

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mentouri Constantine



Faculté des Sciences de l'Ingénieur
Département d'Informatique

N° Ordre:

N° Série:

Thèse

pour obtenir le **Diplôme de**
Doctorat en Sciences
Spécialité: Informatique

Titre

Apprentissage Automatique dans un Agent

Présentée par:

HAMMOUD Djamilia épouse BOUDJAADA

Soutenue le _____, devant le Jury:

Pr N. ZAROOUR	Professeur, Université-Constantine	Président-Jury
Dr R.MAAMRI	Maître de conférence, Université-Constantine	Rapporteur
Pr Z.SAHNOUN	Professeur, Université-Constantine	Co-Rapporteur
Dr F. BELALA	Maître de conférence, Université-Constantine	Examineur
Dr F.BENCHIKHA	Maître de conférence, Université-Skikda	Examineur
Dr O.KAZAR	Maître de conférence, Université-Biskra	Examineur

Remerciements

Ce travail de thèse n'aurait jamais été mené à terme sans le soutien, du Dieu le tout puissant qui m'a donnée la force pour l'achever, et d'un grand nombre de personnes qui m'ont aidée et encouragée tout au long de ce travail laborieux. Que Dieu reçoive toute ma gratitude, qui va aussi à toutes ces personnes que je tiens vivement et sincèrement à remercier.

En premier lieu, je tiens à faire part de ma reconnaissance envers le Docteur R. Maamri, directeur de l'équipe « Génie Logiciel et Intelligence Artificielle: GLIA » du laboratoire LIRE, de l'université de Constantine, qui a pris la charge de la direction de mon travail durant les dernières années de cette thèse. Je le remercie vivement pour sa disponibilité, pour ses conseils pertinents dont il m'a fait bénéficier avec une grande attention et patience. Qu'il trouve dans ces mots concis, l'expression de ma profonde gratitude.

J'adresse aussi mes sincères remerciements au Professeur Z. Sahnoun, le directeur initial de cette thèse; qui m'a accueillie comme membre de l'équipe GLIA. Je tiens à lui exprimer ma profonde gratitude pour ses conseils bénéfiques et ses critiques fructueux, pour son aide, sa patience et son attention bienveillante. Qu'il trouve dans ces courtes lignes, le témoignage de ma reconnaissance.

Je tiens à remercier vivement le Professeur N. Zarour, de l'université de Constantine, pour m'avoir fait l'honneur d'évaluer mon travail et de présider le jury de cette thèse.

J'adresse mes sincères remerciements au Docteur F. Belala, de l'université de Constantine, pour m'avoir fait l'honneur de participer au jury de cette thèse et d'examiner mon travail.

Je tiens à remercier profondément le Docteur F. Benchikha, de l'université de Skikda, pour m'avoir fait l'honneur d'examiner mon travail et de participer au jury de cette thèse.

Je suis tout à fait honorée de la présence à ce jury du Docteur O. Kezar, de l'université de Biskra, je le remercie vivement pour avoir examiné mon travail et participer à son jury.

Je tiens aussi à remercier le directeur du laboratoire LIRE, le Professeur M. Boufaïda qui œuvre pour améliorer les conditions de travail et promouvoir la recherche scientifique.

Mes profonds et sincères remerciements vont aussi à tous mes collègues et amies du département d'informatique pour leur soutien et encouragements; ainsi qu'à toutes mes amies.

Enfin, c'est avec une grande émotion que j'évoque ici l'aide et le soutien permanents des membres de ma famille; qui n'ont pas cessé de me soutenir et de m'encourager. Je tiens à leur exprimer tous mes sincères, profonds et chaleureux remerciements et je ne peux m'empêcher de les citer. A commencer par mon père, qui m'a toujours encouragée et bénie, qu'il repose en paix. Ma mère qui s'est toujours dévouée et sacrifiée pour ses enfants, que Dieu la protège. Ma belle mère, qu'elle repose aussi en paix. Mon mari pour qui ma gratitude est bien grande, son aide m'est très précieuse; il a toujours été présent pour m'aider et me soutenir du mieux qu'il pouvait. Mes filles: Lina, Soumeïya et Yousra qui ont su être patientes et compréhensives, malgré leur jeune âge. Mes sœurs et belles sœurs qui m'ont toujours encouragée, particulièrement mes frangines: Hakima et Fouzia ainsi que son mari qui nous a toujours soutenus. Mes frères, beaux frères, neveux et nièces. L'ampleur de mes sentiments à l'égard des membres de ma famille, ne peut être exprimée par ces quelques mots, mais qu'ils trouvent dans la dédicace de cette thèse un témoignage de ma profonde reconnaissance et gratitude pour tous ce qu'ils m'ont apportée tout au long de cette thèse.

Résumé

Actuellement, tant pour la robotique que pour le développement de logiciels, l'un des principaux défis est de créer des systèmes informatiques capables d'apprendre à agir d'une façon appropriée, en présentant de l'autonomie, de l'initiative, de l'intelligence et de l'adaptation. La technologie Agent, l'intelligence artificielle et l'apprentissage automatique peuvent coopérer pour lever ce défi. Cette thèse s'inscrit dans le cadre général de ces trois disciplines et plus précisément, elle concerne l'apprentissage automatique dans un Agent.

Le premier objectif de ce travail, est l'établissement d'une mise en correspondance entre les types d'Agents-Apprenants et les méthodes d'apprentissage automatique. Pour atteindre ce but, d'une part une détermination des types d'Agents nécessitant le recours à l'apprentissage automatique, basée sur un ensemble de critères de classification d'Agents, a été accomplie. D'autre part, confrontés à une pléthore de méthodes d'apprentissage, nous avons caractérisé ces méthodes et étudié les plus représentatives pour les approches d'apprentissage déléguées.

Le second objectif, de cette thèse, est la construction d'un modèle pour l'apprentissage automatique dans un système basé Agent. Par conséquent, nous proposons un modèle général et générique pour les systèmes basés Agent-Apprenant. Ce modèle est indépendant de tout domaine et il peut être instancié par diverses applications, il couvre les différentes approches et méthodes de l'apprentissage automatique. Le modèle proposé se base sur une assise conceptuelle formée par un ensemble de concepts renforçant ses propriétés de généricité et de généralité. Afin d'exploiter le modèle établi et le valider, des instanciations et des expérimentations ont été menées. Ces dernières ont permis de mettre en œuvre les différents composants ainsi que les éléments de la base conceptuelle du modèle proposé.

Mots clés: Technologie Agent, Apprentissage Automatique, Agent-Apprenant, Modèle pour les Systèmes basés Agent-Apprenant, Instanciation du Modèle Général et Générique.

Abstract

Currently, so much for robotics than for software development, one of the most challenges is to create information processing systems able to learn how to act in an adapted way, by presenting autonomy, initiative, intelligence and adaptation. Agent technology, artificial intelligence and machine learning can cooperate to raise this challenge. This thesis lies within the general scope of these three disciplines and more precisely, it relates to machine learning in an Agent.

The first objective of this work is the establishment of a mapping between the learning Agent's types and machine learning methods. To achieve this goal, on one hand a determination of Agent's types requiring the recourse to machine learning, based on a set of Agent's classification criteria, was accomplished. On the other hand, confronted to a plethora of machine learning methods, we have characterized these methods and studied the most representative for the released learning approaches.

This thesis second objective is the construction of a model for machine learning in an Agent based system. Consequently, we propose a general and generic model for learning Agent based systems. This model is independent of any field and it can be instantiated by various applications, it covers different machine learning approaches and methods. The proposed model is based on a conceptual base formed by a set of concepts reinforcing its general and generic properties. In order to exploit the established model and to validate it, some instantiations and experimentations were carried out. These last made it possible to experiment the various components as well as the elements of the proposed model conceptual base.

Key words: Agent Technology, Machine Learning, Learning-Agent, Model for Learning Agent based Systems, Instantiation of the General and Generic Model.

ملخص

حالياً، سواء بالنسبة للأنظمة الآلية أو لتطوير البرمجيات، من بين التحديات الرئيسية، إنشاء نظم الكمبيوتر القادرة على تعلم كيفية العمل المناسبة، عن طريق التحكم الذاتي، المبادرة، الذكاء والتكيف. تكنولوجيا العامل أو الوكيل، الذكاء الاصطناعي والتعلم الآلي يمكنها التعاون للتغلب على هذا التحدي. هذه الأطروحة تدرج ضمن الإطار العام لهذه التخصصات الثلاث وبشكل أكثر تحديداً، تهتم بالتعلم الآلي لدى

يتمثل الهدف الأول من هذا العمل في إنشاء تطابق و انسجام بين أنواع العمال أو الوكلاء المتعلمين وأساليب لتحقيق هذه الغاية، أو لا تم تحديد أنواع الوكلاء الذين يتطلبون استخدام التعلم الآلي، على أساس مجموعة من المعايير لتصنيفهم. من ناحية أخرى، عند مواجهة عدد كبير من أساليب التعلم الآلي، قمنا بتصنيف هذه الأساليب ودراسة الأكثر تمثيلاً لمنهجيات التعلم الآلي المتحصل عليها. الهدف الثاني لهذه الأطروحة هو بناء نموذج للتعلم الآلي في نظام يستند إلى عامل أو وكيل. لذا، نقترح نموذجاً عاماً للأنظمة المستعملة للوكيل القائم على التعلم. هذا النموذج مستقل عن أي ميدان، ويمكن تمثيله بتطبيقات مختلفة، وهو يغطي مختلف المناهج وطرق التعلم الآلي. ويستند النموذج المقترح على أساس مجموعة من المفاهيم التي تعزز صفة عموميته. والتحقق من صحته، تجسيدات و تجارب تم إجراءها. ساعدت هذه الأخيرة لتنفيذ مختلف مكونات وعناصر من الأساس المفاهيمي للنموذج المقترح.

نموذج للأنظمة المستعملة للوكيل

الوكيل، التعلم الآلي،

الكلمات المفتاحية: تكنولوجيا

تجسيد

Table des Matières

Introduction Générale

1. Contexte du travail.....	8
2. Problématique	10
3. Objectifs.....	12
4. Organisation de la Thèse.....	12

Chapitre 1: La Technologie Agent et L'Apprentissage Automatique

1.2. Introduction.....	14
1.2. Intelligence Artificielle.....	15
1.2.1. Définitions de L'Intelligence Artificielle.....	16
1.2.2. Intelligence Artificielle Distribuée.....	17
1.3. La Technologie Agent	19
1.3.1. Historique.....	20
1.3.2. Qu'est ce qu'un Agent ?.....	21
1.3.3. Caractéristiques Essentielles d'un Agent	24
1.3.4. Systèmes Multi-Agents	25
1.3.5. Approches d'Agents.....	26
1.3.5.1. Approche Réactive	27
1.3.5.2. Approche Cognitive.....	27
1.3.5.3. Approche Hybride.....	27
1.3.6. Architectures des Agents.....	28
1.3.6.1. Architectures Réactives	28
1.3.6.2. Architectures Cognitives.....	29
1.3.6.3. Architectures Hybrides.....	32
1.3.7. Applications de la Technologie Agent.....	33

1.4. Apprentissage Automatique.....	34
1.4.1 Apprentissage Naturel.....	35
1.4.2. Apprentissage automatique: Quelques Définitions.....	36
1.4.3. Apprentissage automatique: Apports et Motivations.....	37
1.5. Apprentissage Automatique dans un Agent.....	40
1.5.1. Travaux Existants	40
1.5.2. Synthèse	43
1.5.3. Objectifs de Notre Travail.....	44
1.6. Conclusion	45

Chapitre 2: Approches et Méthodes de l'Apprentissage Automatique

2.1. Introduction	47
2.2. Apprentissage Supervisé.....	48
2.3. Apprentissage non-Supervisé.....	49
2.4. Approches de l'Apprentissage Automatique	50
2.4.1 Approche d'Apprentissage Inductive	51
2.4.2 Approche d'Apprentissage Déductive	52
2.4.3. Approche d'Apprentissage Par Renforcement.....	52
2.4.4. Approche d'Apprentissage Hybride.....	54
2.5. Méthodes d'Apprentissage Inductives.....	55
2.5.1. Apprentissage par Arbres de Décisions	56
2.5.2. Apprentissage par Réseaux de Neurones Artificiels.....	58
2.5.3. Apprentissage par Algorithmes Génétiques.	60
2.5.4. Apprentissage par Programmation Logique Inductive.	62
2.5.5. Apprentissage basé sur l'Analogie.....	64
2.6. Méthodes d'Apprentissage Déductives.....	66
2.6.1. Apprentissage basé Explication.....	67
2.7. Méthodes d'Apprentissage par Renforcement.....	68
2.7.1. Le Q-Learning.....	68
2.8. Méthodes d'Apprentissage Hybrides.....	69
2.8.1. Les Réseaux KBANN.....	69
2.9. Conclusion	70

Chapitre 3 : Mise en Correspondance entre Agents Apprenants et Approches/Méthodes d'Apprentissage

3.1. Introduction.....	72
3.2. Critères de Classification d'Agents.....	73
3.2.1. L'Autonomie.....	74
3.2.2. La Réactivité et La Pro-activité.....	74
3.2.3. L'Intelligence.....	75
3.2.4. La Flexibilité.....	75
3.2.5. L'Adaptation.....	76
3.2.6. L'Apprentissage automatique.....	76
3.2.7. Les Services de l'Agent.....	77
3.2.8. La Mobilité.....	77
3.3. Détermination des Types d'Agents-Apprenants.....	77
3.3.1. Agent Réactif Simple.....	78
3.3.2. Agent Réactif Adaptatif.....	79
3.3.3. Agent Intelligent.....	79
3.3.4. Agent Interface ou Personnel.....	80
3.3.5. Agents d'Informations.....	81
3.3.6. Agent Mobile.....	81
3.3.7. Agent Hybride.....	82
3.3.8. Systèmes basés Agent-Apprenant.....	82
3.4. Mise en Correspondance entre Agents-Apprenants et les Approches/Méthodes d'Apprentissage.....	84
3.4.1. Agent Réactif Adaptatif et Approches/Méthodes d'Apprentissage.....	85
3.4.2. Systèmes basés Agent-Apprenant et Approches/Méthodes d'Apprentissage.....	87
3.4.2.1 Systèmes basés Agent-Apprenant et l'Approche Dédutive.....	88
3.4.2.2 Systèmes basés Agent-Apprenant et l'Approche Inductive.....	89
3.4.2.3. Systèmes basés Agent-Apprenant et l'Approche Hybride.....	91
3.4.2.4. Systèmes basés Agent-Apprenant et Méthodes d'Apprentissage.....	93
3.5. Conclusion.....	96

Chapitre 4: Modèle Général et Générique de Systèmes basés-Agent Apprenant

4.1. Introduction.....	97
4.2. Pourquoi un Modèle de Systèmes basés-Agent Apprenant?.....	98
4.3. Principales Activités de la Modélisation.....	99

4.4. Fonctionnalités Principales d'un Système Général basé Agent-Apprenant Générique.....	100
4.5. Concepts de base du Modèle Général de Systèmes basés Agent-Apprenant.....	103
4.5.1. La Classe des Tâches.....	105
4.5.2. Une Mesure de Performance.....	105
4.5.3. Les Exemples d'Entraînement.....	106
4.5.4. Les Connaissances Préalables.....	107
4.5.5. La Représentation des Connaissances.....	107
4.5.6. Une Approche /Méthodes d'Apprentissage.....	108
4.6. Composants du Modèle des Systèmes basés Agent-Apprenant.....	109
4.6.1. Composant Interface.....	110
4.6.2. Composant Application.....	111
4.6.3. Composant Perception.....	111
4.6.4. Composant Bases d'Apprentissage.....	111
4.6.5. Composant Apprentissage Automatique.....	113
4.6.6. Composant Connaissances.....	115
4.6.7. Composant Raisonnement.....	116
4.6.8. Composant Evaluation Performance.....	117
4.7. Conclusion.....	118

Chapitre 5 : Instanciation du Modèle Général et Générique: Expérimentations et Etudes de Cas

5.1. Introduction.....	119
5.2. Instanciation du Modèle Général et Générique par le Domaine des E-mails.....	120
5.2.1. Architecture du Système basé Agent-Apprenant pour la Prise en Charge Intelligente des E-mails.....	120
5.2.1.1 Instanciation de l'Environnement.....	122
5.2.1.2. Composant Interface.....	122
5.2.1.3. Composant Perception.....	123
5.2.1.4. Composant Notification.....	123
5.2.1.5. Composant Extraction Connaissances.....	124
5.2.1.6. Base de Faits.....	124
5.2.1.7. Composant Bases d'Apprentissage.....	124
5.2.1.8. Composant Application.....	126
5.2.1.9. Composant Connaissances.....	127
5.2.1.10. Composant Raisonnement.....	128
5.2.1.11. Composant Apprentissage Automatique.....	130
5.2.1.12. Composant Evaluation Performance.....	134

5.3. Instanciation du Modèle Général et Générique	
par le Domaine des Pathologies des Plantes.....	137
5.3.1. La Plateforme JADE.....	138
5.3.2 Architecture du Système basé Agent-Apprenant pour le Diagnostic	
des Pathologies des Plantes	140
5.3.2.1. Instanciation de l'Environnement	141
5.3.2.2. Composant Interface	141
5.3.2.3. Composant Perception	141
5.3.2.4. Composant Bases d'Apprentissage	141
5.3.1.5. Composant Application.....	143
5.3.1.6. Composant Connaissances.....	144
5.3.1.7. Composant Raisonnement.....	146
5.3.1.8. Composant Apprentissage Automatique	147
5.3.1.9. Composant Evaluation Performance.....	151
5.4. Conclusion.....	152
Conclusion Générale	
1. Introduction.....	154
2. Contributions.....	155
3. Perspectives.....	157
Références et Bibliographie.....	159

Listes des Figures et Tables

Figure 1.1 : Schématisation d'un Agent.	23
Figure 1.2: Architecture réactive de subsomption.	29
Figure 1.3 : Architecture BDI d'un Agent.	31
Figure 1.4: Architecture InteRRaP.	32
Figure 1.5: Aspect pluridisciplinaire de l'apprentissage automatique.	36
Figure 2.1 : Schématisation de l'apprentissage par renforcement.....	53
Figure 2.2: Arbre de décision correspondant à la table 2.1 [Quinlan, 1986].....	57
Figure 2.3: Eléments constitutifs d'un neurone artificiel.....	58
Figure 2.4: Différentes architectures de RNA.....	60
Figure 2.5: Principe général des algorithmes génétiques.....	61
Figure 2.6: Apprentissage par analogie.....	65
Figure 2.7: Principe des méthodes EBL.	67
Figure 3.1: Critères de classification d'Agents.....	74
Figure 3.2: Une typologie d'Agents.....	78
Figure 3.3: Les différents types d'Agents-Apprenants.....	82
Figure 3.4: Caractéristiques des systèmes basés Agent-Apprenant.....	84
Figure 4.1: Vue globale du modèle général.	101
Figure 4.2 : Composants du modèle général et générique de systèmes basé Agent-Apprenant.	110
Figure 4.3 : Eléments du composant bases d'apprentissage.	112
Figure 4.4 : Généricité des approches d'apprentissage du modèle proposé.....	113
Figure 4.5 : Généricité des méthodes d'Apprentissage du modèle proposé.....	115
Figure 4.6 : Composant connaissances.....	116
Figure 5.1: Architecture de l'Agent-Apprenant de prise en charge des E-Mails.....	121
Figure 5.2: La notification de nouveaux E-mails au niveau de l'I.M.N.H.....	124
Figure 5.3: Composant bases d'apprentissage de l'I.M.N.H.....	125
Figure 5.4: Ecrans du système de prise en charge des E-mails.....	127
Figure 5.5: Composant connaissances de l'I.M.N.H.	128
Figure 5.6: Fonctionnement du composant de raisonnement de l'I.M.N.H.....	129
Figure 5.7: Composant d'apprentissage de l'I.M.N.H.....	132
Figure 5.8: Graphes des tests de la performance de l'I.M.N.H.....	136
Figure 5.9: Deux pathologies de plantes.	137
Figure 5.10: Agent-Apprenant de diagnostic des pathologies des plantes sous JADE.	139

Figure 5.11: Architecture du système pathologies-plantes.....	140
Figure 5.12: C.B.A du système des pathologies-plantes.....	142
Figure 5.13: Ecrans du système des pathologies-plantes.....	144
Figure 5.14: Composant connaissances du système pathologies-plantes.....	145
Figure 5.15: Composant raisonnement du système pathologies-plantes.....	146
Figure 5.16: Composant d'apprentissage du système pathologies-plantes.....	149
Figure 5.17: Graphes des tests de la performance du système pathologies-plantes.....	152
Table 2.1: Exemples d'entraînement de données météorologiques [Quinlan, 1986].....	57
Table 3.1 : Les entrées des approches d'apprentissage automatique.....	93

Introduction Générale

1. Contexte du Travail

Les innovations technologiques sont historiquement parmi les grands moteurs de la transformation de notre mode de vie. Les nouveaux procédés de l'information et de la communication envahissent véritablement notre quotidien; à travers, des lignes téléphoniques fixes et mobiles, des réseaux câblés, des écrans d'ordinateurs (au bureau, à la maison ou même à bord de certains véhicules), des agendas intelligents et des appareils ménagers à mémoire. Les systèmes informatiques continuent à s'intégrer de plus en plus dans les différents aspects de notre vie et dans les divers outils que nous manipulons chaque jour, afin de nous faciliter au maximum l'accès aux avancées technologiques et aux innovations récentes.

Mais ces systèmes se trouvent, parfois, limités dans les domaines qui requièrent autonomie, intelligence et adaptation dynamique. Dans l'ère de l'information, ce sont ces caractéristiques qui gagnent en importance et qui nécessitent encore l'intervention humaine. Ainsi, tant pour la robotique que pour le développement de logiciels, l'un des principaux défis est de créer des systèmes informatiques capables d'apprendre à agir d'une façon appropriée, en présentant de l'autonomie, de la flexibilité, de l'initiative, de l'intelligence, de la créativité et de l'adaptation. Il n'est pas possible de gérer avec des techniques d'ingénierie classiques les nombreuses possibilités, les déroulements et les situations inattendues qui se posent dans divers domaines.

Une technologie informatique, dite la technologie Agent, émergeant de la recherche en intelligence artificielle distribuée, augmentée par des concepts et des techniques de l'apprentissage automatique, semble lever le défi posé pour le développement de systèmes informatiques modernes.

Pour sa puissance et son grand potentiel, la technologie Agent, suscite beaucoup d'attention ces dernières années; elle est maintenant reconnue comme une technologie prometteuse pour la construction de systèmes autonomes, complexes et intelligents [Hakansson et al., 2010]. En effet d'une part, la technologie Agent offre un répertoire puissant de concepts, de paradigmes, d'outils et de techniques qui ont le potentiel d'améliorer considérablement le développement de systèmes informatiques [Demazeau et Müller, 1990; Ferber, 1995; Jennings et al., 1998; Jennings, 2001; Jennings et Wooldridge, 1998; Maes, 1990; Wooldridge, 2000].

D'autre part, l'apprentissage automatique, un sous domaine de l'intelligence artificielle permettant d'introduire de la connaissance dans des ordinateurs, est de nos jours particulièrement attrayant. Une construction automatique des connaissances est en effet nécessaire pour la résolution de problèmes complexes pour lesquels on ne trouve pas de solutions algorithmiques, qui peuvent également être mal définis ou décrits seulement de manière informelle et incomplète [Michalski et al., 1993; Mitchell, 1997; Russell et Norvig, 2006]. Ces problèmes posent de vrais goulots d'étranglement au développement de systèmes informatiques. A ces problèmes s'ajoutent les avancées technologiques, vécues actuellement, l'explosion combinatoire d'informations électroniques engendrées et les nouveaux défis précédemment soulevés. Michalski et son équipe [Michalski et al., 1998] soulignent qu'à notre époque, il est nécessaire de pouvoir extraire et rentabiliser, par apprentissage automatique, des formes significatives et des connaissances stratégiques à partir des grands stocks d'informations tenues électroniquement.

