrences.html</info>
  <date_second>1400762187</date_second>
  <duree_second>50</duree_second>
</log>
-<log>
  <id>585</id>

```

```

<cours>3</cours>
<date>jeu 22/5/2014 14:37</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
  http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html</info>
<date_second>1400762237</date_second>
<duree_second>42</duree_second>
</log>
-<log>
  <id>591</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:37</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html</info>
  <date_second>1400762279</date_second>
  <duree_second>13</duree_second>
</log>
-<log>
  <id>594</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:38</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html</info>
  <date_second>1400762292</date_second>
  <duree_second>23</duree_second>
</log>
-<log>
  <id>596</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:38</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communicatio
    n_de_la_plateforme_jade.html</info>
  <date_second>1400762315</date_second>
  <duree_second>34</duree_second>
</log>
-<log>
  <id>599</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:39</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>

```



```

    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/les_protocoles_fipaquery
    _et_fiparequest.html</info>
    <date_second>1400762349</date_second>
    <duree_second>117</duree_second>
  </log>
-<log>
  <id>609</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:41</date>
  <action>view</action>
  <ip>127.0.0.1</ip>
  <info>3</info>
  <date_second>1400762466</date_second>
  <duree_second>14</duree_second>
</log>
-<log>
  <id>611</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:41</date>
  <action>pre-view</action>
  <ip>127.0.0.1</ip>
  <info>1</info>
  <date_second>1400762480</date_second>
  <duree_second>15</duree_second>
</log>
-<log>
  <id>613</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:41</date>
  <action>view</action>
  <ip>127.0.0.1</ip>
  <info>1</info>
  <date_second>1400762495</date_second>
  <duree_second>6</duree_second>
</log>
-<log>
  <id>615</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:41</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/architecture_logiciel_de_l
    a_plateforme_jade.html</info>
    <date_second>1400762501</date_second>
    <duree_second>21</duree_second>
  </log>
-<log>
  <id>617</id>
  <cours>3</cours>

```

```

<date>jeu 22/5/2014 14:42</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/langage_de_communic
    ation_de_la_plateforme_jade.html</info>
<date_second>1400762522</date_second>
<duree_second>13</duree_second>
</log>
-<log>
    <id>619</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:42</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/les_protocoles_fipaquery
        _et_fiparequest.html </info>
    <date_second>1400762535</date_second>
    <duree_second>5</duree_second>
</log>
-<log>
    <id>621</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:42</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/le_protocole_contractn
        et_de_jade.html </info>
    <date_second>1400762540</date_second>
    <duree_second>7</duree_second>
</log>
-<log>
    <id>623</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:42</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/comportements_des_agents
        _dans_la_plateforme_jade.html</info>
    <date_second>1400762547</date_second>
    <duree_second>8</duree_second>
</log>
-<log>
    <id>625</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:42</date>
    <action>launch</action>

```

```

<ip>127.0.0.1</ip>
<info>
  http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_j
  ade.html</info>
<date_second>1400762555</date_second>
<duree_second>6</duree_second>
</log>
-<log>
  <id>627</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:42</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/rfrences.html</info>
  <date_second>1400762561</date_second>
</log>
</logs>

```

Cas Source 2

```

-<logs>
-</user>
  <id>3</id>
  <nom>Cas source2</nom>
</user>
-<log>
  <id>510</id>
  <cours>1</cours>
  <date>jeu 22/5/2014 13:44</date>
  <action>login</action>
  <ip>127.0.0.1</ip>
  <info>3</info>
  <date_second>1400759069</date_second>
  <duree_second>77</duree_second>
</log>
-<log>
  <id>524</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:45</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html</info>
  <date_second>1400759146</date_second>
  <duree_second>498</duree_second>
</log>
-<log>
  <id>532</id>
  <cours>3</cours>

```

```
<date>jeu 22/5/2014 13:54</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/la_norme_fipa.html
</info>
  <date_second>1400759644</date_second>
  <duree_second>309</duree_second>
</log>
-<log>
  <id>536</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:59</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/comportements_des_agents_dans_la_plateforme_jade.html</info>
  <date_second>1400759953</date_second>
  <duree_second>147</duree_second>
</log>
-<log>
  <id>540</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:1</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/la_norme_fipa.html</info>
  <date_second>1400760100</date_second>
  <duree_second>198</duree_second>
</log>
-<log>
  <id>542</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:4</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/architecture_logiciel_de_la_plateforme_jade.html</info>
  <date_second>1400760298</date_second>
  <duree_second>249</duree_second>
</log>
-<log>
  <id>545</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:9</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
```