Ainsi, le recours à l'apprentissage automatique s'avère de plus en plus indispensable. Ce domaine puissant, de l'intelligence artificielle, offre des techniques et des méthodes avanguardistes capables de découvrir toutes sortes de formes remarquables et de construire un maximum de connaissances générales. Ces techniques, de l'apprentissage automatique, sont capables de tirer profit des avancées technologiques et surtout d'être une alternative, à l'approche classique, pour le développement de systèmes informatiques complexes.

La technologie Agent et l'apprentissage automatique, deux disciplines pluridisciplinaires, représentent donc le contexte et le cadre général de cette thèse. L'intégration des concepts et des techniques de l'apprentissage automatique dans la technologie Agent, permet donc de renforcer le répertoire de cette dernière et la rendre plus apte au développement de systèmes informatiques, caractérisés par un haut degré d'autonomie, d'intelligence, de flexibilité, d'apprentissage et d'adaptation.

Dans ce travail de thèse, nous nous sommes intéressés à la technologie Agent avec sa tendance individuelle; donc la tendance qui considère un Agent individuel comme une entité autonome agissant en isolation d'autres Agents artificiels. Elle s'intéresse aux mécanismes et aux connaissances relatives au fonctionnement interne de l'Agent autonome. La tendance sociale tournée vers la collectivité, les mécanismes et la communication associés concernant les activités du groupe, dit Systèmes Multi-Agents (SMA), ne fait donc pas partie de ce travail.

Un Agent autonome, ou un système basé Agent, est un système informatique situé dans un environnement. Il perçoit et agit de façon autonome dans cet environnement, et en faisant ainsi il réalise un ensemble de buts ou de tâches pour lesquels il a été conçu [Brustoloni, 1991; Castelfranchi, 1995; Jennings et al., 1998; Meas, 1990; Wooldridge, 2000].

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques [Michalski et al., 1993; Mitchell, 1997]. L'apprentissage automatique permet, entre autres, à un système basé Agent d'opérer dans un environnement inconnu, ou partiellement connu, au départ et d'apprendre à partir de son expérience à s'y adapter pour opérer de manière appropriée. Par conséquent l'apprentissage automatique permet à un Agent de devenir plus compétent et d'améliorer sa performance.

2. Problématique

L'apprentissage automatique dans un Agent! Pour aborder ce thème de recherche, qui regroupe deux axes intégrant chacun de multiples disciplines, initialement, nous nous sommes penchés sur la recherche bibliographique et les lectures des travaux autour de ce thème. A l'issue d'une première synthèse, nous avons abouti aux constatations suivantes, qui ont impliqué un certain nombre d'interrogations et de préoccupations:

- D'une part, la technologie Agent est un domaine de recherche récent qui n'est pas à sa maturité; en effet ce domaine est en pleine effervescence et il continue toujours à évoluer. Vu ses multiples apports, cette technologie suscite l'attention des concepteurs pour le développement d'applications à base d'Agents. En plus des universités et des laboratoires de recherches, la technologie Agent est largement adoptée par les développeurs d'applications pour résoudre des problèmes réels dans des domaines variés et multiples. L'intérêt, pour la technologie Agent, a été aussi favorisé par la croissance de

l'Internet; la prolifération de ce réseau et son utilisation massive ont donné naissance à de nouvelles applications d'Agents. Ainsi, pour mener notre travail de recherche, il est primordial de répondre aux interrogations suivantes:

1. Est-ce que tout Agent doit faire recours à l'apprentissage automatique? La réponse à cette question est évidente, mais elle engendre la suivante:
 2. Quels sont les types d'Agents les plus concernés par l'apprentissage automatique?
- D'autre part, l'apprentissage automatique est pluridisciplinaire et ouvre lui-même plusieurs axes et perspectives de recherche. La question, pour nous, est donc:
 1. Selon quelle perspective faut-il aborder ce domaine? Une fois la perspective de recherche fixée, la préoccupation suivante s'impose.
 2. Pour chaque type d'Agents quelles sont les méthodes de l'apprentissage automatique les plus appropriées?

Pour répondre à cette interrogation, nous nous sommes trouvés devant une large panoplie et une pléthore de méthodes d'apprentissage. En effet, l'apprentissage automatique se situe au carrefour de plusieurs disciplines, contribuant à sa richesse [Mitchell, 1997] et intègre une grande variété de concepts et de techniques datant des débuts de l'informatique [McCulloch et Pitts, 1943]. Ainsi, cette pléthore nous pose problème et rend difficile la réponse à notre préoccupation.

Les constats finaux des études bibliographiques menées, nous ont révélé l'existence de plusieurs travaux théoriques et pratiques concernant l'utilisation de l'apprentissage automatique dans les systèmes multi-agents, pour la résolution de plusieurs aspects résultant de la tendance sociale. Alors que, l'intégration de l'apprentissage automatique dans un Agent à tendance individuelle reste généralement pratique; dans le sens où il existe un nombre important de systèmes basés Agent avec des capacités d'apprentissage automatique développés pour des domaines spécifiques. Ces travaux partent d'applications précises, de systèmes basés Agent, et optent pour des méthodes d'apprentissage trouvées adéquates pour leur domaines d'applications. D'où la nécessité de travaux de base, concernant l'apprentissage automatique dans un Agent, qui soient indépendants d'applications précises.

3. Objectifs

Pour répondre à la problématique posée, nous nous sommes fixés un certain nombre d'objectifs visant à apporter un fondement et un support de base, une assise conceptuelle, un modèle général et des expérimentations réelles à l'apprentissage automatique dans un Agent.

Ainsi, cette thèse est chargée, en premier lieu, de la détermination des types d'Agents nécessitant le recours à l'apprentissage automatique. Cette tâche est primordiale pour le premier objectif du fondement et du support de base, qui correspondant à l'établissement d'une mise en correspondance entre les types d'Agents-Apprenants et les méthodes d'apprentissage automatique.

- Donc, d'une part, nous avons à définir une classification d'Agents avec un focus sur l'apprentissage automatique; afin de pouvoir déterminer les différents types d'Agents-Apprenants.
- D'autre part, avec la pléthore de méthodes d'apprentissage, nous avons à les catégoriser pour disposer d'approches de l'apprentissage automatique; puis étudier les méthodes les plus représentatives et les plus connues pour chacune des approches dégagées.

Le second objectif, de cette thèse, est l'établissement d'un modèle général et générique pour les systèmes basés Agent-Apprenant. Une assise conceptuelle permettant de supporter ce modèle et renforcer ses propriétés de généralité et de genericité est nécessaire. Ce modèle doit être indépendant de tout domaine et de toute méthode d'apprentissage mais il doit permettre des instanciations par différentes applications.

La réutilisation et l'exploitation du modèle, que nous proposons, est une finalité en soit. Ainsi, des expérimentations doivent projeter le modèle sur des domaines d'applications afin de l'instancier. Ces expérimentations, constituant aussi des validations du modèle, doivent mettre en œuvre ses différents composants ainsi que les éléments de sa base conceptuelle.

4. Organisation de la Thèse

L'organisation de ce mémoire de thèse reflète la démarche que nous avons adoptée lors de la réalisation de notre travail; ainsi, il est composé des cinq chapitres suivants:

Chapitre 1: La Technologie Agent et L'Apprentissage Automatique

Le premier chapitre présente la technologie Agent avec ses origines, ses concepts de base, ses différentes approches, architectures et applications. Il introduit l'apprentissage automatique et décrit ses apports pour la résolution de

problèmes complexes. Nous analysons, dans ce chapitre les travaux existants et nous présentons les principaux objectifs de notre travail.

Chapitre 2: Approches et Méthodes de l'Apprentissage Automatique

Vu la pléthore des méthodes d'apprentissage automatique, au chapitre 2 nous procédons à une catégorisation de ces méthodes pour disposer d'approches d'apprentissage globales. Ensuite étudier et analyser les approches dégagées ainsi que les méthodes les plus représentatives pour chacune de ces approches.

Chapitre 3 : Mise en Correspondance entre Agents Apprenants et Approches/Méthodes d'Apprentissage

Le chapitre 3 présente les différents types d'Agents-Apprenants déterminés, avec au préalable la définition d'un ensemble de critères de classification. Puis, il décrit la mise en correspondance établie entre les types d'Agents-Apprenants et les approches ainsi que les méthodes d'apprentissage automatique.

Chapitre 4: Modèle Général et Générique de Systèmes basés-Agent Apprenant

Le chapitre 4 est réservé à la présentation du modèle général et générique, proposé pour les systèmes basés Agent-Apprenant. Nous décrivons, dans ce chapitre, la démarche de construction de ce modèle, puis l'ensemble des concepts formant l'assise conceptuelle du modèle proposé ainsi que ses différents composants.

Chapitre 5 : Instanciation du Modèle Général et Générique: Expérimentations et Etudes de Cas

Le chapitre 5 décrit des réutilisations du modèle proposé, en présentant deux expérimentations et études de cas réels. Ces dernières sont des instanciations du modèle, par des domaines d'applications, permettant aussi de valider le modèle proposé, par la mise en œuvre de ses différents composants ainsi que des éléments de sa base conceptuelle.

A la fin de cette thèse, nous émettons nos conclusions en précisant nos contributions ainsi que les perspectives envisageables pour notre travail de recherche.

Chapitre 1

La Technologie Agent et L'Apprentissage Automatique

1.1. Introduction

La technologie Agent, émergeant de la recherche en intelligence artificielle distribuée dans les années quatre-vingt, a suscité beaucoup d'attention ces dernières années; à cause de sa puissance et son grand potentiel. Elle est maintenant reconnue comme une technologie prometteuse pour la construction de systèmes autonomes, complexes et intelligents [Hakansson et al., 2010].

Nous nous intéressons, dans ce chapitre, à cette technologie, qui constitue un axe central pour notre travail intitulé: l'apprentissage automatique dans un Agent.

Le concept Agent a été influencé par une variété de disciplines et d'expériences. Cependant, les origines de ce terme proviennent de l'intelligence artificielle et plus particulièrement de l'intelligence artificielle distribuée. Nous commençons donc par les racines de la technologie Agent et les disciplines ayant contribué à son émergence, puis nous présenterons ses concepts de base, la notion d'Agent ainsi que ses propriétés prédominantes. Des approches d'Agents selon différentes visions seront aussi exposées pour aboutir ensuite à des architectures définissant la structure des Agents.

L'apprentissage automatique, comme sous-domaine de l'intelligence artificielle, permet une construction automatique des connaissances et de nos jours, il est particulièrement attrayant. Dans ce chapitre, cette discipline avec ses apports multiples est présentée comme une alternative à l'approche classique, pour le développement de systèmes informatiques complexes. L'avant dernière section, de ce chapitre, présente les résultats de l'étude de l'état de l'art concernant notre thème de recherche et l'analyse des travaux existants. Elle introduit aussi les axes stratégiques que nous avons tracés afin de mener notre travail de thèse et ses objectifs principaux.

1.2. Intelligence Artificielle

La technologie Agent et l'apprentissage automatique constituent les deux axes principaux de notre travail de recherche, ces deux disciplines sont fortement liées au domaine de l'Intelligence Artificielle (I.A). Cette dernière tente, depuis longtemps, non seulement de comprendre le fonctionnement des entités intelligentes, mais aussi d'en construire. Un objectif central de l'I.A est de créer des systèmes caractérisés d'intelligence artificielle, autrement dit des systèmes intelligents. Ainsi, depuis longtemps, les travaux en intelligence artificielle continuent à progresser pour réaliser cet objectif. En réalité, si on considère l'histoire de l'I.A, il est possible d'identifier trois grandes périodes [Alliot et Schiex, 1994; Igor, 1987; Kurzweil, 1990; Russell et Norvig, 2006]:

1. Les débuts des années quarante jusqu'à la fin des années cinquante, l'I.A naît en tant que science entourée d'espérances grandioses et de promesses impressionnantes.
2. Ensuite, le manque de résultats convaincants, finit par réorienter la recherche vers un discours plus réticent et moins éblouissant, cherchant des résultats concrets et des expérimentations réelles loin de l'optimisme démesuré.
3. Enfin, à partir des années quatre vingt et à la suite de ses propres progrès et aussi à cause des nouvelles découvertes provenant de ses frontières interdisciplinaires, l'I.A reprend ses ambitions et ses espoirs.

En effet, dès leurs premières idées sur l'I.A, les chercheurs ont tenté d'imiter l'intelligence humaine. A partir des années quarante, plusieurs scientifiques ont travaillé sur la modélisation du cerveau humain, en particulier: McCulloch et Walter Pitts, avec leurs travaux sur les neurones artificiels [McCulloch et Pitts, 1943]. Aussi, en 1950 une vision de l'intelligence artificielle a été énoncée par Alan Turing dans son article «les ordinateurs et l'intelligence» [Turing, 1950]. Et c'est à la conférence de Darmouth College [McCarthy et al., 1955] que la discipline nommée intelligence artificielle a vraiment pris naissance avec tant d'espérances et de promesses. Par la suite, les problèmes auxquels s'est confrontée l'I.A, surtout pour générer des comportements intelligents et gérer les explosions combinatoires pour certains domaines, ont réduit l'enthousiasme des débuts de

cette discipline. Malgré cette réticence, les travaux de l'I.A ont continué à creuser et ont repris des théories et des résultats expérimentaux d'autres disciplines. Ainsi, l'I.A est parvenue à prendre un nouvel élan; en effet le domaine de l'intelligence artificielle a vécu, ces dernières décennies, un développement ininterrompu et accéléré de ses recherches. Ces progrès considérables de l'I.A sont alliés d'une part, au surgissement de nouvelles conceptions théoriques, modèles et techniques; et d'autre part aux avancées technologiques en matière de vitesse d'exécution et de capacité mémoire. Après cet historique, on peut définir l'I.A, pour en déduire la définition des systèmes qu'elle se charge de construire.

1.2.1. Définitions de L'Intelligence Artificielle

Si nous voulons définir l'intelligence artificielle et par conséquent les systèmes qu'elle construit, nous dirons qu'il n'y a pas un consensus; mais plusieurs définitions coexistent, selon différentes visions.

Pour certains l'intelligence artificielle est la réalisation de programmes imitant dans leur fonctionnement le raisonnement humain. Ainsi Bellman définit l'intelligence artificielle comme étant « l'automatisation des activités que nous associons à la pensée humaine, des activités telles que: la prise de décision, la résolution de problèmes, l'apprentissage...» [Bellman, 1978].

Kurzweil définit l'intelligence artificielle comme étant: « l'art de créer des machines capables de prendre en charges des fonctions exigeant de l'intelligence quand elles sont réalisées par des gens» [Kurzweil, 1990]. Cette définition est proche de celle stipulant que: « L'I.A est la science de programmer les ordinateurs pour qu'ils réalisent des tâches qui nécessitent de l'intelligence lorsqu'elles sont réalisées par des êtres humains»

D'autres chercheurs, entre autres: [Charniak et McDermott, 1985; McCulloch et Pitts, 1943] ont tenté d'imiter le fonctionnement du cerveau humain et définissent l'intelligence artificielle par: « l'étude des facultés mentales à travers l'utilisation de modèles informatiques».

Vers la fin des années quatre vingt, l'I.A a pris un nouvel élan, d'où ses nouvelles définitions, entre autres: « l'intelligence artificielle est l'étude de la conception d'agents intelligents. » [Poole et al., 1998].

La définition la plus répandue affirme que l'intelligence artificielle commence là où l'informatique classique s'arrête; ainsi tout problème pour lequel il n'existe pas d'algorithme connu ou raisonnable permettant de le résoudre relève à priori de

L'I.A. A notre avis, ces définitions sont complémentaires et couvrent le maximum de domaines d'application de l'intelligence artificielle, entre autres:

- La Vision automatique et reconnaissances de formes.
- Le Traitement automatique du langage naturel.
- La Traduction automatique de documents.
- Le Raisonnement et la résolution de problèmes.
- La Représentation des connaissances.
- Les Interfaces intelligentes.
- L'Apprentissage automatique.
- La Fouille de données.
- Les Systèmes experts et Système à base de connaissances.
- Les Jeux (échecs, dames...).
- L'Ordonnancement, la satisfaction de contraintes.
- L'Optimisation de parcours ou de charge.
- La Robotique.

Nous terminons cette section par une définition énumérant les caractéristiques des systèmes intelligents, formulée par Alain Turing [Turing, 1950] qui considère que: « les principales composantes d'un système d'intelligence artificielle doivent être: les connaissances, le raisonnement, la compréhension du langage naturel et l'apprentissage ».

L'importance, pour nous, de cette dernière caractéristique réside dans le fait que l'apprentissage est l'une des composantes principales d'un système intelligent. Cette caractéristique ainsi que l'historique et la naissance de l'intelligence artificielle reflètent la forte relation entre l'I.A et l'apprentissage automatique. En effet, les premiers travaux appartenant à l'intelligence artificielle, sur la modélisation du cerveau humain, sont en réalité du domaine de l'apprentissage automatique. Ce dernier, considéré comme une approche de l'intelligence artificielle [Michalski et al., 1993] connaît des recours intensifs, surtout avec l'arrivée de nouvelles technologies et de l'intelligence artificielle distribuée.

1.2.2. Intelligence Artificielle Distribuée

L'Intelligence Artificielle Distribuée (I.A.D) est née de la difficulté d'intégrer dans une même base de connaissances, l'expertise, les compétences et la connaissance de différentes entités qui communiquent et collaborent pour réaliser un but commun. L'I.A.D consiste à distribuer l'expertise au sein d'une société d'entités appelées Agents dont le contrôle et les données sont distribués [Ferber, 1995; Hewitt et Inman, 1991; Wooldrige et Jennings, 1995].

Avec les avancées technologiques en matière de réseaux, de traitements parallèles et d'ouverture des systèmes informatiques, l'intelligence artificielle distribuée est de plus en plus considérée. Cette discipline est une approche de l'intelligence artificielle qui est concernée par l'étude et la construction d'entités autonomes, nommées Agents pouvant interagir entre elles et avec l'environnement. Ces Agents, qui sont relativement indépendants et autonomes interagissent dans des modes simples ou complexes de coopération pour accomplir un objectif global, notamment la résolution de problèmes complexes [Demazeau et Müller, 1990; Durfee et al., 1989; Hewitt et Inman, 1991; Jennings et al., 1998; Sen et Weiss, 1999].

L'I.A.D est donc un domaine concerné par les systèmes ouverts et distribués dont les entités présentent une sorte d'intelligence et qui essayent d'accomplir des buts qui peuvent être implicites ou explicites. L'I.A.D [Hewitt et Inman, 1991] est la branche de l'intelligence artificielle qui s'intéresse aux systèmes distribués plutôt que mono-agent et se sous divise en deux sous disciplines:

1. Résolution distribuée de problèmes [Durfee et al., 1989].
2. Systèmes multi-agents [Demazeau et Müller, 1990; Ferber, 1995].

Le premier domaine met l'accent sur la résolution d'un problème spécifique en le distribuant sur un ensemble d'autres modules qui coopèrent en échangeant des données et des solutions, tandis que le deuxième s'intéresse à la synthèse d'un comportement intelligent à partir d'un ensemble d'Agents indépendants qui interagissent dans une société commune, dite système multi-agents. Les thèmes de l'intelligence artificielle distribuée sont nombreux, ils incluent: la description des tâches, leur décomposition, leur distribution et allocation; l'interaction et l'organisation, la coopération et la compétition entre Agents.

La forte relation entre l'apprentissage automatique et l'intelligence artificielle, a été établie depuis longtemps et elle s'est renforcée de plus en plus avec l'avènement de l'intelligence artificielle distribuée. C'est ce que stipule la citation de Gérard Weiss suivante:

«L'intersection de l'intelligence artificielle distribuée et de l'apprentissage automatique constitue un domaine de recherche relativement jeune mais important qui suscite une attention croissante ces dernières années. La raison de cette attention est en grande partie due au fait que la complexité, des systèmes étudiés en intelligence artificielle distribuée, fait que la spécification correcte et complète de leur répertoire comportemental et leur dynamique est souvent extrêmement difficile ou même impossible. Il est par conséquent

largement convenu que ces systèmes devraient être équipés de la capacité d'apprendre, pour auto-améliorer leur performance future »¹ [Weiss, 1997].

Avec l'émergence de la nouvelle intelligence artificielle, visant à résoudre des problèmes dans un monde réel et dynamique, la notion d'Agent a eu une double évolution: l'Agent autonome caractérisé par son individualité et l'Agent faisant partie d'un système multi-agents. La première notion, qui recouvre les notions d'Agent autonome, est rencontrée surtout dans les disciplines de l'intelligence artificielle et de la robotique [Brustoloni, 1991; Castelfranchi, 1995; Jennings et al., 1998; Meas, 1990; Wooldridge, 2000], tandis que la deuxième, Agent social, est rencontrée dans l'intelligence artificielle distribuée et la vie artificielle [Cohen et Levesque, 1988; Demazeau et Müller, 1990; Durfee et al., 1989; Ferber, 1995] et vise à exploiter la complexité émergente des systèmes composés d'entités relativement simples en interaction.

Notre travail de thèse concerne l'apprentissage automatique dans un Agent individuel; ainsi dans la section suivante, nous nous intéressons à la technologie Agent pour aborder ensuite l'apprentissage automatique et les systèmes basés Agent utilisant les techniques de cette dernière discipline.

1.3. La Technologie Agent

Pour commencer cette section, nous faisons référence au rapport du troisième symposium «Agent and Multi-Agent Systems: Technologies and Applications» paru dans le journal: «KES International Journal of Knowledge-based and Intelligent Engineering Systems» affirmant que:

« Les systèmes d'Agent et Multi-Agents sont un type innovateur des systèmes logiciels modernes et ont été longtemps reconnus comme une technologie prometteuse pour la construction de systèmes autonomes, complexes et intelligents » [Hakansson et al., 2010]².

¹ *“The intersection of distributed artificial intelligence and machine learning constitutes a relatively young but important area of research that has received steadily increasing attention in the past years. The reason for this attention is largely based on the insight that the complexity of the systems studied in distributed artificial intelligence often makes it extremely difficult or even impossible to correctly and completely specify their behavioral repertoires and their dynamics. It is therefore broadly agreed that these systems should be equipped with the ability to learn, that is, to improve their future performance on their own” [Weiss, 1997].*

² *“Agent and Multi-Agent systems are an innovative type of modern software systems and have long been recognized as a promising technology for constructing autonomous, complex and intelligent systems” [Hakansson et al., 2010].*

Il est clair que ce rapport met en évidence l'importance de la technologie Agent, en effet elle offre un répertoire puissant de concepts, de paradigmes, d'outils et de techniques qui ont le potentiel d'améliorer considérablement le développement des systèmes informatiques. Avant de présenter l'essentiel de ce répertoire, nous revenons aux origines de cette technologie et son émergence, par l'historique suivant.

1.3.1. Historique

La notion d'Agent est utilisée dans différentes disciplines: la sociologie, la biologie, la psychologie, les systèmes à base de connaissances, la robotique et d'autres domaines de l'informatique. D'après Jennings, Sycara et Wooldrige [Jennings et al., 1998], des chercheurs très actifs dans ce domaine, les Agents autonomes sont issus de plusieurs disciplines dont les principales sont:

- L'intelligence artificielle.
- La programmation orientée-objet et les systèmes basés-objets concurrents.
- Les interfaces homme-machine avancées.

En intelligence artificielle, plusieurs auteurs ont plus ou moins contribué à l'émergence de la technologie Agent, entre autre les travaux sur la planification en I.A de l'équipe de Allen [Allen et al., 1990], de Nilsson et ses collaborateurs [Chapman, 1987; Fikes et Nilsson, 1971; Wilkins, 1988]. Mais ce sont les travaux de:

- Brooks sur le comportement intelligent et son émergence [Brooks, 1986; Brooks, 1991a; Brooks 1991b];
- et ceux de Bratman sur les états mentaux et les modèles BDI (Beliefs, Desire et Intentions) [Bratman, 1987; Bratman, 1988; Bratman et al., 1992], qui ont considérablement contribué à la naissance de cette nouvelle technologie.

Brooks s'est intéressé à l'émergence de comportement intelligent à partir de l'interaction de plusieurs comportements simples. Il a développé une architecture réactive qui est fortement liée aux Agents Réactifs que nous présenterons par la suite. Les travaux de Bratman ont concerné les modèles mentaux et ont donné naissance à l'architecture la plus référencée pour les Agents dits cognitifs qui seront présentés dans une section suivante.

La programmation orientée-objet et les systèmes basés-objets concurrents ont aussi contribué à la technologie Agent, en effet le concept Agent peut être considéré comme une extension des concepts objet et acteur. Ainsi les travaux de plusieurs chercheurs dans ce domaine, particulièrement ceux de Booch [Booch,

1994; Masini et al., 1989] et ceux de Agha [Agha, 1986; Agha et Hewitt, 1988] ont participé, même indirectement, aux bases de la technologie Agent.

Les travaux sur les interfaces homme-machine avancées avaient pour but de faire évoluer les interfaces du paradigme dit «direct manipulation» vers des interfaces capables d'initiative. Cette tendance voulait avoir des systèmes informatiques, qui dans certaines circonstances peuvent agir plutôt que d'attendre l'utilisateur, qui leur demande exactement ce qu'il faut faire. Et c'est particulièrement les travaux de Patties Maes et son équipe ainsi que ceux de Nicolas Negroponte [Maes et Schneiderman, 1994; Negroponte, 1995] qui ont convergé vers les Agents assistants.

Maintenant que les origines, de la technologie Agent, sont connues, reste à connaître l'élément sur lequel repose cette technologie.

1.3.2. Qu'est ce qu'un Agent ?

Etant donné les origines diverses du concept Agent, nous ne pouvons pas lui attribuer une seule signification pour répondre à la question posée ci-dessus. Plusieurs définitions ont été proposées par différents auteurs pour clarifier ce concept et jusqu'à présent, il n'existe pas une définition universelle d'un Agent. A ce propos, Carl Hewitt [Hewitt et Inman, 1991] stipule que : « la question qu'est-ce qu'un Agent? est aussi embarrassante pour la communauté informatique que la question qu'est ce que l'intelligence? pour la communauté d'intelligence artificielle ». Alors plusieurs définitions, du concept Agent, coexistent et nous commençons par présenter celle d'un dictionnaire de la langue française.

Le dictionnaire Robert [Rey et al., 2007] qualifie un Agent de: « *Personne chargée des affaires et des intérêts d'un individu, d'un groupe ou d'un pays, pour le compte desquels elle agit* ». De cette définition générale, le terme «agit» du verbe « agir » est défini par: « *faire quelque chose, s'occuper, produire un effet* ».

Ainsi nous retenons deux aspects clés:

- un Agent accomplit quelque chose.
- un Agent agit à la demande de quelqu'un (Agent ou utilisateur).

Mais, pour nous, informaticiens, que signifie exactement le terme «Agent»? En réalité avec l'avènement de la technologie Agent et son immense potentiel, plusieurs auteurs, concepteurs et développeurs se sont intéressés à cette technologie. D'où l'existence, au départ, d'une multiplicité de définitions, ainsi nous présentons quelques unes de ces définitions, afin de cerner la notion d'Agent.

- Un Agent est une entité qui fonctionne continuellement et de manière autonome dans un environnement où d'autres processus se déroulent et d'autres agents existent. L'Agent est doté d'un état mental, qui représente ses connaissances, croyances, intentions et engagements vis-à-vis de lui-même et des autres agents [Cohen et Levesque, 1988; Shoham, 1993].
- Un Agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs (buts) pour lesquels il a été conçu [Wooldrige et Jennings, 1995, Jennings et al., 1998].
- Un Agent est une entité qui peut être considérée comme percevant son environnement grâce à des capteurs et qui agit sur cet environnement via des effecteurs [Russell et Norvig, 2006].
- Un Agent autonome est un système capable d'actions autonomes selon un objectif dans un environnement réel [Brustoloni, 1991].
- Un Agent autonome est un système informatique situé dans un environnement dynamique et complexe. Il perçoit et agit de façon autonome dans cet environnement, et en faisant ainsi il réalise un ensemble de buts ou de tâches pour lesquels il a été conçu [Maes, 1990].
- Le terme Agent représente une entité intelligente qui agit de manière rationnelle et intentionnelle en fonction de ses buts et de l'état courant de ses connaissances [Demazeau et Müller, 1991].
- Les Agents intelligents exécutent sans interruption trois fonctions : perception des conditions dynamiques dans l'environnement; action pour affecter des conditions dans l'environnement; et le raisonnement pour interpréter des perceptions, résoudre des problèmes, mener des inférences et déterminer des actions [Hayes-Roth, 1995]

Les points communs entre ces définitions peuvent être schématisés par la figure suivante qui illustre les aspects fondamentaux d'un Agent.

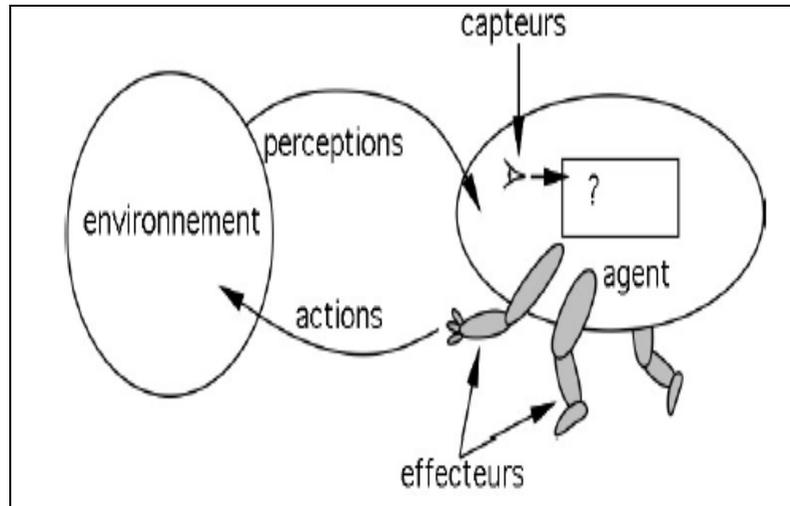


Figure 1.1 : Schématisation d'un Agent.

Pour conclure cette section, nous présentons une définition de Jaque Ferber [Ferber, 1995] qui synthétise, à notre avis, plusieurs aspects d'un Agent.

- Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et de ses interactions avec les autres agents».

Autrement Ferber [Ferber, 1995; Ferber, 1997] définit un Agent, par une entité physique ou virtuelle possédant les capacités et les compétences suivantes:

1. qui est capable d'agir dans un environnement;
2. qui peut communiquer directement avec d'autres agents;
3. qui est mue par un ensemble de tendances (sous forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser);
4. qui possède des ressources propres;
5. qui est capable de percevoir (mais de manière limitée) son environnement;
6. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune);
7. qui possède des compétences et offre des services;
8. qui peut éventuellement se reproduire.

Et dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

A notre avis les définitions présentées ci-dessus sont complémentaires et chacune d'elles apporte un plus au concept Agent. Les auteurs de ces définitions ont focalisé leur attention sur les aspects qui leurs semblent les plus importants, et la réunion des différents aspects montre la puissance du concept Agent.

1.3.3. Caractéristiques Essentielles d'un Agent

Dans la littérature, il existe plusieurs caractéristiques des Agents, en particulier pour ceux nommés Agents intelligents. Notre objectif ici est de présenter les propriétés essentielles du concept Agent indépendamment de son type. Nous constatons que les définitions présentées précédemment [Brustoloni, 1991; Cohen et Levesque, 1988; Demazeau et Müller, 1999; Ferber, 1995; Hayes-Roth, 1995; Jennings et al., 1998; Maes, 1990; Russell et Norvig, 2006; Shoham, 1993; Wooldrige et Jennings, 1995] énoncent un certain nombre de propriétés et caractérisent essentiellement un Agent par:

L'Existence dans un environnement: l'Agent est situé dans un espace où il va agir; cet espace peut être physique ou virtuel. Et ainsi l'environnement d'un Agent est un espace physique, ou un espace virtuel tels que l'internet, l'utilisateur ou d'autres Agents. L'Agent doit être capable de percevoir son environnement et d'agir en conséquence.

L'Autonomie: qui est la caractéristique principale du concept Agent, en effet l'ensemble des définitions ont mis l'accent sur cette propriété. L'autonomie signifie que l'Agent doit pouvoir prendre des initiatives et agir sans intervention de l'utilisateur final ou des autres Agents, donc c'est la capacité de l'Agent à exécuter des actions de sa propre initiative. Autrement dit elle caractérise l'Agent par une indépendance de décision pour satisfaire ses objectifs.

L'autonomie est la caractéristique principale qui distingue le concept Agent du concept objet. En réalité, le paradigme Agent a été influencé par le paradigme objet, comme nous l'avons déjà évoqué dans la section historique de ce chapitre. Mais l'autonomie permet de bien délimiter les frontières entre les deux concepts. En effet tout comme les Agents, les objets encapsulent leurs états internes, c'est-à-dire leurs données. Ils peuvent également poser des actions sur ces états par le biais de leurs méthodes et ils communiquent en s'envoyant des messages. Mais les objets diffèrent des Agents par leur degré d'autonomie. Un objet n'a pas de contrôle sur le lancement de ses propres actions; en effet une méthode doit être invoquée par un autre objet pour pouvoir accomplir ses effets. Alors qu'un Agent, à la réception d'une requête, peut décider d'exécuter l'action demandée ou la refuser. La notion d'invocation diffère de la notion de requête, pour cette dernière il s'agit de demander et d'attendre l'acceptation ou le refus. Par contre l'invocation implique une exécution de la demande.

L'Interactivité: L'Agent existe dans un environnement et ainsi il doit pouvoir interagir avec cet environnement. La figure 1.1 met en évidence la caractéristique d'interactivité d'un Agent par les deux flèches en double sens entre l'Agent et son environnement, représentant les perceptions et les actions qui en résultent. Pour l'environnement, l'Agent peut interagir avec: des interfaces vers des humains ou des capteurs, d'autres Agents ou des logiciels qui ne sont pas des Agents.

La Rationalité ou L'intelligence: un Agent peut avoir un comportement rationnel et intelligent qui lui permet d'atteindre les objectifs pour lesquels il a été conçu, tout en prenant en considération les changements de l'environnement. Cette caractéristique qui peut, ne pas exister chez un Agent, nécessite des connaissances et des mécanismes de raisonnement.

La communication : Un Agent est capable de communiquer avec son environnement, afin d'échanger des informations pouvant l'aider dans l'accomplissement de ses objectifs et peut aussi aider les autres Agents.

C'est la communication entre Agents qui fait qu'un groupe d'Agents forment un système multi-Agents que nous allons présenter brièvement. Nous rappelons ici que notre travail s'intéresse à la technologie Agent avec la tendance individuelle d'un Agent, nommé aussi par certain chercheur Agent autonome, système basé-Agent ou système mono-agent [Brustoloni, 1991; Castelfranchi, 1995; Jennings et al., 1998; Meas, 1990; Wooldridge, 2000].

1.3.4. Systèmes Multi-Agents

Un Système Multi-Agents (souvent référencé par un SMA) est défini comme un ensemble d'Agents, ayant un ou plusieurs buts à accomplir, et qu'ils œuvrent à atteindre en exécutant concurremment une ou plusieurs tâches [Demazeau et Müller, 1990; Ferber, 1995; Hewitt et Inman 1991; Jennings et al., 1998 ; Wooldrige et Jennings, 1995] .

Les SMA ne sont pas qu'un simple groupe d'Agents dans un environnement commun, mais une réelle organisation avec des règles sociales et des interactions permettant la coopération pour la résolution de problèmes que les systèmes centralisés ne pourraient pas résoudre. Ainsi les SMA sont caractérisés [Jennings et al., 1998] par:

- Chaque Agent possède des informations ou des capacités de résolution de problèmes limitées.
- Pas de contrôle global d'un SMA.
- Les données sont décentralisées.
- Le calcul est asynchrone.

Une description des systèmes multi-Agents, proposée par Ferber [Ferber, 1995], énumère les éléments suivants pour la composition d'un SMA:

- Un environnement.
- Un ensemble d'objets situés.
- Un ensemble d'Agents représentant des entités actives du système.
- Un ensemble de relations qui unissent des objets entre eux.
- Un ensemble d'opérations permettant aux Agents de percevoir, produire, consommer, transformer et manipuler des objets.
- Des opérateurs chargés de représenter l'application de ces opérations.

Les SMA utilisent les ACL (Agent Communication Language) pour la communication inter-Agent. Les principaux ACL utilisés sont: KQML (Knowledge Query and Manipulation Language) [Finin et al., 1992] et FIPA-ACL (FIPA Agent Communication Language) [FIPA, 2006]. Les deux langages sont fondés sur la théorie des actes de langage [Vanderveken, 1990] et ont été conçus pour l'échange d'information, de connaissances ou encore de services entre Agents.

Les apports des SMA sont nombreux, les plus importants sont la modélisation et la simulation de systèmes complexes avec un certain degré de fiabilité et d'efficacité, aussi leur capacité à faire inter-opérer des systèmes hétérogènes dits «legacy systems» ainsi que leur habilité à résoudre des problèmes où les données, l'expertise et le contrôle sont distribués.

Après ce bref aperçu sur les systèmes multi-Agents, nous allons revenir au centre d'intérêt de ce travail de thèse, les Agents autonomes pour présenter leurs différentes approches ainsi que les architectures qui leur sont associées afin de mieux comprendre leurs fonctionnements et structures internes. Nous présentons ensuite les principaux domaines d'application de la technologie Agent.

1.3.5. Approches d'Agents

Dans la littérature, il est généralement considéré qu'il existe deux grandes approches ou visions d'Agents [Ferber, 1995; Jennings et al., 1998; Meas, 1990; Wilkins, 1988; Wooldrige et Jennings, 1995]. Le rapprochement des deux approches de base a donné naissance à une troisième approche, et ainsi nous avons:

1. L'Approche Réactive.
2. L'Approche Cognitive.
3. L'Approche Hybride.

1.3.5.1. Approche Réactive

L'approche réactive, qui est défendue par plusieurs chercheurs et auteurs appartenant à l'école réactive [Allen et al., 1990; Brooks, 1986; Brooks, 1991a; Brooks 1991b; Chapman, 1987], s'intéresse aux Agents dit réactifs. Ces Agents ne possèdent aucune représentation de l'environnement et dont le comportement est décrit par des boucles sensorimotrices simples basées sur des mécanismes associant les actions directement aux perceptions.

Les Agents réactifs sont souvent qualifiés de ne pas être "intelligents" par eux-mêmes, mais qu'un comportement intelligent devrait émerger de l'interaction entre ces agents réactifs et l'environnement. Ils sont des composantes très simples qui perçoivent l'environnement et sont capables d'agir sur celui-ci [Brooks, 1986; Brooks, 1991a]. Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent pas de croyances.

1.3.5.2. Approche Cognitive

En intelligence artificielle, certains chercheurs [Bratman, 1987; Bratman, 1988; Bratman, 1992; Müller, 1996; Rao et Georgeff, 1995; Shoham et Cousins 1994] ont étudié le comportement rationnel permettant aux Agents d'atteindre leurs buts et se sont intéressés aux problèmes de planification et de raisonnement. Cette voie de recherche était à la base de l'approche cognitive et ainsi des Agents cognitifs.

Les Agents cognitifs dits aussi délibératifs et intelligents sont caractérisés principalement par un niveau symbolique de la représentation des connaissances et par des notions mentales. Les Agents cognitifs maintiennent une représentation interne de l'environnement, des buts explicites, ils sont capables de planifier leur comportement, mémoriser leurs actions passées et communiquer par envoi de messages [Ferber, 1997; Meas 1990; Wilkins, 1988; Wooldrige et Jennings, 1995]. Les Agents délibératifs ont la capacité de résoudre des problèmes complexes.

1.3.5.3. Approche Hybride

Les approches réactive et cognitive considérées séparément l'une de l'autre sont diamétralement opposées, mais elles peuvent être complémentaires par une hybridation. L'approche hybride combinant les deux tendances réactive et cognitive peut tirer profit des avantages de chacune d'elles. Cette troisième tendance a donné naissance aux Agents hybrides qui intègrent l'aspect cognitif et réactif [Müller, 1996]. Ainsi un Agent hybride comporte une partie cognitive avec une représentation symbolique du monde et une partie réactive. La composante délibérative, de l'Agent hybride, est dotée des techniques de raisonnement et de prise de décision issues de l'intelligence artificielle; et

s'occupe ainsi de la résolution des problèmes complexes. Quand à la composante réactive, elle se charge de la réaction à l'environnement aux moments opportuns.

A chacune des ces trois approches correspond une ou plusieurs architectures dont nous allons présenter les plus connues.

1.3.6. Architectures des Agents

L'architecture d'un Agent décrit sa structure interne et son fonctionnement et à partir des approches précédentes, il est évident que l'architecture d'un Agent découle de son aspect réactif ou cognitif ou les deux ensembles. Ainsi, il existe trois types d'architectures:

1. Architectures Réactives.
2. Architectures Cognitives.
3. Architectures Hybrides.

1.3.6.1. Architectures Réactives

Comme, nous l'avons évoqué dans la section historique de la technologie Agent, l'idée d'architectures d'Agents réactifs a été essentiellement introduite par Rodney Brooks [Brooks, 1986; Brooks, 1991a; Brooks, 1991b] avec ses travaux sur le comportement intelligent et son émergence.

Les architectures réactives, dites aussi comportementales, représentent le fonctionnement de l'Agent au moyen de composantes avec une structure de contrôle simple, et sans représentation évoluée des connaissances de l'Agent. Le comportement intelligent de l'Agent résulte des interactions entre ces composantes et l'environnement. Un Agent avec une telle architecture possède la capacité de réagir rapidement à des problèmes simples, qui ne nécessitent pas un haut niveau de raisonnement.

L'architecture réactive la plus connue est celle proposée par Brooks [Brooks, 1986], appelée architecture de subsomption ou de hiérarchie de tâches. Cette architecture est composée de plusieurs modules, dit modules de compétence avec une organisation en couches hiérarchisées. Chaque module est responsable d'un comportement spécifique de l'Agent et permet la réalisation d'une tâche particulière. Les tâches éventuellement concurrentes sont affectées à priori par des priorités. La figure suivante illustre cette architecture.

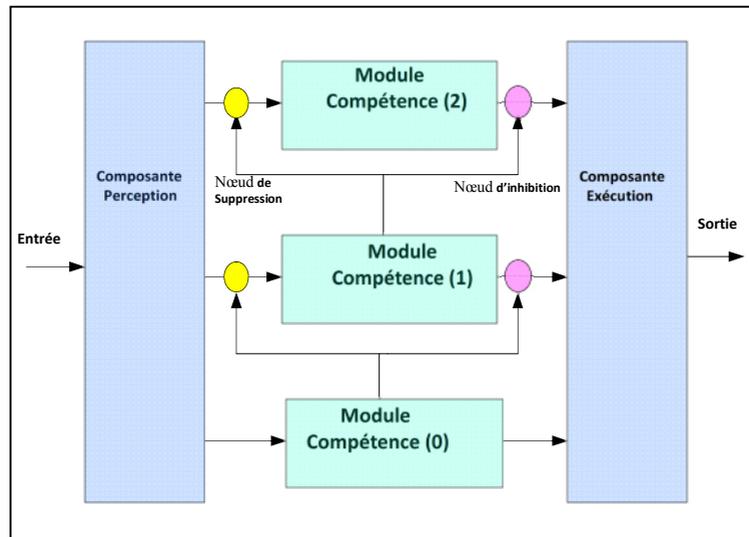


Figure 1.2: Architecture réactive de subsomption.

Les couches supérieures ont une priorité plus petite que les couches inférieures, car les modules de ces dernières sont responsables de tâches plus simples mais plus urgentes. Ainsi un module sur une couche inférieure peut modifier l'entrée d'un module supérieur ou inhiber son action grâce aux nœuds de suppression et d'inhibition, illustrés par la figure 1.2.

Le fonctionnement de l'Agent réactif est décrit par un ensemble de règles de comportement, de type situation-action et à chaque module de compétence est associé un sous-ensemble de règles de comportement spécifiques à ses compétences.

Les architectures réactives ont l'avantage de la simplicité et de l'efficacité du calcul, mais elles deviennent insuffisantes pour les applications où la prise de décision nécessite une modélisation de l'environnement [Jennings et al., 1998] et une prise en considération de l'historique et des expériences passées.

1.3.6.2. Architectures Cognitives

Les architectures cognitives permettent de représenter la structure et le fonctionnement des Agents cognitifs, et considèrent que ces derniers raisonnent à partir de connaissances décrivant explicitement leur environnement ainsi que les autres Agents.

L'architecture d'Agents délibératifs la plus importante est issue des travaux de Michael Bratman [Bratman, 1987; Bratman, 1988; Bratman et al., 1992] sur le raisonnement rationnel, elle est nommée: architecture BDI (Belief, Desire, Intention). Cette architecture décrit l'état interne d'un Agent cognitif en termes des trois attitudes mentales précédentes et définit le contrôle permettant à l'Agent

de sélectionner le cours d'action de ses attitudes. Dans des approches BDI plus étendues tel que «l'Intelligent Resource-bounded Machine Architecture: IRMA» [Müller, 1996], ces trois attitudes mentales ont été étendues, par les deux notions de buts et de plans. Avant de présenter l'architecture BDI, nous allons définir ses cinq notions de base selon les significations données par [Bratman, 1987; Müller, 1996; Rao et Georgeff, 1991; Shoham et Cousins, 1994; Wooldrige et Jennings, 1995]:

1. «Belief» ou croyances sont les informations décrivant l'état de l'environnement du point de vue d'un Agent, elles expriment ce que l'Agent croît sur l'état courant de son environnement. Les croyances peuvent changer par perception et interaction avec l'environnement.
2. «Desire» ou désirs représentent les préférences sur l'état futur de l'environnement d'un Agent. Une caractéristique importante des désirs est qu'un Agent peut avoir des désirs contradictoires, d'où la nécessité du choix d'un sous-ensemble consistant. Les désirs consistants sont parfois identifiés avec les buts de l'Agent.
3. Les buts représentent les engagements d'un Agent pour atteindre un ensemble d'états de l'environnement. Ainsi un désir est une étape dans le processus de création d'un but. La notion d'engagement d'atteindre un but décrit la transition des buts aux intentions.
4. Les intentions représentent les actions que l'Agent s'engage à exécuter. A cause des ressources limitées, un Agent ne peut poursuivre tous ses buts; ainsi il doit choisir un certain nombre de buts pour lesquels il s'engage. C'est ce processus qui est appelé la formation des intentions.
5. Les plans représentent les intentions que l'Agent s'engage à exécuter pour atteindre ses buts. Par conséquent, il est possible de structurer les intentions en plans plus étendus, et de définir les intentions courantes d'un Agent comme les plans partiels qui sont couramment adoptés et qui seront raffinés par la suite.