```

    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_
    jade.html</info>
    <date_second>1400760547</date_second>
    <duree_second>195</duree_second>
  </log>
-<log>
  <id>548</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:12</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comportements_des_ag
    ents_dans_la_plateforme_jade.html</info>
    <date_second>1400760742</date_second>
    <duree_second>185</duree_second>
  </log>
-<log>
  <id>551</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:15</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractn
    et_de_jade.html</info>
    <date_second>1400760927</date_second>
    <duree_second>172</duree_second>
  </log>
-<log>
  <id>554</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:18</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/architecture_logiciel_d
    e_la_plateforme_jade.html</info>
    <date_second>1400761099</date_second>
    <duree_second>175</duree_second>
  </log>
-<log>
  <id>558</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:21</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communicat
    ion_de_la_plateforme_jade.html</info>

```

```
<date_second>1400761274</date_second>
<duree_second>484</duree_second>
</log>
-<log>
  <id>562</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:29</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/la_norme_fipa.html</info>
  <date_second>1400761758</date_second>
  <duree_second>364</duree_second>
</log>
-<log>
  <id>569</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:35</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/les_protocoles_fipaquery
    _et_fiparequest.html</info>
  <date_second>1400762122</date_second>
  <duree_second>19</duree_second>
</log>
-<log>
  <id>572</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:35</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet
    _de_jade.html</info>
  <date_second>1400762141</date_second>
  <duree_second>20</duree_second>
</log>
-<log>
  <id>575</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:36</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communicat
    ion_de_la_plateforme_jade.html </info>
  <date_second>1400762161</date_second>
  <duree_second>23</duree_second>
</log>
```



```
-<log>
  <id>578</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:36</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/rfrences.html</info>
  <date_second>1400762184</date_second>
  <duree_second>33</duree_second>
</log>
-<log>
  <id>582</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:36</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/index.html</info>
  <date_second>1400762217</date_second>
  <duree_second>40</duree_second>
</log>
-<log>
  <id>587</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:37</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/bref_description_de_jade
    .html </info>
  <date_second>1400762257</date_second>
  <duree_second>18</duree_second>
</log>
-<log>
  <id>590</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:37</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/la_norme_fipa.html</info>
  <date_second>1400762275</date_second>
  <duree_second>12</duree_second>
</log>
-<log>
  <id>593</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:38</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
```

```

    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/architecture_logiciel_de_
    la_plateforme_jade.html</info>
    <date_second>1400762287</date_second>
    <duree_second>14</duree_second>
  </log>
-<log>
  <id>595</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:38</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/langage_de_communicat
    ion_de_la_plateforme_jade.html</info>
    <date_second>1400762301</date_second>
    <duree_second>43</duree_second>
  </log>
-<log>
  <id>598</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:39</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/les_protocoles_fipaquery
    _et_fiparequest.html</info>
    <date_second>1400762344</date_second>
    <duree_second>58</duree_second>
  </log>
-<log>
  <id>602</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:40</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/le_protocole_contractn
    et_de_jade.html</info>
    <date_second>1400762402</date_second>
    <duree_second>52</duree_second>
  </log>
-<log>
  <id>607</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:40</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/comportements_des_age
    nts_dans_la_plateforme_jade.html</info>

```

```

    <date_second>1400762454</date_second>
    <duree_second>17</duree_second>
  </log>
- <log>
  <id>610</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:41</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jad
    e.html</info>
  <date_second>1400762471</date_second>
  <duree_second>15</duree_second>
</log>
- <log>
  <id>612</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:41</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/rfrences.html</info>
  <date_second>1400762486</date_second>
  <duree_second>25</duree_second>
</log>
- <log>
  <id>616</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:41</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/index.html</info>
  <date_second>1400762511</date_second>
  <duree_second>17</duree_second>
</log>
- <log>
  <id>618</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:42</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html
  </info>
  <date_second>1400762528</date_second>
  <duree_second>10</duree_second>
</log>
- <log>
  <id>620</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:42</date>

```