Basée sur les notions précédentes, l'architecture BDI est illustrée par la figure 1.3 suivante:

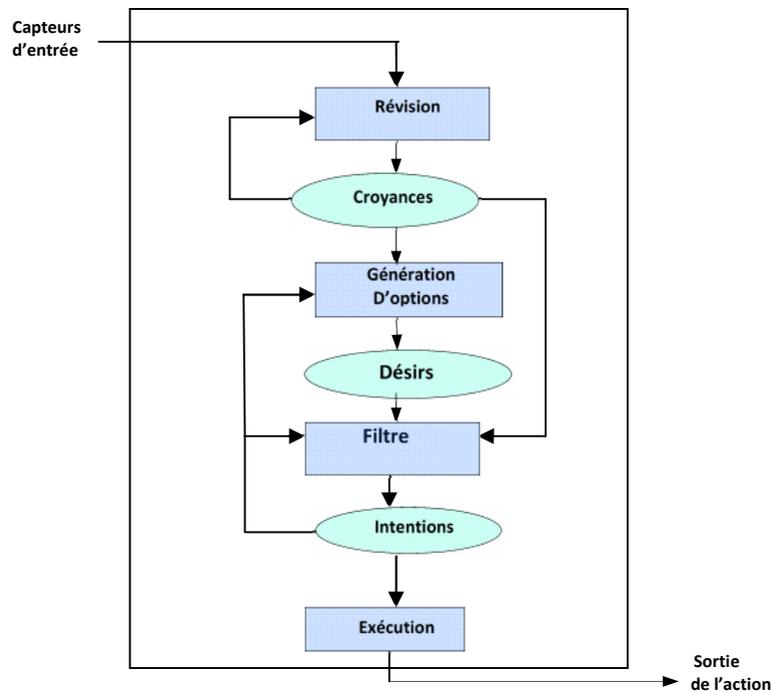


Figure 1.3: Architecture BDI d'un Agent.

Avec cette architecture, l'Agent possède une représentation explicite de ses croyances, désirs et intentions [Cohen et Levesque, 1990; Rao et Georgeff, 1991; Wooldrige et Jennings, 1995], illustrés dans la figure 1.3, par les formes ovales correspondant à des structures de connaissances ou de données.

Pour la résolution d'un problème, l'Agent doit produire des plans, par conséquent une séquence d'actions qu'il va exécuter. Les plans sont créés à partir des attitudes mentales : croyances, désirs et intentions, en se basant sur les composants: Révision, Génération d'options, le Filtre et Exécution. La fonction Révision, permet de réviser les croyances de l'Agent selon les entrées perçues. La fonction de Génération d'options détermine les options disponibles à l'Agent, donc ses désirs. Le Filtre est la partie de l'architecture qui a la responsabilité de bâtir des plans pour réaliser les intentions de l'Agent. Cette fonction représente le processus de délibération de l'Agent, d'où la nomination d'Agent délibératif. Le Filtre produit des plans partiels, dits intentions, puis les raffine en plans étendus en fonction des croyances, désirs et intentions, afin d'obtenir un plan exécutable qui est un sous ensemble des intentions. Les actions d'un plan étendu sont exécutées par le composant Exécution de l'architecture illustrée par la figure 1.3.

L'architecture BDI, reste la plus référencée pour les architectures cognitives. Elle permet de modéliser les connaissances de l'Agent et ses croyances sur l'environnement, ainsi que les changements qui peuvent affectés ce dernier. Mais une architecture hybride combinant les aspects réactifs et cognitifs des Agents est certainement plus souhaitable.

1.3.6.3. Architectures Hybrides

Les architectures hybrides sont une solution aux problèmes nécessitant des Agents hybrides, donc intégrant des comportements réactifs et cognitifs. L'architecture hybride la plus connue est « l'Integration of Reactive behavior and Rational Planning: InteRRaP » de Müller [Müller et al., 1995; Müller, 1996]. Cette dernière est la plus étudiée et la plus référencée. Généralement une architecture hybride est composée de deux ou plusieurs couches hiérarchiques verticales ou horizontales, qui interagissent entre elles afin d'atteindre un état cohérent de l'Agent avec ses deux aspects: réactif et cognitif.

L'architecture InteRRaP permettant l'intégration du comportement réactif et la planification rationnelle [Cohen et Levesque, 1990; Ferber, 1995; Müller, 1996; Wooldrige et Jennings, 1995] est composée de couches hiérarchiques verticales où les données d'entrée passent d'une couche à l'autre. Cette architecture est composée de trois couches de contrôle et trois bases de connaissances, tel qu'il est schématisé dans la figure 1.4 suivante.

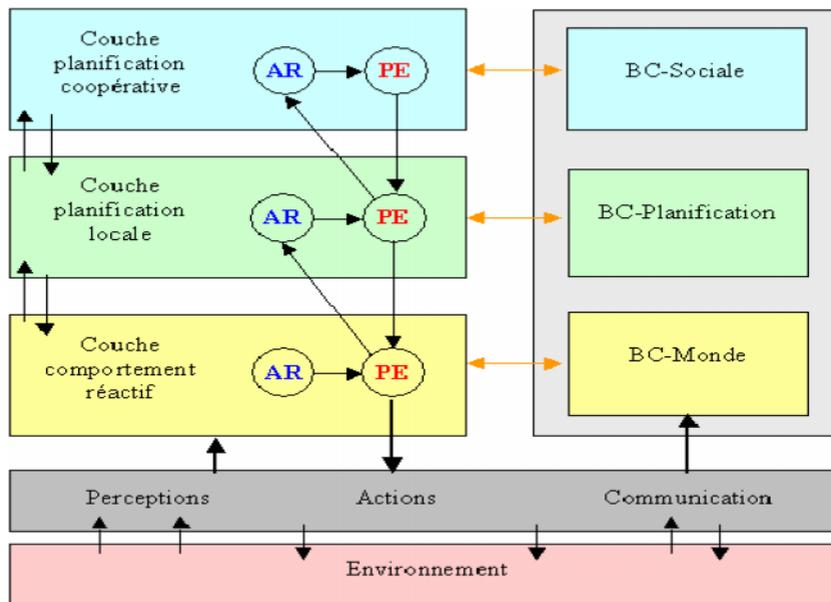


Figure 1.4: Architecture InteRRaP.

Les bases de connaissances: BC-Monde représente les croyances de l'Agent sur l'environnement, BC-Planification est une bibliothèque de plans et BC-Sociale représente les croyances de l'Agent sur les autres Agents du système et les possibilités de coopération. La couche au plus bas niveau est purement réactive et les deux autres cognitives s'occupant de la planification locale et coopérative. Chaque couche comprend deux modules: un module pour l'Activation des buts et la Reconnaissance des situations (AR) et un module de Planification et d'Exécution (PE). Les perceptions sur l'environnement sont transmises au module AR de la première couche et, de module en module, arrivent à la couche

supérieure. Pour le flux de contrôle des actions c'est le chemin inverse qui est entrepris pour aboutir à l'exécution des actions planifiées.

Les architectures hybrides sont ainsi une solution idéale pour un comportement hybride d'Agents, mais des fois elles sont difficiles à mettre en œuvre.

Après avoir présenté les notions de base de la technologie Agent, nous allons voir son étendue et ses domaines d'applications.

1.3.7. Applications de La Technologie Agent

La technologie Agent, relativement à d'autres paradigmes et techniques de l'informatique, a rapidement dépassé les frontières des universités et des laboratoires de recherches pour rejoindre les domaines industriels et commerciaux et résoudre des problèmes réels [Jennings et al., 1998; Jennings et Wooldridge, 1998; Maes, 1994; Parunak, 1987]. Ainsi les domaines d'applications de la technologie Agent sont tellement variés qu'il est difficile de les énumérer exhaustivement, néanmoins nous citons les principaux:

- *La robotique*: un domaine où l'application de la technologie Agent ne cesse de progresser avec les deux tendances de robot réactif et de robot cognitif [Meas, 1990].
- *Les applications de contrôle de processus*: elles étaient parmi les premières à profiter des apports de la technologie Agent, surtout que les contrôleurs de processus sont eux-mêmes des systèmes réactifs autonomes. Ces applications incluent entre autre: le contrôle de trafic aérien, routier et ferroviaire et les processus de contrôle nucléaire [Ingrand et al., 1992; Ljunberg et Lucas, 1992; Wang et Wang, 1997].
- *Les applications industrielles*: les développements ont concerné le contrôle et l'ordonnancement des fabrications dans les entreprises [Chung et Wu, 1997; Parunak, 1987].
- *Les réseaux de télécommunications*: ici, les applications ont porté sur l'optimisation des prestations de services et leur fiabilité [Griffeth et Velthuijsen, 1994].
- *Le commerce électronique*: ce domaine a connu beaucoup de systèmes à base d'Agents pour la prise en charge des transactions commerciales, entre autre: [Doorenbos et al., 1997; Etzioni, 1996].
- *Le domaine médicale*: des applications de surveillance médicale [Ferguson et al., 2009], de prise en charge des informations médicales (l'imagerie) et la télémédecine.

- *L'enseignement intelligent assisté par ordinateur*: des Agents intelligents basés sur les profils des apprenants ont été utilisés pour l'enseignement intelligent [Mitchell et Garuana, 1994].
- *La recherche et le filtrage d'information*: avec l'explosion combinatoire des informations sur le net, la prise en charge de la masse volumineuse d'informations est l'un des champs les plus actifs de l'application de la technologie Agent. En effet plusieurs systèmes ont été développés [Chen et Sycara, 1998; Lieberman, 1995].

Les applications de la technologie Agent sont nombreuses, elles montrent l'étendue de cette technologie et son grand potentiel. Certaines de ses applications nécessitent des capacités d'apprentissage automatique; entre autres: l'enseignement intelligent basé sur les profils des apprenants, la recherche d'informations personnalisée [Chen et Sycara, 1998; Mitchell et Garuana, 1994; Lang, 1995]. Un enseignement ou une recherche d'informations qui prennent en considération les centres d'intérêts de l'utilisateur, sont des applications qui nécessitent le recours à l'apprentissage automatique. Ce dernier constitue un axe central pour notre travail, la section suivante introduit cet axe et présente certains de ses apports pour le développement de systèmes informatiques complexes.

1.4. Apprentissage Automatique

Pour plusieurs équipes de recherches, l'intelligence artificielle est fortement corrélée à l'apprentissage automatique. Les chercheurs de l'université Carnegie Mellon, tels que : Jaime Carbonell, Yves Kodratoff, Ryszard Michalski et Tom Mitchell considèrent que pour définir ce que représente pour une machine le fait d'être intelligente, l'apprentissage est une question clé. En effet, d'après [Michalski et al., 1993]: *«Pour pouvoir être considérée comme intelligente, une machine doit posséder les caractéristiques d'apprentissage et de créativité»*. Ils appuient leur point de vue, auquel nous adhérons complètement, par le fait que: pour rendre une machine intelligente, il ne suffit pas d'accumuler une masse de connaissances dans celle-ci, mais c'est aussi la doter de capacités d'apprentissage à partir des événements observés. C'est aussi le pouvoir de la machine de profiter de son expérience, afin d'être mieux préparée à réagir, à l'avenir, à des événements similaires. Aussi, pour être intelligente, la machine doit être dotée de capacités de raffinement et de création de nouvelles connaissances, afin de s'adapter à de nouvelles situations. Ainsi pour avoir un système informatique intelligent, il doit posséder la capacité d'apprentissage automatique lui permettant de s'auto-améliorer.

La recherche en apprentissage automatique est ancienne, et depuis le succès des premiers travaux [Buchanan et al., 1972; Gold, 1965; Lenat, 1977; McCulloch et Pitts, 1943; Michalski et Chilausky, 1980; Mitchell, 1977; Mitchell et al., 1983; Samuel, 1959; Winston, 1975], ce domaine ne cesse de se développer. Les progrès de la recherche en apprentissage automatique influencent énormément le développement de l'intelligence artificielle et touchent plusieurs de ses sous-domaines. Ainsi il existe une forte relation entre l'apprentissage automatique et l'intelligence artificielle. Et comme nous l'avons évoqué, au début de ce chapitre, cette relation est de plus en plus renforcée avec l'avènement de l'intelligence artificielle distribuée et de la technologie Agent. Semblablement à l'intelligence artificielle, l'apprentissage automatique tente de comprendre et d'imiter l'apprentissage humain, en particulier, et naturel en général. La compréhension du fonctionnement du cerveau humain était parmi les premières inspirations de la recherche en apprentissage automatique qui continue toujours à s'inspirer de l'apprentissage des entités vivantes, donc de l'apprentissage naturel.

1.4.1. Apprentissage Naturel

L'apprentissage naturel, surtout humain, est à l'apprentissage automatique ce que l'intelligence humaine est à l'intelligence artificielle, essentiellement une source de modèles; d'où son importance. Pour un être humain, la réalisation d'une tâche est inséparablement liée à un processus d'apprentissage [Michalski et al., 1993; Nilsson, 1997]; en effet les humains, en cours de la réalisation d'une activité quelconque, s'efforcent généralement d'améliorer la manière dont ils l'accomplissent. Ainsi les humains profitent de leurs expériences, en apprenant de nouveaux procédés et modèles leur permettant de prendre en charge efficacement les tâches usuelles ou mêmes nouvelles. Heureusement, chaque être humain arrive au monde avec une grande capacité d'apprentissage, sinon l'humanité ne serait pas au point auquel elle est actuellement, sur tous les plans positifs de son développement (et malheureusement ceux qui sont négatifs aussi). L'apprentissage et l'évolution sont les deux propriétés de bases des créatures sur terre. En plus des humains, les colonies de fourmis et leur découverte de chemin minimal, pendant la recherche de la nourriture; ainsi que les termites et leurs constructions en tunnels complexes sont des exemples de systèmes biologiques capables d'apprentissage [Holland, 1975; Hopfield, 1982; Iredi et al., 2001]. Ainsi l'apprentissage naturel est une immense source d'inspiration pour l'apprentissage automatique.

Pour la science, l'apprentissage constitue un point commun à toutes les disciplines; en effet il est présent dans tout processus scientifique par la création itérative de modèles ou de théories [Mjolsness et Decoste, 2001], selon le cycle:

- Observations.

- Création d'hypothèses.
- Test théorique ou expérimental.

Ce cycle est un processus scientifique partagé par toutes les disciplines. Pour cela, tout développement dans chacune des phases de ce cycle, apporte des contributions à l'apprentissage. Ce dernier permet donc la création itérative de modèles à partir d'observations d'un environnement, pour une interaction avec ce dernier, plus efficace. Autrement dit, l'apprentissage peut être défini comme une modification du comportement de l'apprenant résultant d'une interaction répétée avec l'environnement.

1.4.2. Apprentissage Automatique: Quelques Définitions

L'apprentissage automatique ou artificiel regroupe tellement de concepts différents et variés qu'il est difficile d'en donner une définition unique. En effet l'apprentissage automatique est pluridisciplinaire et fait intervenir des concepts et des techniques de plusieurs domaines dont il tire ses inspirations et ses outils. L'apprentissage automatique représente un champ de recherche et d'application des plus foisonnants, d'après Tom Mitchell [Mitchell, 1997], il se situe au carrefour de plusieurs disciplines incluant: l'intelligence artificielle, l'analyse des données et les statistiques, la philosophie, la psychologie, la théorie de l'information, la biologie, les sciences cognitives, la théorie de la complexité... Nous avons schématisé ce carrefour par la figure suivante.

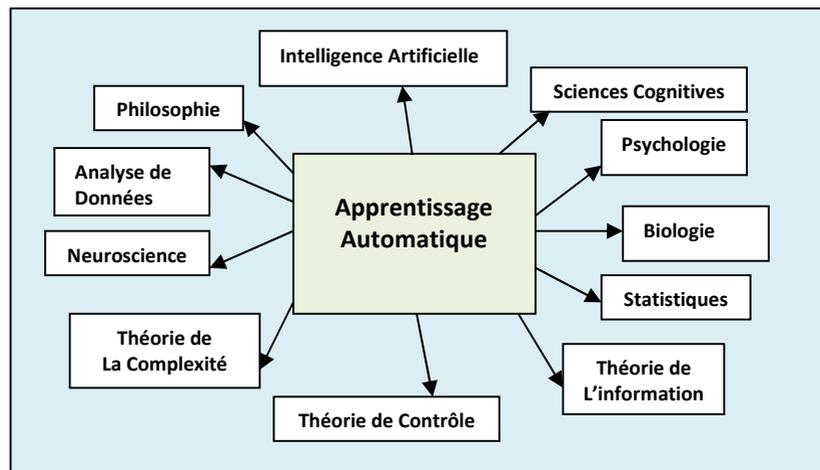


Figure 1.5: Aspect pluridisciplinaire de l'apprentissage automatique.

Alors pour ce domaine pluridisciplinaire, nous présentons les définitions suivantes:

- *Définition 1:* Du point de vue des machines, on dira qu'une machine apprend dès lors qu'elle change sa structure, son programme ou ses données en

fonction de données en entrée ou de réponses à son environnement de sorte à ce que ses performances futures deviennent meilleures [Michalski et Chilausky, 1980; Nilsson, 1997; Winston, 1975].

- *Définition 2:* L'apprentissage dénote des changements dans un système qui lui permettent de faire la même tâche plus efficacement la prochaine fois [Simon, 1983].
- *Définition 3:* L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques [Michalski et al., 1993].
- *Définition 4:* Un programme informatique apprend à partir de l'expérience E par rapport à une classe de tâches T et une mesure de performance P, si sa performance à l'exécution de tâches de T, mesurée par P, s'améliore avec l'expérience E [Mitchell, 1997; Mitchell, 2006].

Une synthèse de ces définitions fait ressortir un élément clé, l'apprentissage automatique permet à une machine de s'auto-améliorer. Ainsi, l'apprentissage automatique est analogue à l'apprentissage naturel. Il faut noter que cette analogie est au niveau des définitions des deux types d'apprentissage, mais pas au niveau de leurs capacités et leurs puissances.

Pour bien comprendre l'intérêt de l'apprentissage automatique et les problèmes pour lesquels il peut intervenir nous allons présenter certains de ses apports et les motivations de son utilisation.

1.4.3. Apprentissage Automatique: Apports et Motivations

Les motivations de l'apprentissage automatique découlent des domaines intervenant dans cette discipline. En effet comme l'apprentissage automatique est pluridisciplinaire, alors il peut apporter plusieurs aides et intervenir dans de multiples processus de décision. Pour montrer l'intérêt et l'apport de l'apprentissage automatique, nous commençons par présenter des questions précises auxquelles, cette discipline est capable de fournir des réponses. Puis nous synthétisons en présentant les cas généraux où l'apprentissage automatique est d'une grande importance. Les questions suivantes sont des exemples de problèmes précis pour lesquels les techniques de l'apprentissage automatique peuvent fournir des réponses satisfaisantes et adéquates:

- Basé sur les dommages causés par un séisme, peut-on déduire de nouvelles règles parasismiques?

- Cet hiver, y aura t-il des inondations dans une région basse?
- Un patient donné aura-t-il un accident cardio-vasculaire dans les dix ans à venir?
- Une molécule médicinale, qu'un laboratoire désire commercialiser, est-elle cancérigène?
- Quels documents pertinents faudra t-il présenter à un internaute?
- Comment une entreprise pourra-t-elle augmenter ses bénéfices?
- ???

La suite de point d'interrogation dénote des questions multiples auxquelles l'apprentissage automatique, peut apporter une aide considérable. Certes, pour plusieurs domaines, les experts humains peuvent être consultés pour fournir des réponses à ces questions, tels que des experts en parasismique pour étudier les dommages des constructions et proposer des normes antisismiques. Mais ces experts auront la difficulté d'exploiter toutes les données possibles et le risque d'oubli et d'omission serait néfaste; et c'est identique pour les inondations ou le domaine médical. Une prise de décision automatique pour fournir des réponses aux questions précédentes est intéressante. D'où le développement de systèmes experts, qui sont capables de mener un raisonnement à partir de faits décrivant le problème à résoudre et d'une expertise bien définie [Alliot et Schiex, 1994; Igor, 1987; Russel et Norvig, 2006]. Mais la difficulté est alors d'acquérir cette expertise, les experts humains sont le plus souvent incapables d'expliquer leurs raisonnements ou leurs savoirs faire. Alors de cette difficulté découle une motivation principale de l'apprentissage automatique: l'exploration de plusieurs données et expertises pour *la génération automatique des connaissances* [Cornuéjols et Miclet, 2002; Michalski et Chilausky, 1980; Mitchell, 1997; Nilsson, 1997]. Le système apprenant est donné des exemples de cas déjà traités pour générer automatiquement des connaissances générales; et ainsi les experts humains n'ont qu'à fournir leurs réponses sur des cas précis sans aucune définition de leurs expertises et raisonnements.

Pour les problèmes, où il n'existe pas d'expertise humaine, les cas traités peuvent être obtenus à partir des expériences passées ou à partir d'observations. Par exemple en utilisant les molécules qui ont été déterminées cancérigènes ou non par de lents tests en laboratoire sur des rats, le système apprenant peut décider si une nouvelle molécule est cancérigène ou non. Aussi pour présenter à un internaute des documents pertinents, aucune expertise n'est disponible. Pour ces problèmes, il ne s'agit plus d'expliquer une expertise humaine mais bien de découverte scientifique et de construction de modèles par apprentissage automatique [Dietterich, 2002; Michalski et Chilausky, 1980; Mjolsness et Decoste, 2001; Witten et Eib, 2005].

Maintenant si nous voulons généraliser et synthétiser afin de décrire les problèmes pour lesquels l'apprentissage automatique est d'une grande importance, nous caractérisons ces problèmes par l'une des propriétés suivantes:

- Les tâches ne sont bien définies que via un ensemble d'exemples spécifiant des couples d'entrées/sorties sans relations explicites et concises entre les entrées et les sorties.
- La quantité de connaissances disponibles est énorme et son exploitation, par des humains, est difficile.
- L'environnement change constamment et les applications doivent s'adapter à cet environnement ou évoluer au cours du temps.

Pour ces problèmes:

- Il est possible que les quantités importantes de données renferment des corrélations et des relations importantes, que les méthodes d'apprentissage automatique permettent de découvrir.
- L'apprentissage automatique peut permettre de mieux exploiter les masses volumineuses de connaissances.
- L'apprentissage automatique permet aux machines de s'adapter aux changements de l'environnement par création itérative de modèles de prise de décision.

Pour conclure cette section, qui constitue une partie complémentaire de la section précédente, nous établissons une analogie entre un développement ordinaire ou classique d'un programme et un développement basé sur l'apprentissage automatique.

En informatique, nous savons que le développement classique d'un programme nécessite d'explicitier les différentes étapes, permettant de construire la sortie attendue à partir de l'entrée. Donc de détailler de manière exhaustive la relation qui existe entre les entrées et les sorties en écrivant des algorithmes. Pour les problèmes caractérisés par les propriétés énumérées ci-dessus, l'écriture d'algorithmes est impossible soit parce que:

- Il est difficile d'établir la relation entre les entrées et les sorties.
- Les données sont tellement nombreuses, et sans relations explicites, qu'il est difficile de les exploiter par un algorithme.
- Toutes les données nécessaires ne sont pas disponibles lors du développement du programme, éventuellement à cause de changements ou d'évolution.

Pour ces problèmes, les techniques de l'apprentissage automatique offrent une méthode de développement alternative. Donc, plutôt que d'écrire une spécification formelle (c'est à dire les instructions à suivre) du comportement du programme; le programmeur fournit une base d'apprentissage, composée d'exemples d'entrée et de leur sortie attendue. Et les techniques de l'apprentissage automatique sont alors capables de généraliser en inférant la sortie associée à une entrée n'apparaissant pas dans la base d'apprentissage [Michalski et al., 1993; Mitchell, 1997]. Cette génération automatique de connaissances permet aux systèmes basés sur l'apprentissage automatique de résoudre les problèmes ne pouvant pas être résolus par un développement classique.

Pour connaître les techniques de l'apprentissage automatique, sachant que c'est un domaine pluridisciplinaire très riche en concepts et méthodes, le chapitre 2, de cette thèse, présentera une étude et analyse des principales méthodes de cette discipline.

1.5. Apprentissage Automatique dans un Agent

Après avoir présenté le contexte de base de notre travail, incluant : la technologie Agent et l'apprentissage automatique; nous arrivons au thème de cette thèse: l'apprentissage automatique dans un Agent. Un domaine de recherche récent et très vaste qui regroupe deux disciplines, chacune connue pour être pluridisciplinaire. Ayant défini les concepts et les notions de base de la technologie Agent et introduit l'apprentissage automatique et ses apports, nous pouvons maintenant présenter la problématique de notre travail ainsi que ses orientations principales. Nous commençons par une analyse des travaux existants. Puis à partir d'une synthèse, concernant l'existant, nous présenterons les objectifs et les axes principaux que nous avons dégagés pour notre travail de thèse.

1.5.1. Travaux Existants

Avec l'émergence de la nouvelle vision de l'intelligence artificielle, cherchant à résoudre des problèmes dans un monde réel, ouvert et dynamique, la notion d'Agent a eu une double évolution: l'Agent autonome caractérisé par son individualité et l'Agent social faisant partie d'un système multi-agents. Ainsi un Agent possède deux tendances:

1. Une tendance Individuelle qui considère un Agent comme une entité autonome agissant en isolation de d'autres Agents artificiels. Cette entité est définie, conçue et développée avec une vision centrée sur les règles et les mécanismes du fonctionnement de l'Agent individuel et autonome.

2. Une tendance Sociale tournée vers la collectivité et la cohabitation des Agents dans une société d'Agents, dite système multi-agents, ainsi que les mécanismes de communication et d'interaction dans la société d'Agents.

Les Agents autonomes et les systèmes multi-agents constituent un domaine de recherche en plein effervescence qui continue toujours à évoluer. Les premiers travaux théoriques, dans ce domaine, se sont intéressés d'une part aux architectures d'Agents réactifs, cognitifs et hybrides; et d'autres part aux différents aspects des systèmes multi-agents tels que: la communication, la coopération et la compétition.

Vu les apports de l'apprentissage automatique, le recours à ses techniques est d'une grande importance pour les deux tendances d'Agents. Les travaux, concernant l'apprentissage automatique dans les systèmes multi-agents, couvrent plusieurs aspects théoriques et pratiques. En effet le recours à l'apprentissage automatique dans les SMA et l'étude de ses apports ont été abordés très tôt et ils continuent à être des axes de recherches [Berthet et al., 1992; Crawford et Veloso, 2007; Edwards et Davies, 1993; Fischer et al., 2004; Huhns et Weiss, 1998; Kaelbling, 1993; Rovatsos et al., 2003; Sen et Weiss, 1999; Sian, 1991; Stone et Veloso, 2000; Thomas et al., 2007; Wang et Silva, 2008; Weiss, 1996; Weiss, 1997; Weiss, 1999 ; Zeng et Sycara, 1997]

Pour un Agent autonome ou système basé-Agent, l'apprentissage automatique diffère de l'apprentissage multi-agents. En effet Gérard Weiss [Weiss, 1996; Weiss, 1997] considère que ce dernier nécessite la prise en considération des interactions entre Agents. L'apprentissage multi-agents prend donc en compte l'influence de l'information échangée entre les Agents ainsi que les normes et les conventions qui règlent les comportements dans la société d'Agents. L'apprentissage multi-agents permet la cohabitation des Agents dans un SMA et la résolution des problèmes propres aux systèmes multi-agents [Sen et Weiss, 1999; Weiss, 1996; Weiss, 1997; Zeng et Sycara, 1997], tels que: la coopération, la planification, la compétition et la négociation. En effet, une recherche menée par Peter Stone et son équipe [Stone et Veloso, 2000] a montré que l'apprentissage automatique peut intervenir dans la résolution de plusieurs problèmes posés dans les SMA, entre autre :

- La coordination entre Agents.
- La planification des tâches des différents Agents dans un SMA.
- L'allocation de ressources aux différents Agents d'un SMA.
- La stabilité du SMA et sa convergence vers son but global.

Pour l'apprentissage automatique dans un Agent ou un système basé-Agent, l'étude de l'existant a fait ressortir un nombre important de travaux liés à des applications pratiques. Donc l'existence de plusieurs systèmes basés-Agent intégrant des techniques de l'apprentissage automatique pour des applications bien définies. La capacité d'apprentissage dans un Agent était, depuis les débuts de la technologie Agent, une haute qualité qui ne fait pas partie des caractéristiques de base d'un Agent, mais dont la plupart des auteurs [Jennings et al., 1998; Maes, 1990; Nwana et Wooldridge, 1996; Wooldridge et Jennings, 1995] souhaitaient avoir dans un Agent. D'où l'intérêt croissant du développement d'applications à base d'Agents utilisant des capacités de l'apprentissage automatique.

Les apports de la technologie Agent et ceux de l'apprentissage automatique ont orienté les travaux vers des applications pratiques. Ainsi les travaux existants sur l'apprentissage automatique dans les systèmes basés-Agent sont des systèmes dédiés à des domaines spécifiques. Ces travaux sont aussi variés et multiples que les domaines d'applications le sont. Ainsi une liste exhaustive ne peut être fournie, mais nous citons un sous ensemble représentatif comportant entre autres les travaux des auteurs les plus actifs et les plus connus dans le domaine de la technologie Agent, tels que: [Allen et al., 2007; Armstrong et al., 1995; Billsus et Pazzani, 1999; Chen et Sycara, 1998; Etzioni et Weld, 1994; Ferguson et al., 2009; Hsinchen et al., 1996; Kernchen et al., 2007; Koyama et al., 2001; Kozierok et Maes, 1993; Lang, 1995; Lieberman, 1995; Lieberman et al., 2001; Maes, 1992; Maes, 1994; Mencke et Dumke, 2007; Mitchell et Garuana, 1994; Naderi et al., 2007; Ndumu et al., 1998; Okamoto et al., 2009; Schwab et Koychev, 2000; Selker, 1994; Sheirin et Lieberman, 2001; Sheth, 1994; Tama et Wooldrige, 2004; Tecuci et al., 2002; Vivacqua, 1999; Wang et al., 2005; Wenyin et al., 2009; Yang et al., 2009; Zanker et Jessenitschnig, 2009]. Donc, un nombre important de systèmes, à base d'Agents augmentés par des techniques de l'apprentissage automatique, sont aujourd'hui disponibles, entre autres: Agents virtuel de bureau, Agents de recherche sur le Web, Agents de filtrage d'informations personnalisés, Agents de data mining, Agents d'enseignement intelligent basés sur les profils des usagers, Agents de voyage personnalisés, Agents dans le domaine commercial, médical et militaire.

En effet, ces dernières années, la technologie Agent a attiré l'attention des concepteurs d'applications pour plusieurs raisons, entre autre:

- La possibilité de récupération et de filtrage des quantités énormes d'informations disponibles sur les réseaux d'entreprise et particulièrement sur l'Internet.

- L'intégration des techniques de l'intelligence artificielle dans cette technologie permet d'obtenir des applications informatiques intelligentes et autonomes.
- La possibilité de personnalisation des applications et des services en capturant les préférences, buts et désirs des utilisateurs grâce à l'apprentissage automatique.

Ainsi, la technologie Agent, couplée à des techniques de l'intelligence artificielle, en particulier l'apprentissage automatique, est fortement présente dans de nombreuses applications de bureau, d'entreprise, d'industrie, d'enseignement, de médecine, d'interfaces adaptables, ...

1.5.2. Synthèse

L'analyse des travaux existants, concernant l'apprentissage automatique dans un Agent, nous a révélé l'existence d'un nombre considérable de travaux, à base d'Agent couplé à des techniques de l'apprentissage automatique, mais qui sont dédiés et spécifiques. En effet, ces travaux sont réalisés, à partir de la définition des fonctionnalités qu'un système basé-Agent doit offrir selon le domaine de l'application; puis ils choisissent et appliquent des méthodes de l'apprentissage automatique. Le constat auquel nous sommes arrivés est que chacun de ces travaux démarre d'un domaine d'application précis, puis opte pour une méthode d'apprentissage qu'il trouve adéquate pour l'application. Pour motiver leurs choix, certains travaux appliquent plus d'une méthode d'apprentissage pour le même problème, deux généralement, puis décident de la meilleure méthode en se basant sur les résultats expérimentaux des différents choix. Ainsi, ces travaux correspondent à des systèmes liés aux domaines d'applications et possèdent des architectures dédiées et spécifiques.

Nous pensons que cette spécificité est justifiable par plusieurs arguments, entre autres :

1. Vu ses multiples apports, la technologie Agent suscite l'attention des concepteurs pour le développement d'applications à base d'Agents pour des problèmes bien déterminés. La technologie Agent a rapidement dépassé les frontières des universités et des laboratoires de recherches pour être adoptée aussi par les développeurs d'applications. Ainsi, elle est largement utilisée pour résoudre des problèmes réels dans des domaines variés et multiples. La section 1.3.7 a montré une partie de ces domaines. La technologie Agent offre un répertoire puissant de concepts, de paradigmes, d'outils et de techniques permettant d'améliorer considérablement le développement de systèmes informatiques, d'où son large étendue.

2. La croissance rapide de l'intérêt pour la technologie Agent a été favorisée par la croissance de l'Internet, infrastructure et support important pour les applications à base d'Agents. Aussi, la prolifération de l'Internet et son utilisation massive ont donné naissance à de nouvelles applications d'Agents.
3. Les apports de l'apprentissage automatique, entre autres: la résolution de problèmes complexes, l'adaptation aux changements de l'environnement et l'amélioration des performances des systèmes, poussent les concepteurs à recourir à ses techniques.

Pour pallier le manque de travaux de base, indépendants des domaines d'applications, concernant l'apprentissage automatique dans un Agent, nous avons tracé un certain nombre d'objectifs vers lesquels nos propositions et contributions visent à converger. La sous section suivante présente brièvement ces objectifs et les chapitres suivants présenteront le travail accompli pour les atteindre.

1.5.2. Objectifs de Notre Travail

Pour la recherche, la technologie Agent est un domaine récent, qui n'est pas encore à sa maturité. Pour l'apprentissage automatique dans un Agent, la recherche fait appel aux contributions pouvant constituer une base pour des travaux ultérieurs et ouvrants de nouvelles perspectives.

L'étude des concepts de base de la technologie Agent et des apports de l'apprentissage automatique, a engendré, chez nous, un certain nombre d'interrogations, entre autres :

- *Est-ce que tout Agent doit faire recours à l'apprentissage automatique?*
- *Quels sont les différents types d'Agents?*
- *Quels sont les types d'Agents les plus concernés par l'apprentissage automatique?*

Pour répondre à ces préoccupations, nous avons fixé comme premier objectif, la détermination des différents types d'Agents nécessitant le recours aux techniques de l'apprentissage automatique. Alors une classification d'Agents, selon des critères à bien définir au préalable, est nécessaire. Cette classification d'Agents, avec ses critères de base, et la définition des différents types d'Agents Apprenants seront présentés au chapitre 3 de cette thèse.

Pour l'apprentissage automatique, il était question de déterminer:

- *Selon quelle perspective faut-il aborder ce domaine? Faut-il étudier une ou plusieurs méthodes d'apprentissage?*

L'étude d'une méthode d'apprentissage donnée a pour buts : son utilisation dans un domaine d'application précis ou l'optimisation de certains de ses paramètres afin d'optimiser la méthode étudiée. Pour notre travail le recours à l'apprentissage automatique ne vise pas les deux objectifs précédents. Ainsi, pour nous il s'agit d'étudier et d'analyser les différentes catégories et méthodes d'apprentissage, pour pouvoir répondre à notre préoccupation suivante:

- *Pour chaque type d'Agents Apprenants quelles sont les méthodes d'apprentissage les plus appropriées?*

A ce niveau, nous nous sommes trouvés devant une large panoplie et une pléthore de méthodes d'apprentissage avec des concepts et des techniques multi-variés. Il faut donc catégoriser ces méthodes et étudier les plus représentatives; ce travail fera l'objet du chapitre 2 de ce mémoire de thèse.

Notre objectif global est d'apporter un support de base, une assise conceptuelle et un modèle général validés par des expérimentations réelles, au thème de: l'apprentissage automatique dans un Agent. Selon cet objectif, la première finalité de notre travail est l'établissement d'une mise en correspondance entre les types d'Agents-Apprenants d'une part et les approches/méthodes d'apprentissage automatique d'autre part. Le chapitre 3 présentera cette mise en correspondance.

Le second but tracé, pour notre travail, est la construction d'un modèle général et générique pour les systèmes basés Agent-Apprenant. Un modèle qui doit identifier tous les éléments nécessaires à l'apprentissage automatique dans les systèmes basés Agent. Pour atteindre les propriétés de généralité et de généricité, ce modèle doit être indépendant de tout domaine d'application et doit couvrir les différentes approches/méthodes de l'apprentissage automatique. Le chapitre 4 présentera le modèle que nous avons construit ainsi que l'ensemble des concepts, formant l'assise conceptuelle pour ce modèle.

Pour exploiter et réutiliser le modèle établi, des instanciations et des expérimentations pratiques sont à mener. Le chapitre 5 présentera deux instanciations complètes, de notre modèle ainsi que de ces concepts de base. Les deux études de cas correspondent à des domaines d'applications de systèmes basés Agent nécessitant le recours à l'apprentissage automatique.

1.6. Conclusion

La technologie Agent et l'apprentissage automatique offrent des concepts et des outils puissants et prometteurs pour le développement de systèmes informatiques et pour la résolution de problèmes complexes. A travers ce chapitre, nous avons présenté les concepts de base de la technologie Agent, qui

constitue un axe central pour notre travail. Le paradigme Agent, ses caractéristiques essentielles, ses différentes approches et visions ainsi que les architectures réactives, cognitives et hybrides ont été présentés. Pour l'apprentissage automatique, c'est surtout ses apports pour la résolution de problèmes complexes qui ont été discutés. Le chapitre suivant sera consacré à l'étude et l'analyse des différentes catégories et méthodes de l'apprentissage automatique.

L'étude de l'existant concernant l'apprentissage automatique dans un Agent, nous a révélé un nombre considérable de travaux à base d'Agents faisant recours aux techniques de l'apprentissage automatique. Mais ces travaux correspondent à des systèmes liés à des domaines d'applications et possèdent ainsi des architectures dédiées et spécifiques. Notre travail vise à contribuer au thème de l'apprentissage automatique dans un Agent, par la réalisation de certains objectifs dont le premier est l'établissement d'une mise en correspondance entre les types d'Agents-Apprenants et les approches/méthodes de l'apprentissage automatique; cet appariement est élaboré à partir d'un support de base préalablement défini. Le second objectif est la construction d'un modèle général et générique, pour les systèmes basés Agent-Apprenant, fondé sur un ensemble de concepts bien définis. Finalement des expérimentations menées, instanciant et mettant en œuvre les différents composants du modèle général ainsi que sa base conceptuelle, seront présentées. Ces études de cas permettent d'exploiter et de réutiliser ce modèle et aussi le valider. Les chapitres suivants, de cette thèse, présenteront le travail réalisé pour atteindre les objectifs tracés.

Chapitre 2

Approches et Méthodes de L'Apprentissage Automatique

2.1. Introduction

Le grand attrait du domaine de l'apprentissage automatique pour les chercheurs est qu'il offre une diversité considérable de tâches de recherche et de terrains pour l'expérimenter. Cette diversité est due au fait que l'apprentissage peut accompagner n'importe quel type de résolution de problèmes, et peut donc être étudié dans un grand nombre de contextes différents [Michalski et al., 1993]. L'objectif de l'étude de l'apprentissage automatique, pour notre contexte, a été défini au chapitre 1. Il correspond à l'étude et l'analyse des différentes catégories et méthodes de l'apprentissage automatique. Ce dernier est un processus qui s'effectue progressivement, il consiste à construire et à modifier des représentations à partir d'une série d'expériences. En d'autres termes c'est le processus d'acquisition automatique de connaissances, de pratiques et de compétences par l'observation, l'imitation, l'essai et la répétition [Cornuéjols et Miclet, 2002; Michalski et al., 1993; Mitchell, 1997]. Il permet à une machine d'accroître ses connaissances et de modifier son comportement à la suite de ses expériences et de ses actes passés.

Tom Mitchell stipule que « *Depuis que les ordinateurs ont été inventés, nous nous sommes demandés s'ils pourraient être faits pour apprendre. Si nous pourrions comprendre*

comment les programmer pour apprendre, et pour s'améliorer automatiquement avec l'expérience, l'impact serait spectaculaire¹». [Mitchell, 1997]

Cette citation montre l'importance de l'apprentissage automatique. En effet depuis longtemps les chercheurs ont tenté de concevoir et de construire des machines capables d'apprendre, en particulier les travaux de McCulloch et Pitts sur le neurone artificiel en 1943 [McCulloch et Pitts, 1943]. Ainsi les travaux, du domaine de l'apprentissage automatique, remontent aux débuts de l'informatique et ils ne cessent de se développer. Les recherches et les développements pratiques, dans ce domaine ont produit une grande variété de techniques d'apprentissage. La pléthore des méthodes d'apprentissage existantes rend difficile leur étude. Ainsi, pour atteindre l'objectif de ce chapitre, il faut se baser sur une classification des méthodes d'apprentissage en catégories et approches globales.

Dans la littérature, l'apprentissage automatique est souvent caractérisé par le fait qu'il soit supervisé ou non supervisé [Cornuéjols et Miclet, 2002; Nilsson, 1997; Russel et Norvig, 2006; Witten et Eib, 2005], nous commençons donc par présenter ces deux types d'apprentissage. Par la suite, nous exposerons une étude des approches de l'apprentissage automatique dégagées, à partir d'un critère de classification que nous avons adopté, ainsi que leurs méthodes les plus représentatives et les plus connues.

2.2. Apprentissage Supervisé

L'apprentissage supervisé, dit aussi avec professeur ou oracle, à pour entrée des données dont l'appartenance est précisée par un superviseur. Une méthode d'apprentissage supervisée permet d'apprendre, à partir des données classées, un modèle pour la classification de nouvelles données [Cornuéjols et Miclet, 2002; Dougherty et al., 1995; Witten et Eib, 2005]. Le but de l'algorithme de classification est de classer correctement les nouveaux exemples dans les classes définies dans la phase d'apprentissage. Alors le système apprend à ordonner selon un modèle de classement.

Pour bien comprendre, nous dirons que l'apprentissage supervisé correspond au cas où l'objectif de l'apprentissage est déterminé explicitement via la définition d'une cible à prédire. Dans ce cas un ensemble D , dit base d'apprentissage, correspond à un ensemble de n paires d'entrées xt et de cibles associées yt :

$$D = \{(xt, yt) / xt \in X, yt \in Y\} \text{ et } t \in \{1, 2, 3, \dots, n\}$$

¹ *Ever since computers were invented, we have wondered whether they might be made to learn. If we could understand how to program them to learn, to improve automatically with experience, the impact would be dramatic. [Mitchell, 1997]*

D peut être récolté en fournissant l'ensemble des entrées à un groupe de personnes et en leur demandant d'associer à chacune de ces entrées une cible appropriée, dans le contexte du problème à résoudre. La tâche d'un algorithme d'apprentissage supervisé est alors d'entraîner un modèle qui puisse imiter ce processus d'étiquetage par un humain, c'est à dire, qui puisse prédire pour une entrée x quelconque la valeur de la cible y qui aurait normalement été donnée par un humain. Autrement dit l'apprentissage supervisé a pour but de trouver ou d'approximer la fonction qui permet d'affecter la bonne étiquette aux nouveaux exemples.

La nature de l'ensemble Y d'où proviennent les cibles dépend du type de problème à résoudre. Deux types de problèmes fréquents sont les problèmes de classification et de régression, qui seront définis par la suite.

Pour s'assurer de la bonne fonction d'approximation, les exemples étiquetés disponibles sont initialement partagés en deux ensembles: un pour l'apprentissage, l'autre pour le test. Ainsi, le système apprenant démarre de l'ensemble d'apprentissage, apprend une fonction d'approximation ou de classification, puis cette fonction est testée en utilisant l'ensemble de test.

Quand les exemples d'apprentissage sont étiquetés, mais il existe un manque au niveau des données ou des étiquettes, on utilise l'apprentissage semi-supervisé [Cornuéjols et Miclet, 2002]. L'algorithme d'apprentissage vise à faire apparaître la distribution sous-jacente des exemples dans leur espace de description. L'apprentissage semi-supervisé est utilisé comme une alternative à l'apprentissage supervisé pour pallier le manque de données ou de leur étiquetage.

2.3. Apprentissage non-Supervisé

L'apprentissage non-supervisé désigné aussi par le clustering, le regroupement, la classification par distance ou l'apprentissage par agrégation permet de réunir les données qui se «ressemblent» dans les mêmes clusters ou groupes [Dougherty et al., 1995; Mitchell, 1997; Witten et Eib, 2005]. Ainsi, lorsque les classes d'appartenance des données ne sont pas connues, l'apprentissage non supervisé permet de découvrir ces classes par regroupement des données présentant une similitude au sens d'une certaine métrique.

Le manque de connaissance peut être lié au manque d'informations sur les données, ou bien, le volume d'informations est trop important pour pouvoir être étiqueté à la main (par exemple en reconnaissance de la parole).

Le système ne dispose que d'exemples, mais pas d'étiquettes, alors le nombre de classe et leur nature ne sont pas prédéfinis. L'apprentissage non-supervisé correspond donc au cas où aucune cible n'est prédéterminée; aussi la nature de la fonction f , qui doit être retournée

par l'algorithme d'apprentissage, n'est pas définie explicitement. Ainsi, l'ensemble d'entraînement ne contient que des entrées:

$$D = \{xt / xt \in X\} \quad \text{et } t \in \{1,2,3,\dots,n\}$$

Dans ce cas l'algorithme d'apprentissage cherche à trouver des régularités dans une collection d'exemples, puisque il ne connaît pas la classe à laquelle les exemples d'apprentissage appartiennent. Une technique employée consiste à implémenter des algorithmes pour rapprocher les exemples les plus similaires et éloigner ceux qui ont le moins de caractéristiques communes. Ces groupes d'exemples similaires sont parfois appelés des prototypes. Ainsi, l'algorithme d'apprentissage non-supervisé capture certains éléments de la véritable distribution ayant généré D .

Nous arrivons maintenant aux approches d'apprentissage pour présenter, par la suite, les méthodes les plus représentatives pour chacune des approches dégagées.

2.4. Approches de l'Apprentissage Automatique

Les méthodes d'apprentissage peuvent être classifiées selon de nombreux points de vue différents, tels que: les informations sur les exemples d'apprentissage, le type de la représentation des connaissances et le type de la stratégie utilisée, dite aussi la technique d'inférence [Cornuéjols et Miclet, 2002; Michalski et al., 1993; Mitchell, 1997].

En conséquence et selon les informations sur les exemples d'apprentissage, nous retrouvons les catégories présentées ci-dessus: supervisé, semi-supervisé ou non supervisé. Si on considère le type de la représentation des connaissances on distingue les méthodes symboliques des méthodes connexionnistes dites aussi numériques, correspondant aux deux grandes classes de mode de représentation des connaissances: la représentation symbolique et la représentation connexionniste [Haton, 1990; Kayser, 1997; Michalski et al., 1993, Mitchell, 1997]. Les stratégies d'inférence, l'induction et la déduction [Nilsson, 1980]; donnent lieu à deux grandes classes de méthodes d'apprentissage: les méthodes inductives et les méthodes déductives.