```

<action>launch</action>
<ip>127.0.0.1</ip>
<info>
  http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/bref_description_de_jade.html</info>
<date_second>1400762538</date_second>
<duree_second>5</duree_second>
</log>
-<log>
  <id>622</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:42</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/la_norme_fipa.html</info>
  <date_second>1400762543</date_second>
  <duree_second>7</duree_second>
</log>
-<log>
  <id>624</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:42</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/architecture_logiciel_de_la_pl
    ateforme_jade.html</info>
  <date_second>1400762550</date_second>
  <duree_second>8</duree_second>
</log>
-<log>
  <id>626</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:42</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communication_
    de_la_plateforme_jade.html</info>
  <date_second>1400762558</date_second>
  <duree_second>6</duree_second>
</log>
-<log>
  <id>628</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:42</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>

```

```

http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/les_protocoles_fipaquery_et_f
iparequest.html </info>
<date_second>1400762564</date_second>
<duree_second>276</duree_second>
</log>
-<log>
  <id>630</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:47</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet_de_j
ade.html</info>
  <date_second>1400762840</date_second>
  <duree_second>16</duree_second>
</log>
-<log>
  <id>631</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:47</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jade.h
tml</info>
  <date_second>1400762856</date_second>
  <duree_second>5</duree_second>
</log>
-<log>
  <id>632</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:47</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/rfrences.html</info>
  <date_second>1400762861</date_second>
  <duree_second>4</duree_second>
</log>
-<log>
  <id>633</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:47</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/architecture_logiciel_de_la_pl
ateforme_jade.html</info>
  <date_second>1400762865</date_second>
  <duree_second>4</duree_second>

```

```

</log>
-<log>
  <id>634</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:47</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communication_
    de_la_plateforme_jade.html</info>
  <date_second>1400762869</date_second>
</log>
</logs>

```

Cas Source 3

```

-<logs>
-</user>
  <id>5</id>
  <nom>Cas source3</nom>
</user>
-<log>
  <id>512</id>
  <cours>1</cours>
  <date>jeu 22/5/2014 13:44</date>
  <action>login</action>
  <ip>127.0.0.1</ip>
  <info>5</info>
  <date_second>1400759087</date_second>
  <duree_second>91</duree_second>
</log>
-<log>
  <id>529</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:46</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/index.html</info>
  <date_second>1400759178</date_second>
  <duree_second>5</duree_second>
</log>
-<log>
  <id>530</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:46</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/bref_description_de_jade
    .html </info>
  <date_second>1400759183</date_second>

```



```
<duree_second>436</duree_second>
</log>
-<log>
  <id>531</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:53</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/rfrences.html </info>
  <date_second>1400759619</date_second>
  <duree_second>106</duree_second>
</log>
-<log>
  <id>534</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:55</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/introduction.html
  </info>
  <date_second>1400759725</date_second>
  <duree_second>77</duree_second>
</log>
-<log>
  <id>535</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 13:56</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/rfrences.html </info>
  <date_second>1400759802</date_second>
  <duree_second>203</duree_second>
</log>
-<log>
  <id>539</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:0</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/la_norme_fipa.html
  </info>
  <date_second>1400760005</date_second>
  <duree_second>333</duree_second>
</log>
-<log>
  <id>544</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:5</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
```

```

<info>
  http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html</info>
<date_second>1400760338</date_second>
<duree_second>217</duree_second>
</log>
-<log>
  <id>546</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:9</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/ouils_de_dbogage_de_j
    ade.html </info>
  <date_second>1400760555</date_second>
  <duree_second>194</duree_second>
</log>
-<log>
  <id>549</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:12</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comportements_des_age
    nts_dans_la_plateforme_jade.html </info>
  <date_second>1400760749</date_second>
  <duree_second>208</duree_second>
</log>
-<log>
  <id>553</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:15</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communicatio
    n_de_la_plateforme_jade.html </info>
  <date_second>1400760957</date_second>
  <duree_second>176</duree_second>
</log>
-<log>
  <id>556</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:18</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/la_norme_fipa.html
  </info>
  <date_second>1400761133</date_second>

```