Pour notre travail, nous avons adopté la technique d'inférence pour dégager les classes des méthodes d'apprentissage. Nous rappelons que notre objectif ne vise pas la classification, elle-même, des méthodes d'apprentissage. Mais, il vise à faire face à la pléthore des méthodes existantes, en dégagant des approches globales d'apprentissage. Ceci nous permet de réduire la complexité des choix délicats des méthodes variées pour les faire correspondre aux Agents-Apprenants et de disposer de facteurs bien définis orientant ces choix. Notre motivation pour la stratégie d'inférence est le fait que celle-ci soit liée au procédé général de la méthode qui peut être: synthétique ou analytique. Un procédé synthétique permet de créer de nouvelles connaissances à partir d'un grand ensemble de

données initiales, et un procédé analytique sert à analyser et à transformer les connaissances disponibles [Michalski et al., 1993; Mitchell, 1997]. Le procédé synthétique ou analytique peut dépendre du domaine d'application et des connaissances à apprendre; ainsi il peut être utilisé dans le choix d'une approche puis d'une méthode d'apprentissage pour un Agent-Apprenant, le chapitre 3 traitera ce sujet.

En se basant sur les deux modes d'inférence: l'induction et la déduction, nous disposons de deux catégories ou approches d'apprentissage automatique. Et afin de préparer un support étendu, pour couvrir les différents types d'Agents-Apprenants, nous avons considéré une troisième catégorie possédant un principe d'apprentissage différent des deux précédentes. La combinaison de deux ou trois catégories est possible, ainsi les approches dégagées sont les suivantes:

1. *Inductive ou Synthétique.*
2. *Déductive ou Analytique.*
3. *Par renforcement.*
4. *Hybride.*

2.4.1. Approche d'Apprentissage Inductive

L'apprentissage par induction reste toujours l'un des principaux axes étudiés dans le domaine de l'apprentissage automatique. Cet apprentissage apprend des règles générales qui représentent les connaissances obtenues à partir d'exemples en se basant sur l'induction comme méthode d'inférence principale.

L'induction cherche des lois générales à partir de faits particulier, elle génère du sens en passant des faits à la loi, du particulier au général [Nilsson, 1980]; ainsi elle enrichit la conscience de nouveaux faits, elle est alors synthétique. Et l'apprentissage inductif est dit aussi apprentissage synthétique, car il consiste à synthétiser et construire de nouvelles connaissances. Les méthodes d'apprentissage basées sur l'induction font partie de l'approche inductive ou synthétique.

Un algorithme d'apprentissage par induction reçoit un ensemble d'exemples en entrée et doit produire des règles de classification, permettant de classer les nouveaux exemples. Le processus d'apprentissage cherche à créer une représentation plus générale des exemples, selon une méthode de généralisation de connaissances. Ce type de méthodes est aussi appelé apprentissage de concepts ou bien acquisition de concepts [Mitchell, 1997; Mitchell, 1999].

Les algorithmes d'apprentissage par induction peuvent fonctionner de façon supervisée ou non supervisée [Nilsson, 1997]; et ils ont besoin d'un nombre significatif d'exemples, dit exemples d'entraînement, pour pouvoir bien généraliser les connaissances, c'est-à-dire induire des règles ou des concepts [Quinlan, 1979]. L'approche d'apprentissage inductive comprend des procédés de généralisation, de spécialisation, de transformation, de correction et de raffinement pour les représentations des connaissances. Cette approche

comporte un nombre important de méthodes d'apprentissage dont les plus représentatives et les plus répandues seront présentées à la section 2.5 de ce chapitre.

2.4.2. Approche d'Apprentissage Déductive

L'apprentissage déductif a une démarche qui s'oppose avec celle de l'apprentissage inductif, c'est un processus qui permet de dériver de nouvelles connaissances à partir d'une théorie de domaine préétablie. Alors la démarche part du général vers le particulier, de l'abstrait vers le concret [Mitchell, 1997]; et utilise la déduction comme méthode d'inférence fondamentale.

La déduction logique se fonde sur des axiomes ou des définitions et produit des résultats tautologiques, qui sont des conséquences de la loi [Nilsson, 1980]. La déduction ne produit donc aucune nouvelle connaissance, au sens où les propositions déduites sont virtuellement contenues dans leurs axiomes; par conséquent la déduction est analytique. L'approche analytique consiste donc en une amélioration des connaissances déjà existantes.

L'approche d'apprentissage analytique, utilise des connaissances préexistantes et un raisonnement déductif pour augmenter l'information fournie par les ensembles d'exemples. Dans les méthodes déductives, les connaissances sont dérivées à partir d'un simple cas par explication des raisons pour lesquelles il représente un exemple du concept appris [Michalski et al., 1993]. Les connaissances préexistantes sont utilisées pour analyser, ou expliquer, comment chaque exemple, observé lors de l'apprentissage, satisfait les concepts existants. Ensuite, cette explication est utilisée pour différencier, les attributs pertinents de l'exemple d'apprentissage, de ceux qui ne le sont pas. De cette façon, l'exemple pourra être généralisé par un raisonnement logique.

2.4.3. Approche d'Apprentissage par Renforcement

L'apprentissage par renforcement est une approche qui vise à résoudre des problèmes complexes où on n'indique au système que le but à atteindre et ce dernier doit apprendre par une succession d'essais/erreurs, comment atteindre le but fixé [Dietterich, 2002, Sigaut et Buffet, 2008; Sutton et Barto, 1998]. L'apprentissage par renforcement fait référence donc à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences, ce qu'il convient de faire en différentes situations, de façon à optimiser une récompense numérique au cours du temps, [Sun, 2001; Sutton et Barto, 1998]. Il s'agit d'une évaluation non assistée qui fournit une mesure relative de la performance. Le système doit identifier les actions à exécuter pour que cette mesure soit maximisée. L'apprentissage par renforcement consiste à apprendre quoi faire, comment associer des actions à des situations, afin de maximiser quantitativement une récompense, dite aussi le signal de renforcement. La figure suivante montre la relation entre un système utilisant un apprentissage par renforcement et son environnement.

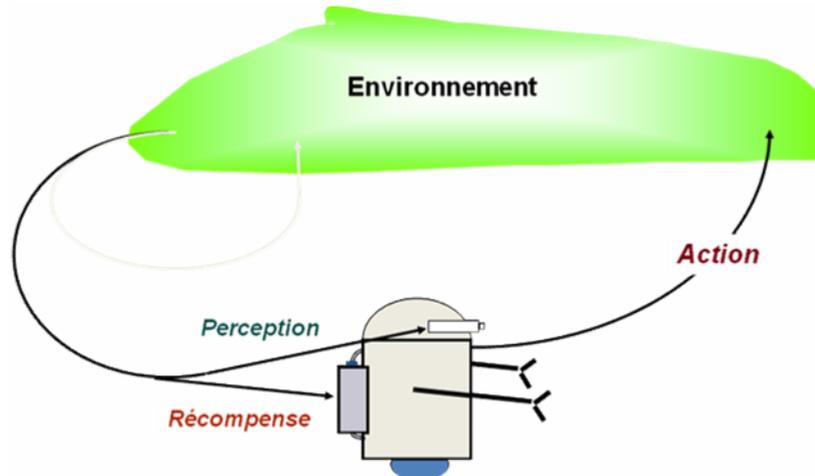


Figure 2.1 : Schématisation de l'apprentissage par renforcement.

D'après [Mitchell, 1997; Sutton et Barto, 1998], les éléments essentiels d'un apprentissage par renforcement sont :

- La politique.
 - La fonction de renforcement.
 - La fonction valeur.
- La politique, définit le comportement de l'apprenant, c'est une application des perceptions, ou situations de l'environnement, dans les actions. C'est donc un ensemble d'associations (situation, action) avec un algorithme de recherche de la bonne action pour la situation courante.
 - La fonction de renforcement définit le but à atteindre, elle fait correspondre à chaque couple (situation, action) une récompense servant à récompenser ou pénaliser le système pour avoir choisi une action donnée pour la situation courante.
 - La fonction valeur est un indicateur de la qualité du choix d'une action au long terme. La fonction d'évaluation notée $Q(s,a)$ pour Qualité (situation, action) permet de déterminer la récompense accumulée sur le long-terme.

Une difficulté de l'apprentissage par renforcement est le dilemme *exploitation/exploration* [Mitchell, 1997; Sutton et Barto, 1998]. L'exploitation sert à se rappeler quelles actions ont bien fonctionné par le passé et l'exploration à pour but de trouver les bonnes actions. Le système exploite ce qu'il connaît afin d'obtenir la plus grande récompense, mais pour découvrir l'action qui conduit au meilleur rendement, il a besoin d'explorer plusieurs possibilités, des fois très larges pour être explorées. Au chapitre trois, nous prenons en considération ce dilemme.

A la section 2.4, nous avons signalé que le principe de l'approche par renforcement diffère des principes des approches inductive et déductive. Les concepts d'action et de signal de renforcement sont ceux qui distinguent le plus l'apprentissage par renforcement des apprentissages déductif et inductif qu'ils soient supervisés ou non supervisés. Contrairement aux approches inductive et déductive, le comportement à apprendre n'est pas explicitement donné par une cible à prédire ou des entrées dont il faut modéliser la structure, mais doit plutôt être défini à l'aide d'un signal de renforcement. L'apprentissage par renforcement s'intéresse à la notion d'action qui est différente de celle des entrées d'apprentissage associées aux approches inductive et déductive.

2.4.4. Approche d'Apprentissage Hybride

L'apprentissage hybride est une approche qui combine deux ou éventuellement trois approches parmi celles précédemment présentées. Généralement, l'apprentissage hybride, combine les deux approches inductive et déductive ou l'une de ces deux approches avec un apprentissage par renforcement [Michalski et al., 1993; Mitchell, 1997; Sun, 1992]. L'apprentissage déductive-inductive est utilisé surtout quand la théorie d'un domaine est incomplète, dans ce cas les exemples d'entraînement sont associés à cette théorie pour former l'entrée du processus d'apprentissage. L'approche hybride permet certainement de tirer profit de chacune des approches qui la constituent, mais elle reste difficile à mettre en œuvre [Towel et Shavlik, 1989]. Une approche déductive-inductive, par exemple, permet d'exploiter toutes les connaissances disponibles sur le problème à traiter, donc les théories de domaines et les exemples d'entraînement.

En résumé, l'approche d'apprentissage inductive vise principalement à créer des connaissances nouvelles à partir de l'information donnée en entrée. Cette approche utilise l'induction comme méthode d'inférence fondamentale. L'approche d'apprentissage déductive a pour but principal l'analyse et la transformation, sous une forme plus efficace, des connaissances existantes. L'approche analytique utilise la déduction comme méthode d'inférence. La différence entre ces deux approches d'apprentissage réside donc, d'une part dans la méthode d'inférence utilisée et d'autre part dans l'exploitation des connaissances préalables du domaine. L'apprentissage inductif ne nécessite aucune connaissance sur le domaine, mais ses résultats doivent être validés. Le point fort de l'apprentissage analytique est la déduction, mécanisme d'inférence préservant la vérité. Pour des connaissances du domaine incomplètes, une approche hybride est souhaitable elle permet de compléter la théorie d'un domaine par les exemples d'entraînement. L'apprentissage par renforcement, dont le principe diffère des autres approches, est adéquat pour l'apprentissage de politiques d'actions permettant à un système de maximiser les récompenses reçues. Maintenant que les approches d'apprentissage sont définies, nous présentons pour chacune d'elles les méthodes les plus représentatives.

2.5. Méthodes d'Apprentissage Inductives

Les méthodes d'apprentissage inductives, étant basées sur l'induction, permettent de faire des prédictions en opérant des généralisations ou des classifications. En effet une méthode d'apprentissage inductive a pour objectif d'induire, à partir d'un ensemble d'exemples, une fonction de prédiction ou un classifieur $f: X \rightarrow Y$.

Le classifieur f associe à une entrée ou observation x de X une sortie ou étiquette $f(x) = y$ de Y . Ainsi une méthode d'apprentissage inductive doit apprendre une fonction de prédiction, permettant au système apprenant de réaliser la classification de nouveaux exemples. Les méthodes inductives peuvent être caractérisées par le type de classification supportée; en effet, il existe trois types de classification [Cornuéjols et Miclet, 2002; Duda et al., 2000]:

1. La classification binaire qui correspond au cas où deux étiquettes sont possibles, et qui sont arbitrairement désignées par 1 et -1 ; donc $Y = \{-1, +1\}$. Une entrée est dite positive si son étiquette est 1 et négative sinon. Une telle fonction de prédiction permet de prédire une classe d'appartenance ou sa négation. Par exemple, prédire s'il fera beau demain ou non à partir de l'observation des paramètres météorologiques d'aujourd'hui, tels que la pression barométrique, la température, le vent, ...etc.
2. La classification multi-classes qui correspond au cas où Y est un espace discret comportant un «petit» nombre d'éléments: $Y = \{1, 2, \dots, n\}$. Ce type de classifieur permet de prédire plusieurs classes et ainsi il est plus puissant qu'un classifieur binaire; en effet il permet de qualifier, par exemple, le temps qu'il fera demain par une valeur choisie parmi un ensemble de valeurs prédéfinies (beau, pluvieux, nuageux, ...).
3. La régression qui correspond au cas où Y est l'ensemble des réels R . Un tel classifieur permet de prédire la température qu'il fera demain. L'espace d'entrée X est une représentation numérique de l'observation qui prend généralement la forme d'un vecteur de réels. Chaque coordonnée x_i de ce vecteur décrit une caractéristique (feature), c'est-à-dire une propriété de l'observation qui est jugée pertinente pour la tâche de prédiction.

Les méthodes d'apprentissage inductives ou synthétiques sont très variées, les plus connues sont [Cornuéjols et Miclet, 2002; Michalski et al., 1993; Mitchell, 1997; Nilsson, 1997; Witten et Eib, 2005]:

- L'apprentissage par arbres de décisions.
- L'apprentissage par réseaux de neurones artificiels.
- L'apprentissage par algorithmes génétiques.
- L'apprentissage par programmation logique inductive.
- L'apprentissage basé sur l'analogie.

2.5.1. Apprentissage par Arbres de Décisions

L'induction par arbres de décisions est l'une des formes d'apprentissage les plus simples et les plus efficaces, souvent utilisée en pratique. Les arbres de décisions inductifs font partie de l'apprentissage de concept, où il s'agit d'apprendre des concepts généraux ou des catégories générales [Mitchell, 1997; Nilsson, 1996]. Les arbres de décisions permettent la prédiction de classes (concepts) à partir d'exemples, ils sont utilisés principalement dans le cadre de l'apprentissage supervisé ne nécessitant pas de connaissances du domaine. Pour l'apprentissage à base d'arbres de décisions, nous disposons d'un ensemble d'exemples $E = \{(x, c)\}$, où chaque exemple est constitué:

1. d'un ensemble x d'attributs.
2. de la classe c de l'exemple, qui appartient à l'ensemble des classes possibles C .

Un arbre de décision est construit en découpant successivement l'ensemble d'exemples E .

1. Si tous les exemples sont dans une seule classe, on place une feuille pour cette classe.
2. Sinon, on choisit l'attribut le plus discriminant possible, et on découpe l'ensemble d'exemples suivant cet attribut.

Un arbre de décision est donc la représentation graphique d'une procédure de classification récursive. A toute description complète est associée une seule feuille de l'arbre. Cette association est définie en commençant à la racine de l'arbre et en descendant selon les valeurs des attributs qui étiquettent les nœuds internes jusqu'à une feuille [Nilsson, 1996]. Un exemple étant constitué d'une liste d'attributs, la construction d'un arbre de décision consiste donc à utiliser les attributs pour subdiviser progressivement l'ensemble d'exemples en sous-ensembles de plus en plus fins.

Un arbre de décision est un outil d'aide à la décision et à l'exploration de données discrètes et continues. Sa lisibilité, sa rapidité d'exécution expliquent sa popularité [Utgoff, 1989]. La lisibilité est un point fort des arbres de décision, qui assure en grande partie leur popularité auprès des praticiens. En effet un arbre de décision peut être lu et interprété directement. Aussi il est possible de le traduire en base de règles sans perte d'informations, parce qu'un arbre est un enchaînement hiérarchique de règles logiques construites automatiquement à partir d'une base d'exemples.

A partir d'un ensemble d'exemples d'entraînement, illustré par le table ci-dessous, représentant des données météorologiques (ensoleillement, température, humidité, vent); il s'agit de prédire le comportement de sportifs (jouer ou non) par construction d'un arbre de décision.

Numéro	Ensoleillement	Température (°F)	Humidité (%)	Vent	Jouer
1	soleil	75	70	oui	oui
2	soleil	80	90	oui	non
3	soleil	85	85	non	non
4	soleil	72	95	non	non
5	soleil	69	70	non	oui
6	couvert	72	90	oui	oui
7	couvert	83	78	non	oui
8	couvert	64	65	oui	oui
9	couvert	81	75	non	oui
10	pluie	71	80	oui	non
11	pluie	65	70	oui	non
12	pluie	75	80	non	oui
13	pluie	68	80	non	oui
14	pluie	70	96	non	oui

Table 2.1: Exemples d'entraînement de données météorologiques [Quinlan, 1986].

Le but donc est de prédire à partir de la table de données météorologiques, la variable «jouer» en procédant à des divisions ou partitionnement successif des exemples en groupes les plus homogènes possibles. Ceci correspond à la construction de l'arbre de décision, illustré par la figure 2.2 suivante.

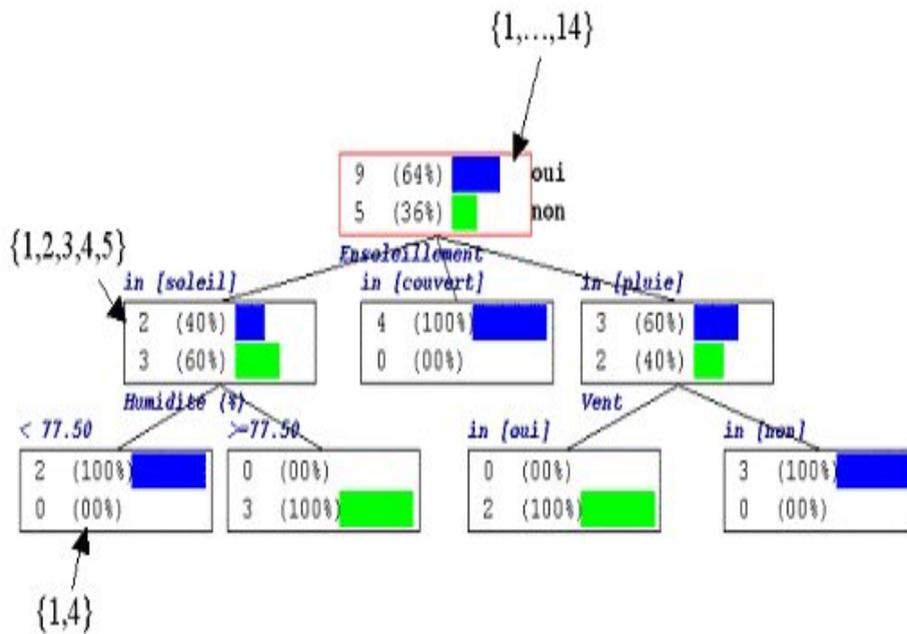


Figure 2.2: Arbre de décision correspondant à la table 2.1 [Quinlan, 1986].

Comme nous le constatons sur la figure 2.2, chaque nœud de l'arbre décrit un partitionnement selon la variable à prédire, ici jouer = oui ou jouer = non. La division est poursuivie jusqu'à obtention de groupes suffisamment homogènes; au niveau des feuilles de l'arbre, qui correspondent aux classes d'affectation des nouveaux exemples.

L'algorithme le plus représentatif de l'induction par arbres de décisions est l'ID3 (Inductive Decision Tree) de Quinlan [Quinlan, 1986]. Cet algorithme est largement utilisé pour les avantages précédemment cités, des arbres de décisions, d'une part et d'autre part pour l'efficacité de sa procédure de classification. ID3 travaille sur des attributs à valeurs discrètes et produit un arbre de décision multi-branches, à partir des exemples d'entraînement. Cet algorithme utilise une procédure récursive et une mesure d'entropie pour la sélection des attributs; cette mesure est fondée sur la théorie de l'information de Shannon [Quinlan, 1979]. ID3 est robuste aux erreurs, capable d'apprendre des expressions disjonctives facilement interprétables.

2.5.2. Apprentissage par Réseau de Neurones Artificiels

Les Réseaux de Neurones Artificiels (RNA) regroupent un certain nombre de modèles avec l'intention d'imiter quelques fonctions du cerveau humain, en reproduisant certaines de ses structures de base. Un réseau de neurones artificiels, dit aussi réseau connexionniste est constitué par un graphe orienté et pondéré. Les nœuds de ce graphe constituent les neurones artificiels ou formels ou bien les unités du réseau. Les unités sont dotées d'un état interne, appelé état d'activation, qu'elles peuvent propager aux autres unités du graphe en passant par des arcs pondérés appelés connexions, liens synaptiques ou bien poids synaptiques [Clément, 1989].

La règle qui détermine l'activation d'un neurone en fonction de ses entrées et de leurs poids respectifs s'appelle règle d'activation ou fonction d'activation [Hopfield, 1982]. Les éléments constitutifs d'un neurone artificiel sont schématisés par la figure 2.3 suivante.

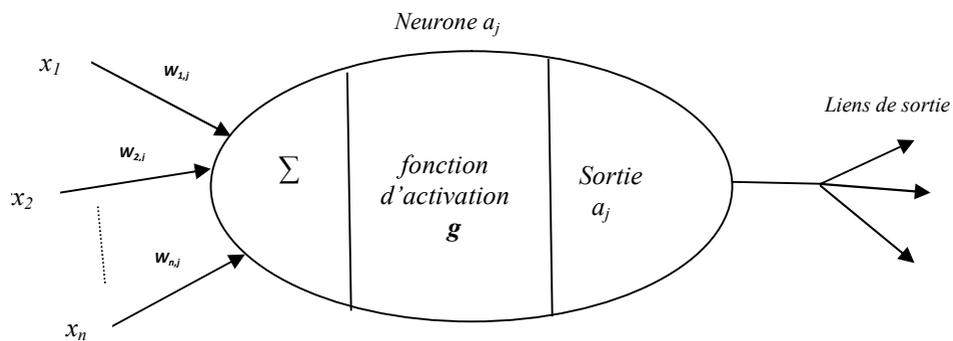


Figure 2.3: Éléments constitutifs d'un neurone artificiel.

La fonction d'activation a généralement des valeurs dans l'intervalle réel $[0,1]$, quand le réel est proche de 1, on dit que le neurone est actif alors que pour une valeur proche de 0, on dit que le neurone est inactif. L'une ou l'autre des deux valeurs réelles représentent la sortie du neurone notée sur la figure 2.3, a_j . Les fonctions d'activation peuvent être

linéaires ou non linéaires; la sortie d'un neurone par une fonction d'activation g est donnée par la formule suivante:

$$a_j = g(\text{entrées}) = g\left(\sum_{i=0}^n w_{i,j} \times x_i\right)$$

Les $w_{i,j}$ représentent les poids associés aux entrées x_i , d'un neurone, tel qu'il est illustré par la figure 2.3 et $w_{0,j}$ correspond au seuil avec son $x_0 = -1$. Le seuil est un coefficient que doit excéder la somme pondérée des entrées réelles pour que l'unité soit active.

Le comportement du réseau est déterminé par les changements apportés aux valeurs des poids synaptiques ($w_{i,j}$) ou dans la structure d'interconnexion des unités du réseau. Ce sont ces changements qui permettent au réseau d'apprendre un nouveau comportement, et la méthode utilisée pour modifier le comportement d'un réseau s'appelle règle d'apprentissage. Donc l'apprentissage par RNA consiste en une modification des poids des connexions des neurones. Ainsi un RNA prend les données, applique un ensemble de poids et une fonction de transformation, tel qu'il est illustré par la figure 2.3 pour produire un résultat. Pour la règle d'apprentissage, la plus part des algorithmes connexionnistes sont basés sur la méthode de la descente du gradient [Clément, 1989; Mitchell, 1997]. Avec cette règle, chaque connexion calcule la dérivée par rapport à son poids, d'une mesure globale de l'erreur commise par le réseau; ensuite le poids est ajusté dans le but de diminuer l'erreur.

Les algorithmes de base d'apprentissage des RNA sont le Perceptron et la Rétro-propagation ou « Backpropagation » [Clément, 1989; Hopfield, 1982]. Le premier permet d'apprendre des fonctions linéaires alors que le second permet l'apprentissage de fonctions non-linéaires. Les réseaux de neurones artificiels peuvent avoir des structures variées, généralement des graphes cycliques ou acycliques, directs ou indirects. Les RNA peuvent être constitués de plusieurs couches. Dans ce cas, les données seront considérées comme des entrées pour la première couche et seront traitées par une ou plusieurs couches intermédiaires cachées. La Backpropagation est utilisée avec des RNA à plusieurs couches. C'est l'interconnexion des unités qui permet d'avoir différentes architectures de RNA. Les plus connus sont les réseaux : à une seule couche, multicouches, récurrents et avec raccourcis [Hopfield, 1982; Mitchell, 1997]. La figure 2.4 suivante illustre ces architectures.

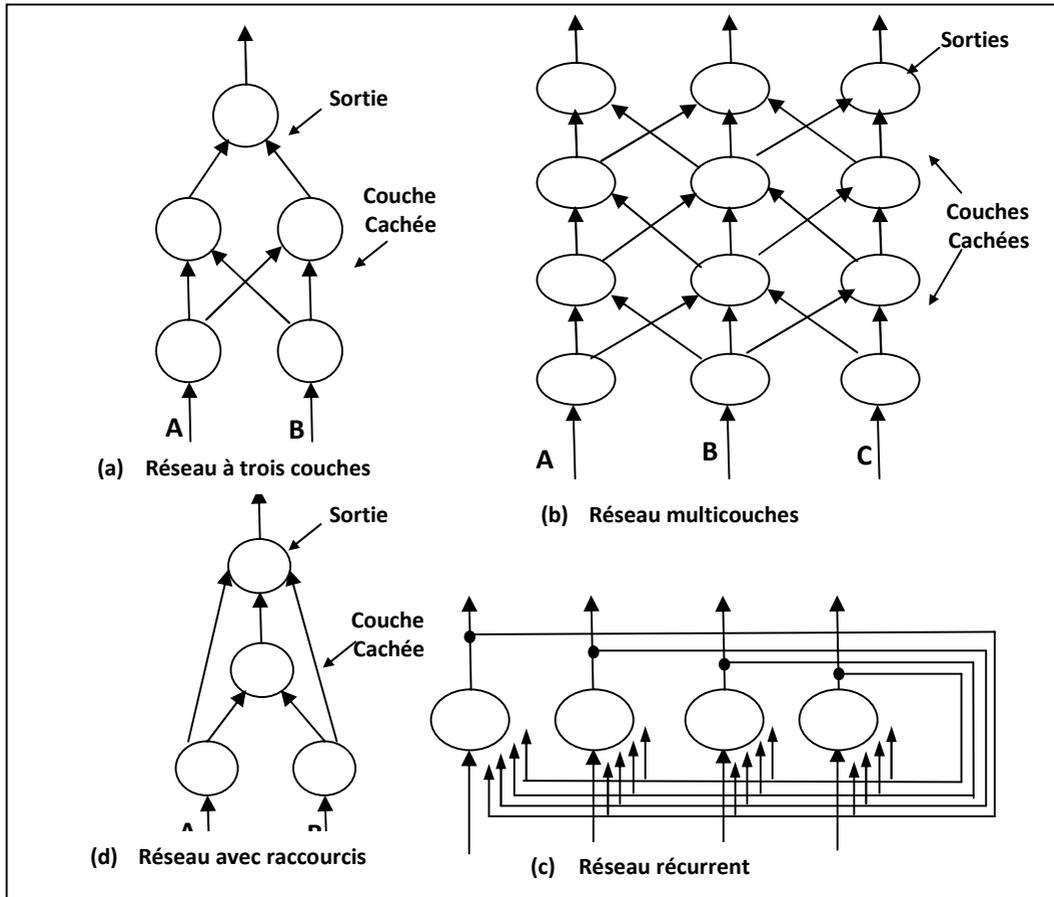


Figure 2.4: Différentes Architectures de RNA.

Les réseaux neuronaux récurrents ont l'avantage de former des topologies arbitraires car les liaisons ne sont pas toutes unidirectionnelles, et il peut y avoir des cycles. Les réseaux de neurones sont capables d'apprendre à partir de données bruitées, donc les RNA sont tolérants aux fautes telles que le manque d'un neurone. Les exemples d'entraînement soumis à un RNA peuvent avoir des valeurs réelles et continues; mais les résultats des RNA sont difficilement explicables dans la réalité. Aussi les RNA nécessitent souvent des temps d'entraînement important.

2.5.3. Apprentissage par Algorithmes Génétiques

Les mécanismes d'évolution reposent essentiellement sur la compétition qui sélectionne les individus les plus adaptés à leur milieu en leur assurant une descendance, et aussi sur une forme de coopération mise en œuvre par la reproduction. Ainsi les recherches sur l'évolution simulée ont donné naissance aux algorithmes génétiques. Ces derniers se basent sur des concepts de l'évolution biologique tels que la variation par croisement, la mutation et la sélection [Holland, 1975]. Les algorithmes génétiques produisent plusieurs solutions à un problème en générant des améliorations aux solutions existantes.

Le principe de l'algorithme génétique repose sur un codage des problèmes et de leurs solutions sous la forme de chaînes d'éléments de base, dit individu. Ce dernier est formé

par une liste de caractéristiques, dite chromosomes. Le codage des individus est généralement de la forme d'une chaîne binaire structurée, de longueur fixe ou variable selon le type de problème. On génère tout d'abord une population de solutions potentielles à un problème donné, sous la forme de telles chaînes, puis on sélectionne, au moyen d'une mesure d'ajustement, les éléments de la population satisfaisants au mieux les contraintes de la solution recherchée [Goldberg, 1994].

Pour bien utiliser un algorithme génétique, il est préalable de choisir un codage génétique représentant le problème à traiter. Le choix de ce codage est primordial car il va déterminer essentiellement les performances de l'algorithme. Les codes des individus subissent des opérateurs génétiques [Holland, 1975], dont les principaux sont:

- La reproduction, qui duplique le code d'un individu pour en créer un nouveau.
- Le croisement, qui mélange les chromosomes des individus parents pour créer le code génétique d'un individu fils.
- La mutation, qui apporte une modification aléatoire au code génétique d'un individu.
- La sélection, qui choisit dans une population les individus qui survivront à la génération suivante, en fonction d'une valeur d'adaptation.
- L'évaluation, qui mesure le degré d'adaptation d'un individu à son environnement.

La figure 2.5 suivante présente le principe général des algorithmes génétiques.

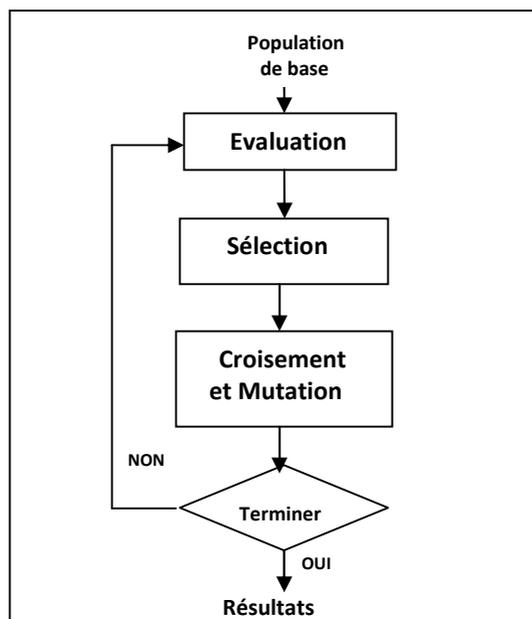


Figure 2.5: Principe général des algorithmes génétiques.

Tel qu'il est illustré par la figure 2.5, les étapes d'un algorithme génétique se résument alors à:

- Sélectionner les individus les mieux adaptés, en:
 1. évaluant le degré d'adaptation de chaque individu à son environnement,

2. sélectionnant des paires d'individus en fonction de leur valeur d'adaptation,
- Appliquer un opérateur de diversification de la population:
 1. l'opérateur de croisement ou,
 2. l'opérateur de mutation.
 - Développer la nouvelle génération, puis recommencer le cycle

Les algorithmes génétiques sont des techniques d'exploration, qui ont une capacité de s'adapter et de se transformer sans avoir à imposer beaucoup de restrictions sur la recherche d'une solution, ce qui leur permet de mieux explorer le problème et de trouver des solutions originales [Goldberg, 1994].

Le choix du codage des individus d'une population est un élément critique dont dépend grandement l'efficacité d'un algorithme génétique. Aussi la fonction d'adaptation, répondant à un problème donné, conditionne pour une grande part cette efficacité. Les algorithmes génétiques sont appropriés aux problèmes d'optimisation effectuant des recherches dans un grand espace d'objets candidats, et sélectionnant le plus adapté en se basant sur la fonction d'ajustement. Mais les algorithmes génétiques souffrent d'explosions combinatoires pour les problèmes de taille importante. Ces dernières années, des méthodes destinées aux problèmes d'optimisation difficiles, dites des méta heuristiques, sont de plus en plus étudiées [Iredi et al., 2001]. Ces méthodes s'inspirent de la vie courante et on y retrouve des inspirations de la physique, des colonies de fourmis, des vols d'oiseaux, on parle alors de bioinformatique ou de vie artificielle. Les recherches en vie artificielle tentent de comprendre la vie par abstraction des principes fondamentaux de la dynamique sous-jacents aux phénomènes biologiques, et la recreation de ces dynamiques en utilisant d'autres moyens physiques comme la machine.

2.5.4. Apprentissage par Programmation Logique Inductive

La dénomination "Programmation Logique Inductive" (*PLI*) est due à Muggleton [Muggleton, 1991]; la *PLI* est née du croisement de l'apprentissage automatique et de la programmation logique. La programmation logique inductive synthétise des hypothèses ou de nouvelles connaissances à partir de données exprimées en logique du premier ordre, appelée aussi logique des prédicats [Mitchell, 1997; Muggleton, 1991]. Les hypothèses sont aussi exprimées dans cette logique qui permet de représenter des propositions avec des variables. La *PLI* est donc une méthode d'apprentissage permettant d'apprendre des règles « if then » contenant des variables, dites aussi des « clauses de Horn de premier ordre ». Le principe de la programmation logique inductive est le suivant [Lavrac et Dzeroski, 1994]:

Les Entrées: Trois ensembles, B , P et N , nommés l'interprétation, constitués chacun par des clauses de Horn avec:

- B : connaissances du domaine, dites aussi le *Background*.

- P : exemples d'entraînement Positifs.
- N : exemples d'entraînement Négatifs.

Sortie: Trouver une hypothèse H sous forme d'un ensemble de clauses de Horn et tel que les propriétés suivantes soient satisfaites:

- La Complétude: $\forall e \in P, H \cup B \models e$
- La Consistance: $\forall e \in N, H \cup B \not\models e$

$\forall e \in P, H \cup B \models e$, se lit $\forall e$ un exemple d'entraînement appartenant à l'ensemble des exemples positifs P , e est satisfait ou couvert par l'hypothèse H et le background B . Cette propriété est dite la complétude, car l'hypothèse H doit être complète en couvrant tous les exemples de P .

Inversement, $\forall e \in N, H \cup B \not\models e$, se lit $\forall e$ un exemple d'entraînement appartenant à l'ensemble des exemples négatifs N , e n'est pas satisfait ou n'est pas couvert par l'hypothèse H et le background B . Cette propriété dénote la consistance de l'hypothèse H qui ne doit couvrir aucun exemple négatif.

La programmation logique inductive tente donc de trouver une hypothèse H qui permet d'expliquer au mieux les exemples positifs, tout en rejetant les exemples négatifs. En prenant en charge le background avec un apprentissage inductif, la *PLI* tire profit des connaissances disponibles. La recherche de l'hypothèse H s'effectue par l'utilisation des techniques liées à la programmation logique comme la substitution, la spécialisation, la généralisation, l'unification et la résolution [Lavrac et Dzeroski, 1994]. Parmi les techniques les plus connues de la *PLI*, l'algorithme FOIL [Quinlan, 1990]. FOIL est un algorithme qui apprend des règles «if then», il est capable d'introduire des variables, et d'exprimer la récursivité.

Avec la *PLI*, les clauses de Horn permettent d'avoir des variables dans les pré conditions et dans les post conditions des règles. Ainsi, les règles if then, apprises par les algorithmes basés sur les principes de la programmation logique inductive sont plus expressives que les règles if then propositionnelles. Il est aussi faisable d'utiliser les mêmes prédicats dans les prémisses et les conséquences; ainsi l'apprentissage de règles récursives est possible. Les méthodes d'apprentissage de règles if then de premier ordre sont d'une grande importance pour d'une part la richesse des expressions de leur formalisme de représentation et d'autre part pour l'exploitation des connaissances du domaine.

2.5.5. Apprentissage basé sur l'Analogie

L'apprentissage par analogie dérive des recherches en psychologie cognitive traitant du raisonnement analogique. Se sont des méthodes de résolution de problèmes qui consistent à mettre en relation une situation connue, appelée la source, avec une nouvelle situation, dite la cible; en se basant sur le critère d'analogie [Aha et al., 1991]. Le fondement de l'apprentissage par analogie repose sur les quatre étapes suivantes [Kolodner, 1993]:

1. *Encodage*: l'encodage ou la représentation de la source A et de la cible B selon le même formalisme pour pouvoir les traiter. Plus tard les représentations sont soit disponibles en mémoire, soit provenant d'un traitement des informations tirées de l'environnement.
2. *Inférence*: inférer une relation ($R1$) entre les représentations de A et de B tel que la source A soit directement liée à la cible B par cette relation.
3. *Mise en correspondance* (le mapping): c'est inférer une relation d'ordre plus élevé ($R2$) du groupe $A-B$ susceptible de correspondre à un autre groupe $C-D$ par analogie. L'appariement entre la source et la cible s'effectue sur la base de traits communs entre les deux situations. Il peut s'agir de traits dits de surface, comme le contexte des deux situations, ou des traits profonds faisant appel à la sémantique des situations.
4. *Application*: C'est le transfert de certains aspects de la source vers la cible pour la résolution. C'est projeter la relation $R1$ sur le terme C pour en déduire le terme D ou sa nature.

Ainsi, au niveau de l'apprentissage basé sur l'analogie, la classification est faite par similarité aux exemples déjà mémorisés [Aha et al., 1991]. La notion de similarité est très importante, et l'apprentissage par analogie est aussi dénommé apprentissage fondé sur la similarité. Les méthodes d'apprentissage par analogie se caractérisent [Mitchell, 1997] par:

- La classification est basée sur la similarité des exemples.
- Les instances sont représentées comme des points dans un espace euclidien.
- La classification ou la généralisation est différée jusqu'à ce qu'une nouvelle instance se présente; elles sont dites « lazy » méthodes.

La méthode la plus connue, fondée sur l'analogie, est dite K plus proches voisins «k-nearest-neighbor» [Mitchell, 1999]. Au niveau du, k-nearest-neighbor, les instances x sont représentées comme des points dans un espace euclidien, donc par un vecteur de coordonnées $(a_1(x), a_2(x), a_3(x), \dots, a_n(x))$ avec $a_i(x)$ dénote la valeur du $i^{\text{ème}}$ attribut de l'instance x . Et la similarité entre deux instances x_i et x_j se calcule par la mesure de la distance euclidienne entre deux points, $d(x_i, x_j)$ qui est définie par la formule suivante:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (a_k(x_i) - a_k(x_j))^2}.$$

Le Case Based Reasoning (CBR) fondé aussi sur l'analogie est capable d'utiliser la connaissance spécifique contenue dans son expérience passée, pour résoudre les nouveaux problèmes. Cette expérience est représentée sous la forme de cas ou de prototypes. Ces cas, qui ont été assignés par l'expert aux classes auxquelles ils appartiennent, constituent ainsi la mémoire d'un système CBR [Kolodner, 1993]. Les cas possèdent une représentation symbolique plus riche que celle des points dans un espace euclidien, en effet ils sont souvent représentés par des frames ou des schémas. Quand un nouveau problème est présenté à un système CBR, il va se rappeler des cas stockés dans sa mémoire, similaires au problème courant. Ensuite, le système adapte la meilleure solution mémorisée et la transfère au problème actuel. Le cas nouveau, qui a été traité par le système et reconnu, est à son tour mémorisé et donc ajouté comme une nouvelle expérience. En général, un système CBR fonctionne selon les phases illustrées par la figure 2.6.

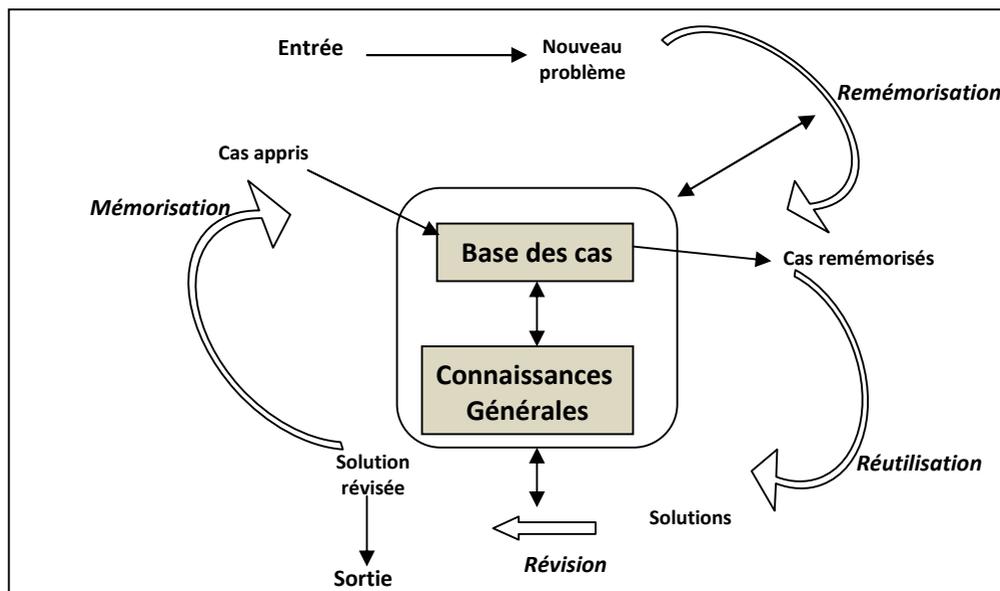


Figure 2.6: Apprentissage par analogie.

Les phases d'un CBR sont donc:

1. *Remémoration* des cas les plus similaires par rapport au cas entré comme nouveau problème.
2. *Réutilisation* de la connaissance du (ou des) cas remémoré(s) pour le nouveau cas.
3. *Révision* de la solution donnée afin de la valider.
4. *Mémorisation* de la nouvelle expérience, pour une utilisation future.

Avec les méthodes d'apprentissage par analogie, les informations contenues dans les exemples d'entraînement ne sont jamais perdues, puisque tous les exemples sont mémorisés. L'inconvénient est que la classification n'est pas faite au préalable mais durant la requête; alors le temps de réponse peut être élevé.

Ainsi nous venons de présenter les principales méthodes représentatives de l'approche d'apprentissage inductive. Nous notons aussi qu'il existe des possibilités de combinaison entre les méthodes d'apprentissage précédentes. Nous citons en exemple les méthodes symboliques-floues qui combinent des algorithmes à représentation de connaissances symbolique, tels que : les arbres de décisions et les algorithmes génétiques avec la logique floue. Le but de ces méthodes hybrides est la prise en charge des connaissances incertaines, avec la logique floue, via des algorithmes bien connus de l'apprentissage de concept. Aussi les méthodes hybrides qui combinent des réseaux de neurones avec de la logique floue ou le CBR [Nauck et Kruse, 1996; Sun, 1992]. Le but est de résoudre les problèmes d'une méthode par les apports d'une autre. Les méthodes inductives sont nombreuses et chacune peut comporter plusieurs algorithmes conçus selon le principe de la méthode mais différents en termes de certains facteurs liés surtout à l'efficacité et à la capacité de prise en charge de données discrètes ou continues ainsi qu'au traitement des données bruitées.

2.6. Méthodes d'Apprentissage Déductives

Les méthodes d'apprentissage déductives ou analytiques ne sont pas aussi nombreuses que les méthodes inductives; elles se basent essentiellement sur la théorie du domaine ou les connaissances préalables, dans leur processus d'apprentissage. Donc en plus de la stratégie d'inférence qui distingue les méthodes inductives des méthodes déductives, ces dernières possèdent une entrée en plus, des exemples d'entraînement E , dite la théorie du domaine ou le «Background knowledge: B » [Michalski et al., 1993; Mitchell, 1997; Russel et Norvig, 2006]; et tel que les connaissances apprises doivent être consistantes avec B et E . Au niveau de la programmation logique inductive, nous avons défini la consistance; cette propriété indique que les connaissances apprises sont satisfaites par l'interprétation, correspondant à B et E . La programmation logique inductive combine à son principe inductif les connaissances préalables et tire profit de ces dernières dans son processus d'apprentissage.

Le principe général des méthodes déductives est l'apprentissage basé sur l'explication. Les méthodes d'apprentissage fondées sur l'explication considèrent que les connaissances préexistantes sont à la fois correctes et complètes, elles sont aussi dites parfaites. Une théorie de domaine est parfaite si elle est complète et correcte, c'est-à-dire qu'elle couvre tous les exemples positifs de l'ensemble des instances, et que toutes ses assertions sont vraies [Mitchell et al., 1986]. Les méthodes analytiques servent donc à améliorer les performances du système, grâce à des traitements qui rendent l'utilisation des connaissances préalables plus efficace.

2.6.1. Apprentissage basé Explication

Les méthodes d'apprentissage basées sur l'explication (Explanation Based Learning: EBL) [Dejong et Mooney, 1986; Mitchell et Thrun, 1993] utilisent des connaissances préexistantes et un raisonnement déductif. Dans l'EBL, les connaissances du domaine sont utilisées pour analyser et expliquer comment chaque exemple d'entraînement satisfait le concept cible [Mitchell et al., 1986]; en effet chaque nouveau exemple est traité de la manière suivante:

- Explication du concept cible à travers l'ensemble d'exemples d'entraînement E et la théorie du domaine B , pour avoir une règle plus ou moins générale.
- Analyse du résultat de l'explication pour omettre les propriétés non pertinentes de l'exemple et produire une règle plus générale.
- Raffinement des règles apprises.

D'après les trois étapes précédentes, les méthodes EBL travaillent selon le cycle illustré par la figure suivante:

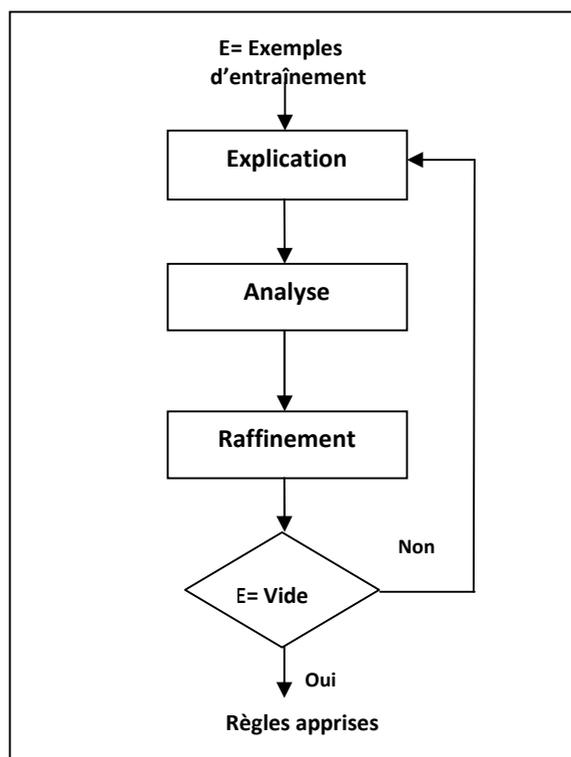


Figure 2.7: Principe des méthodes EBL.