```

    < duree_second>132</duree_second>
  </log>
-<log>
  <id>557</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:21</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communicat
    ion_de_la_plateforme_jade.html </info>
    <date_second>1400761265</date_second>
    <duree_second>484</duree_second>
  </log>
-<log>
  <id>561</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:29</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/la_norme_fipa.html
  </info>
  <date_second>1400761749</date_second>
  <duree_second>381</duree_second>
</log>
-<log>
  <id>570</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:35</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/les_protocoles_fipaquery_
    et_fiparequest.html </info>
    <date_second>1400762130</date_second>
    <duree_second>11</duree_second>
  </log>
-<log>
  <id>571</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:35</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet_
    de_jade.html </info>
    <date_second>1400762141</date_second>
    <duree_second>13</duree_second>
  </log>
-<log>

```

```

<id>574</id>
<cours>3</cours>
<date>jeu 22/5/2014 14:35</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/architecture_logiciel_de_
    la_plateforme_jade.html </info>
    <date_second>1400762154</date_second>
    <duree_second>20</duree_second>
</log>
-<log>
    <id>577</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:36</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/les_protocoles_fipaue
        ry_et_fiparequest.html </info>
        <date_second>1400762174</date_second>
        <duree_second>34</duree_second>
    </log>
-<log>
    <id>580</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:36</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/introduction.html
    </info>
    <date_second>1400762208</date_second>
    <duree_second>0</duree_second>
    </log>
-<log>
    <id>581</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:36</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scm/content/1/introduction.html
    </info>
    <date_second>1400762208</date_second>
    <duree_second>23</duree_second>
    </log>
-<log>
    <id>584</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:37</date>
    <action>launch</action>

```

```

<ip>127.0.0.1</ip>
<info>
  http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/bref_description_de_jade
  .html </info>
<date_second>1400762231</date_second>
<duree_second>19</duree_second>
</log>
-<log>
  <id>586</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:37</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/la_norme_fipa.html
  </info>
  <date_second>1400762250</date_second>
  <duree_second>20</duree_second>
</log>
-<log>
  <id>589</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:37</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/architecture_logiciel_de_
    la_plateforme_jade.html </info>
  <date_second>1400762270</date_second>
  <duree_second>14</duree_second>
</log>
-<log>
  <id>592</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:38</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/langage_de_communicatio
    n_de_la_plateforme_jade.html </info>
  <date_second>1400762284</date_second>
  <duree_second>58</duree_second>
</log>
-<log>
  <id>597</id>
  <cours>3</cours>
  <date>jeu 22/5/2014 14:39</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/les_protocoles_fipaquery

```

```

        _et_fiparequest.html </info>
        <date_second>1400762342</date_second>
        <duree_second>59</duree_second>
    </log>
-<log>
    <id>601</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:40</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractn
        et_de_jade.html </info>
    <date_second>1400762401</date_second>
    <duree_second>48</duree_second>
</log>
-<log>
    <id>606</id>
    <cours>3</cours>
    <date>jeu 22/5/2014 14:40</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>
    <info>
        http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/comportements_des_age
        nts_dans_la_plateforme_jade.html </info>
    <date_second>232</date_second>
</log>
</logs>

```

Cas Source 4

```

-<logs>
    -</user>
    <id>8</id>
    <nom>Cas source4</nom>
</user>
-<log>
    <id>1</id>
    <cours>1</cours>
    <date>jeu 29/5/2014 15:30</date>
    <action>login</action>
    <ip>127.0.0.1</ip>
    <info>6</info>
    <date_second>1401449562</date_second>
    <duree_second>211</duree_second>
</log>
-<log>
    <id>2</id>
    <cours>3</cours>
    <date>jeu 29/5/2014 15:34</date>
    <action>launch</action>
    <ip>127.0.0.1</ip>

```



```

<info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/introduction.html</info>
<date_second>1401449773</date_second>
<duree_second>365</duree_second>
</log>
-<log>
<id>3</id>
<cours>3</cours>
<date>jeu 29/5/2014 15:40</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jad
e.html </info>
<date_second>1401450138</date_second>
<duree_second>261</duree_second>
</log>
-<log>
<id>4</id>
<cours>3</cours>
<date>jeu 29/5/2014 15:44</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet
_de_jade.html </info>
<date_second>1401450399</date_second>
<duree_second>352</duree_second>
</log>
-<log>
<id>5</id>
<cours>3</cours>
<date>jeu 29/5/2014 15:50</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jad
e.html </info>
<date_second>1401450751</date_second>
<duree_second>182</duree_second>
</log>
-<log>
<id>6</id>
<cours>3</cours>
<date>jeu 29/5/2014 15:53</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet
_de_jade.html </info>

```

```

    <date_second>1401450933</date_second>
    <duree_second>196</duree_second>
  </log>
-<log>
  <id>7</id>
  <cours>3</cours>
  <date>jeu 29/5/2014 15:56</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/introduction.html
  </info>
  <date_second>1401451129</date_second>
  <duree_second>161</duree_second>
</log>
-<log>
  <id>8</id>
  <cours>3</cours>
  <date>jeu 29/5/2014 15:59</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/comportements_des_age
    nts_dans_la_plateforme_jade.html </info>
  <date_second>1401451290</date_second>
  <duree_second>158</duree_second>
</log>
-<log>
  <id>9</id>
  <cours>3</cours>
  <date>jeu 29/5/2014 16:2</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/le_protocole_contractnet_
    de_jade.html </info>
  <date_second>1401451448</date_second>
  <duree_second>214</duree_second>
</log>
-<log>
  <id>10</id>
  <cours>3</cours>
  <date>jeu 29/5/2014 16:5</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/32/mod_scorum/content/1/introduction.html
  </info>
  <date_second>1401451662</date_second>
  <duree_second>361</duree_second>
</log>
-<log>

```

```

<id>11</id>
<cours>3</cours>
<date>jeu 29/5/2014 16:11</date>
<action>launch</action>
<ip>127.0.0.1</ip>
<info>
  http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/le_protocole_contractnet_
  de_jade.html </info>
<date_second>1401452023</date_second>
<duree_second>10</duree_second>
</log>
-<log>
  <id>12</id>
  <cours>3</cours>
  <date>jeu 29/5/2014 16:12</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info> http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/rfrences.html </info>
  <date_second>1401452033</date_second>
  <duree_second>60</duree_second>
</log>
-<log>
  <id>13</id>
  <cours>3</cours>
  <date>jeu 29/5/2014 16:13</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>
    http://127.0.0.1/moodle/pluginfile.php/32/mod_scorm/content/1/outils_de_dbogage_de_jad
    e.html </info>
    <date_second>1401452046</date_second>
    <duree_second>384</duree_second>
</log>
-<log>
  <id>14</id>
  <cours>3</cours>
  <date>jeu 29/5/2014 16:20</date>
  <action>view</action>
  <ip>127.0.0.1</ip>
  <info>3</info>
  <date_second>1401452430</date_second>
  <duree_second>2504</duree_second>
  <duree_second>602</duree_second>
</log>
-<log>
  <id>15</id>
  <cours>IDCBR-MAS</cours>
  <date>jeu 29/5/2014 16:30</date>
  <action>login</action>
  <ip>127.0.0.1</ip>

```

```
<info>6</info>
<date_second>1405043005</date_second>
<duree_second>296</duree_second>
</log>
-<log>
  <id>16</id>
  <cours>IDCBR-MAS</cours>
  <date>jeu 29/5/2014 16:35</date>
  <action>view</action>
  <ip>127.0.0.1</ip>
  <info>1</info>
  <date_second>1405043008</date_second>
  <duree_second>114</duree_second>
</log>
-<log>
  <id>17</id>
  <cours>Cours JADE</cours>
  <date>jeu 29/5/2014 16:37</date>
  <action>view</action>
  <ip>127.0.0.1</ip>
  <info>3</info>
  <date_second>1405043012</date_second>
  <duree_second>180</duree_second>
</log>
-<log>
  <id>18</id>
  <cours>Cours JADE</cours>
  <date>jeu 29/5/2014 16:40</date>
  <action>pre-view</action>
  <ip>127.0.0.1</ip>
  <info>1</info>
  <date_second>1405043018</date_second>
  <duree_second>120</duree_second>
</log>
-<log>
  <id>19</id>
  <cours>Cours JADE</cours>
  <date>jeu 29/5/2014 16:42</date>
  <action>view</action>
  <ip>127.0.0.1</ip>
  <info>1</info>
  <date_second>1405043023</date_second>
  <duree_second>182</duree_second>
</log>
-<log>
  <id>20</id>
  <cours>IDCBR-MAS</cours>
  <date>jeu 29/5/2014 16:45</date>
  <action>logout</action>
  <ip>127.0.0.1</ip>
```

```
<info>8</info>  
<date_second>8</date_second>  
</log>  
</logs>
```