L'algorithme représentatif de l'EBL est le Prolog-EBG, qui travaille sur des théories de domaine parfaites. Cet algorithme utilise les clauses de Horn pour la représentation de la théorie du domaine utilisée ainsi que la représentation du concept cible à apprendre [Kedar-Cabelli et McCarty, 1987]. Une explication dans cet algorithme est une preuve Prolog; d'où la nomination Prolog-EBG. Cet algorithme se base d'une part, sur une

représentation et des preuves logiques et d'autres part, sur une généralisation basée sur l'explication «Explanation Based Generalization : EBG».

Dans les méthodes EBL la généralisation est faite selon un raisonnement logique et non statistique, comme c'est le cas pour l'apprentissage de concepts. Aussi l'apprentissage basé explication, permet une généralisation rationnelle des exemples en se servant des théories de domaines. Un aspect important de l'EBL est sa capacité à formuler des caractéristiques nouvelles non explicites dans les exemples d'entraînement. Mais le problème est dans la contrainte des théories de domaine parfaites. Rappelons qu'une théorie de domaine est parfaite si elle est complète et correcte, c'est-à-dire qu'elle couvre tous les exemples positifs de l'ensemble des instances, et que toutes ses assertions sont vraies. Or en réalité les théories de domaines, lorsqu'elles existent, elles ne sont pas toujours parfaites.

2.7. Méthodes d'Apprentissage par Renforcement

L'approche d'apprentissage par renforcement consiste à apprendre quoi faire, comment associer des actions à des situations, afin de maximiser quantitativement une récompense [Dietterich, 2002 ; Sutton et Barto, 1998; Sutton et McCallum, 2006]. Ainsi les méthodes d'apprentissage par renforcement proposent des fonctions valeurs d'association des situations aux actions tout en visant la maximisation des récompenses reçues. La méthode la plus connue est le Q-learning.

2.7.1. Le Q-Learning

Avant de présenter le Q-learning, nous devons reconsidérer quelques notions de base et définir certaines notations. Rappelons que l'apprentissage par renforcement permet d'apprendre une politique de contrôle. La politique désirée est celle qui à partir de n'importe quel état initial, choisit les actions qui maximisent les récompenses accumulées [Mitchell, 1997; Sutton et Barto, 1998].

Le système apprenant existe dans un environnement décrit par une succession d'états (States: S) et il peut effectuer un ensemble d'actions A . A tout moment t où le système effectue une action $a_t \in A$, il reçoit une valeur réelle (reward ou récompense) r_t indiquant le degré d'adéquation de cette transition état \rightarrow action : $a_t \rightarrow s_t$.

Le rôle du système apprenant est d'apprendre une politique de contrôle P , tel que:

$$P: S \rightarrow A \quad / \quad P \text{ maximise } (\sum r_t)$$

Le Q-learning [Watkins et Dayan, 1992] est un algorithme d'apprentissage par renforcement qui permet de déterminer une fonction valeur Q tel que $Q: S \rightarrow A$ et $Q(s; a)$ correspond à la récompense attendue lorsque l'action a est exécutée pour l'état s . Le Q-learning apprend les qualités associées aux couples $(s; a)$, c'est-à-dire il cherche à trouver

la meilleure action a pour chaque état s à travers l'expérience tout en considérant des récompenses à long terme.

Cette méthode consiste donc à évaluer la Q fonction dynamiquement par l'expérience du système dans l'environnement. En effet le principe du Q-learning est de construire itérativement la fonction Q pour chaque couple $(s; a)$ en expérimentant les actions dans l'environnement. A chaque choix d'une action à effectuer, le système choisit une action a dans l'ensemble A , exécute a , reçoit une récompense $r(s; a)$ et observe son nouvel état s' . Il met alors à jour la valeur $Q(s; a)$ en corrigeant sa valeur précédente en fonction de la récompense obtenue $r(s; a)$ et de la récompense future que le système espère obtenir dans le nouvel état s' . La mise à jour de la fonction valeur Q se fait sur la base de l'observation des transitions instantanées et de leurs récompenses associées. Le Q-learning est un algorithme qui ne nécessite pas un modèle de l'environnement et permet d'apprendre une politique d'actions.

2.8. Méthodes d'Apprentissage Hybrides

Les méthodes d'apprentissage hybrides permettent de tirer profit des approches: inductive, déductive et par renforcement tout en palliant les déficiences de chacune d'elles. Ainsi, il existe des méthodes hybrides inductive ou déductive avec un apprentissage par renforcement [Dietterich et Flann, 1995]. Pour les approches: inductive et déductive, la première approche nécessite un grand nombre d'exemples d'entraînement et la seconde repose sur des théories de domaine parfaites. En réalité il existe des problèmes avec des théories approximatives, ainsi pour ces problèmes, les méthodes purement déductives ne peuvent s'appliquer. La solution est de combiner les principes des deux approches inductive et déductive; une méthode représentative pour ce type d'apprentissage est la suivante.

2.8.1. Les Réseaux KBANN

Les réseaux KBANN (Knowledge Based Artificial Neural Networks) développés par [Towel et Shavlik, 1994] se basent sur une méthode qui permet la compilation, le raffinement et l'extraction de règles symboliques en utilisant des Réseaux de Neurones Artificiels (RNA). Les réseaux KBANN sont le résultat de l'implémentation de trois algorithmes:

1. le premier algorithme est responsable de la traduction de règles symboliques approximativement correctes, donc les règles d'une théorie approximative, vers un RNA multicouches;
2. le deuxième algorithme fait le raffinement des connaissances du réseau par apprentissage à partir d'une base d'exemples en utilisant la Rétro-propagation;

3. le troisième algorithme réalise l'extraction des règles raffinées à partir du réseau qui a été entraîné.

Les KBANN permettent donc de construire un réseau de neurones initialement conforme à la théorie du domaine considéré, puis ils utilisent l'algorithme de back-propagation pour la révision des poids du réseau. L'idée est que même si la Théorie du Domaine (TD) est approximativement correcte, l'initialisation du réseau tel qu'il satisfait la TD peut donner une meilleure solution que d'initialiser le réseau aléatoirement.

L'extraction des règles raffinées, à partir du réseau entraîné, se base sur l'idée de trouver les combinaisons des valeurs d'entrée d'une unité qui font que son activation de sortie soit proche de 1. Cette combinaison d'entrées nécessaires à l'activation d'une unité va être exprimée sous la forme d'une règle symbolique. Les méthodes d'extraction cherchent à étudier les relations établies entre l'ensemble des entrées et la valeur du seuil d'activation d'une unité [Mitchell, 1997; Towel et Shavlik, 1994]. Une valeur résultant de la somme pondérée des entrées d'une unité supérieure au seuil implique que cette unité soit activée, et ainsi la méthode d'extraction exprime cette relation entre les entrées et l'activation de la sortie par une règle. Une valeur résultant de la somme pondérée des entrées d'une unité inférieure au seuil implique que cette unité ne va pas être activée, et par conséquent la méthode d'extraction, ne va pas exprimer, par une règle, la relation entre les entrées et la sortie de cette unité.

Les KBANN sont des réseaux basés sur une méthode d'intégration des connaissances du domaine avec les RNA inductifs, ainsi ils permettent de tirer profit des deux approches. En effet d'une part l'apprentissage des réseaux KBANN peut être fait à partir d'une base d'exemples réduite, étant donné que le réseau est initialisé par la théorie du domaine. D'autre part la théorie du domaine peut contenir des connaissances incomplètes. Un autre avantage est l'extraction de règles à partir du réseau, ce qui rend ces résultats facilement interprétables.

2.9. Conclusion

Vu la grande diversité et la variété des méthodes de l'apprentissage automatique, dont les premières, datent des années quarante, nous avons dans ce chapitre procédé à leur classification dans des approches globales. Ceci nous a permis de faire face à la pléthore de méthodes existantes. En conséquence, les approches d'apprentissage automatique dégagées: inductive, déductive, par renforcement et hybride ont été présentées. Chacune de ces approches se base sur un principe d'apprentissage, l'approche inductive est fondée sur l'induction et synthétise les connaissances à partir d'un ensemble étendu d'exemples d'entraînement. Alors que l'approche déductive est basée sur la déduction de connaissances à partir d'un ensemble d'exemples d'entraînement et une théorie de domaine parfaite. L'approche par renforcement permet d'apprendre des politiques

d'actions pour faire correspondre la bonne action à la situation actuelle. L'hybridation des approches précédentes permet de tirer profit de chacune des approches de base; mais, parfois, elle reste difficile à mettre en œuvre

Les méthodes d'apprentissage automatique les plus représentatives des différentes approches ont été présentées. En réalité les méthodes d'apprentissage sont nombreuses et très variées; mais notre objectif n'est pas de recenser toutes les méthodes existantes ou d'étudier une méthode particulière pour son application ou l'optimisation de ses paramètres. Notre objectif était l'étude des approches d'apprentissage ainsi que des méthodes les plus représentatives de ces dernières. Cette étude est un support pour l'établissement d'une mise en correspondance entre les approches/méthodes étudiées d'une part et les types d'Agent nécessitant des capacités d'apprentissage d'autre part. Cette mise en correspondance sera présentée au chapitre 3.

L'étude des principes et caractéristiques des différentes approches et méthodes de l'apprentissage automatique nous a aussi servi de base théorique pour le modèle général et générique, établi pour les systèmes basés Agent-Apprenant, que nous présenterons au chapitre 4.

Chapitre 3

Mise en Correspondance entre Agents Apprenants et Approches/Méthodes d'Apprentissage

3.1. Introduction

Selon les objectifs tracés et présentés au chapitre 1, pour notre travail de thèse, le chapitre en cours a pour finalité l'établissement d'une mise en correspondance entre les types d'Agents-Apprenants et les approches/méthodes d'apprentissage automatique. Pour arriver à cette finalité, des sous objectifs doivent être atteints.

Avec la pléthore des méthodes d'apprentissage automatique, le chapitre 2 a partiellement participé à la réalisation de cette finalité en réduisant la grande variété de ces méthodes par leur catégorisation en approches bien définies. Ce qui a permis l'étude des approches dégagées ainsi que les méthodes les plus représentatives pour chacune de ces approches. Ainsi pour ce chapitre, le travail préalable est la détermination des différents types d'Agents nécessitant le recours à l'apprentissage automatique; donc les Agents-Apprenants. La définition de ces derniers est dégagée d'une typologie d'Agents basée sur un ensemble de critères de classification préalablement définis.

Ce chapitre commence donc par présenter: les critères de classification d'Agents et les Agents-Apprenants déterminés, pour aboutir à la mise en correspondance entre les Agents-Apprenants et les approches/méthodes d'apprentissage automatique. Cet appariement fait correspondre aux différents types d'Agents-Apprenants dégagés les approches/méthodes appropriées parmi les quatre catégories: inductive, déductive, par renforcement et hybride. Des critères de choix, surtout pour les méthodes inductives, sont présentés et la mise en correspondance est appuyée par des exemples de domaines d'applications réels.

3.2. Critères de Classification d'Agents

La classification d'Agents, dans notre travail n'est pas une finalité en soi, notre but est de dégager les types d'Agents concernés par l'apprentissage automatique. Ainsi les critères de classification nous ont servi pour dégager les différents types d'Agents Apprenants. Le but principal visé, étant la mise en correspondance entre les approches/méthodes de l'apprentissage automatique, étudiées au chapitre 2, et les types d'Agents-Apprenants à déterminer dans ce chapitre. Dans la littérature, la classification d'Agents la plus répandue est celle correspondant aux trois approches d'Agents, présentées au chapitre 1, impliquant trois classes d'Agents: réactifs, cognitifs et hybrides [Ferber, 1995; Hayes-Roth, 1995; Jennings et al., 1998; Meas, 1990; Müller et al., 1995; Müller, 1996; Rao et Georgeff, 1995; Wilkins, 1988; Wooldrige et Jennings, 1995; Wooldridge, 2000]. Cette classification est une base, au niveau de la technologie Agent, sur laquelle peuvent être fondées des classifications plus raffinées.

Pour la détermination des différents types d'Agents-Apprenants; une classification d'Agents, avec un focus sur l'apprentissage automatique et la tendance individuelle des Agents, est nécessaire. Rappelons que cette tendance considère un Agent comme une entité autonome agissant en isolation des autres Agents artificiels, elle est centrée sur le fonctionnement interne de l'Agent individuel. Le but, donc est de recenser les différentes catégories d'Agents afin de pouvoir cerner les types d'Agents nécessitant le recours à l'apprentissage automatique. Une typologie d'Agents se base sur des critères de classification, c'est ce que nous devons présenter au préalable.

Après une étude, analyse et synthèse de plusieurs travaux de recherche, concernant la technologie Agent, nous avons énuméré un certain nombre de critères que nous avons considéré pour classer les Agents; ces critères ou caractéristiques sont:

- *L'Autonomie.*
- *La Réactivité.*
- *La Pro-activité.*
- *L'Intelligence.*
- *La Flexibilité.*
- *L'Adaptation.*
- *L'Apprentissage automatique.*

- *Les Services de l'Agent.*
- *La Mobilité.*

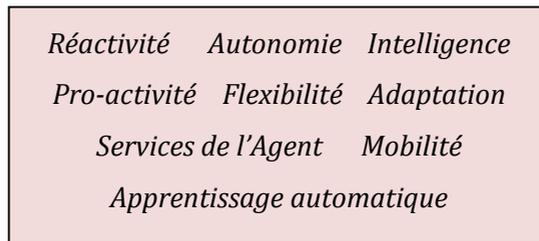


Figure 3.1: Critères de classification d'Agents.

Certains de ces critères ont été présentés comme caractéristiques essentielles d'un Agent, au chapitre 1, ils nous permettent aussi de distinguer les différents types d'Agents; et par conséquent ceux nécessitant le recours à l'apprentissage automatique.

3.2.1. L'Autonomie

L'autonomie est la caractéristique de base des Agents et généralement elle est utilisée pour les définir; c'est la propriété majeure qui fait la différence entre un Agent et un objet, tel que nous l'avons expliquée au chapitre 1. Cette caractéristique signifie qu'un Agent est capable d'entreprendre des actions ou des initiatives sans intervention directe humaine ou celle d'autres systèmes. En effet selon [Guessoum et Briot, 1999; Meas, 1990; Rao et Georgeff, 1995; Wooldridge et Jennings, 1995] l'autonomie est la capacité pour un Agent d'opérer de manière autonome sans une intervention directe d'humains ou d'autres Agents et de contrôler ses actions et son état interne. Pour [Cohen et Levesque, 1990; Shoham, 1993], l'autonomie signifie qu'un Agent, possédant ses propres buts, est capable d'agir selon ses intentions en adoptant certains buts et en abandonnant d'autres afin d'atteindre son objectif final.

L'autonomie d'un Agent peut être partielle ou plus ou moins complète, certaines caractéristiques peuvent améliorer le degré d'autonomie d'un Agent. Nous pensons, que l'autonomie d'un Agent peut être fortement renforcée par l'apprentissage automatique.

3.2.2. La Réactivité et la Pro-activité

La réactivité est le pouvoir de perception et de réaction à temps aux changements qui affectent le perçu. C'est donc la capacité de modification du comportement, de l'entité réactive, suite aux changements des conditions de l'environnement; ainsi le comportement est décrit par des mécanismes directs associant les perceptions aux réactions [Ferber, 1997]. Pour un bon fonctionnement, la réactivité nécessite une perception entière et totale de l'environnement.

La pro-activité est une propriété plus complète que la réactivité, en effet elle permet d'avoir un comportement opportuniste qui n'est pas une simple réaction à

l'environnement; c'est la capacité de prendre des initiatives en exhibant un comportement orienté buts ou objectifs [Meas, 1990; Rao et Georgeff, 1995; Wooldridge et Jennings, 1995]. Un Agent s'occupant de l'optimisation du routage dans un réseau peut décider, de lui-même, pendant un temps où il est non actif, de faire des statistiques sur les routeurs pour améliorer son activité future, c'est un cas typique de la pro-activité. L'apprentissage automatique peut considérablement améliorer l'anticipation et la pro-activité.

3.2.3. L'Intelligence

L'intelligence a principalement trait à l'action rationnelle qui permet à une entité d'atteindre la meilleure solution possible. En d'autres termes la rationalité, principe central de l'intelligence artificielle, considère qu'un système intelligent est descriptible en termes de buts, d'actions et de connaissances; et qu'il utilise ses connaissances pour déterminer quelles actions choisir afin d'atteindre ses buts [Igor, 1987; Nilsson, 1980; Russell et Norvig, 2006]. Ainsi, l'intelligence peut être vue comme la capacité à élaborer des inférences correctes permettant de déterminer les actions qui mèneront aux buts visés. Dans ce sens l'intelligence est liée à la pro-activité, en effet un système intelligent utilise des méthodes heuristiques et emprunte des voies non déterministes pour la résolution de problèmes [Haton, 1990; Hayes-Roth, 1995]. L'anticipation ou la pro-activité ne peuvent être atteintes, par un système, que s'il est doté d'un haut degré d'intelligence.

L'intelligence, pour nous, est essentiellement la capacité à résoudre des problèmes ne possédant pas une solution préalablement bien définie, c'est aussi la possibilité d'adaptation à l'environnement. Par conséquent l'intelligence est fortement liée à l'apprentissage automatique, ceci découle des problèmes qu'elle tente de résoudre. Nous considérons aussi que l'intelligence contribue considérablement au renforcement de l'autonomie; en effet une entité capable de trouver, toute seule, la meilleure solution possible grâce aux techniques intelligentes qu'elle intègre, peut être qualifiée d'autonome.

3.2.4. La Flexibilité

La flexibilité est la capacité de s'adapter, c'est donc la possibilité de changer de comportement selon le perçu. D'après Jennings et son équipe de recherche [Jennings et al., 1998], la flexibilité est une propriété importante distinguant les Agents des systèmes classiques et ils considèrent qu'elle englobe les trois propriétés suivantes:

1. Réactivité.
2. Pro-activité.
3. Sociabilité.

Ainsi définir la flexibilité revient à définir ces trois propriétés, la réactivité et la pro-activité ont été déjà définies. La sociabilité pour un système, c'est l'habilité à interagir avec d'autres systèmes ou humains afin d'atteindre ses propres objectifs et aider les autres dans l'accomplissement de leurs activités [Wooldridge et Jennings, 1995]. Pour un Agent, la

sociabilité est sa capacité à s'intégrer dans un environnement peuplé d'Agents avec qui il échange des messages pour accomplir un but global. La sociabilité est donc une propriété liée aux sociétés d'Agents et à la tendance sociale des Agents. Ainsi, pour un Agent à tendance individuelle, la flexibilité englobe la réactivité et la pro-activité.

3.2.5. L'Adaptation

L'adaptation ou l'adaptabilité est la capacité d'un système à s'adapter à l'environnement. Un Agent est adaptatif s'il est capable de contrôler ses aptitudes communicationnelles et comportementales selon l'environnement dans lequel il évolue et interagit [Jennings et al., 1998; Meas, 1990; Müller, 1996]. L'adaptation permet donc à un système de satisfaire ses objectifs fondamentaux selon son environnement.

L'adaptation comportementale peut être statique ou dynamique, elle est dite statique si un Agent est doté de règles de comportements dès sa conception [Guessoum et Briot, 1999; Meas, 1990]. L'adaptation est dite dynamique si les règles de comportement sont construites dynamiquement au fur et à mesure que l'environnement change. En effet, l'adaptation dynamique est la capacité de l'Agent à s'adapter aux changements imprévus de son environnement par son évolution continue; ce type d'adaptation est réalisable par des capacités d'apprentissage automatique.

Nous constatons que plusieurs propriétés, telles que l'intelligence, la pro-activité, la flexibilité et l'adaptation, sont fortement corrélées et chacune fait référence à l'autre. En effet, il existe une complémentarité entre toutes ces caractéristiques, ce qui peut donner une impression de redondance au niveau de leurs définitions.

3.2.6. L'Apprentissage Automatique

L'apprentissage automatique est la capacité à améliorer les performances au fur et à mesure de l'exercice d'une activité. Il permet de construire un modèle de la réalité à partir de données et d'expériences. Selon [Cornuéjols et Miclet, 2002; Michalski et Chilausky, 1980] l'apprentissage consiste à construire et à modifier des représentations ou des modèles à partir d'une série d'expériences. Il consiste aussi à améliorer les performances d'un système pour des tâches données à partir d'expériences [Mitchell, 1997; Mitchell, 2006]. C'est une propriété qui fournit aux systèmes la capacité d'acquérir la compréhension de certains comportements au cours du temps, sans nécessiter que ces comportements soient programmés manuellement [Mitchell et al., 1983]. L'apprentissage permet, entre autres, à un système d'opérer dans un environnement inconnu (ou partiellement connu) au départ et d'apprendre, à partir de son expérience, à s'y adapter, à devenir plus compétent et par conséquent améliorer sa performance.

A notre avis, l'apprentissage automatique est d'une grande importance pour un Agent, puisqu'il lui permet d'atteindre les caractéristiques précédemment définies, en effet il assure à un Agent d'avoir plus: d'autonomie, de flexibilité, d'intelligence, d'agir dans un environnement inconnu et de s'y adapter. L'apprentissage automatique permet à un Agent d'améliorer ses compétences et de devenir meilleur avec le temps.

3.2.7. Les Services de l'Agent

Les services de l'Agent sont nombreux, mais avec le taux croissant des informations électroniques et leur large diffusion, la recherche d'informations et la navigation sont parmi les services les plus demandés [Jennings et Wooldridge, 1998; Zanker et Jessenitschnig, 2009]. Pour le service de recherche d'informations, il existe différentes fonctions, les plus importantes sont:

- Recherche d'informations: qui peut se faire de manière intelligente par l'utilisation de moteurs de recherche perfectionnés, d'outils d'analyse linguistique des requêtes ou par exploration de liens hypertextes à partir des adresses de pages web.
- Analyse des informations récupérées: indexation sémantique des résultats, résumé automatique.
- Filtrage, édition, archivage, mise à jour de résultats.
- Navigation parmi des pages ou des sites web téléchargés.

Ainsi les tâches importantes que les Agents prennent en charge peuvent être utilisées pour les classer. Et vu l'explosion combinatoire des informations électroniques, l'utilisation massive de l'internet et sa prolifération dans la vie quotidienne, le service de recherche et de filtrage d'informations est de plus en plus important.

3.2.8. La Mobilité

La mobilité est un concept des applications réparties, qui en association avec la technologie Agent ouvre de nouvelles perspectives de conception de systèmes distribués. Ce paradigme est lié à des concepts classiques tels que : l'appel de procédure à distance (le Remote Procedure Call: RPC), l'invocation d'objet à distance et le code à la demande (Code On Demand: COD) [Carzaniga et al., 1997]. La mobilité est d'un grand intérêt pour la mise en œuvre d'applications dont les performances dépendent de la disponibilité et de la qualité des services et des ressources. Le concept de mobilité a beaucoup évolué avec celui du concept d'Agent mobile qui sera défini ultérieurement.

3.3. Détermination des Types d'Agents-Apprenants

L'objectif de la définition des propriétés précédentes est le recensement des différents types d'Agents, ceci pour pouvoir déterminer les types d'Agents Apprenants. L'Agent à tendance individuelle peut être caractérisé par une ou plusieurs des propriétés

précédemment définies; ainsi à partir des critères de classification que nous venons de présenter et en considérant la tendance individuelle des Agents, nous avons dégagé les types d'Agent suivants:

- *Agent Réactif simple.*
- *Agent Réactif adaptatif.*
- *Agent Intelligent.*
- *Agent Interface.*
- *Agent Personnel.*
- *Agent d'informations.*
- *Agent Mobile.*
- *Agent Hybride.*

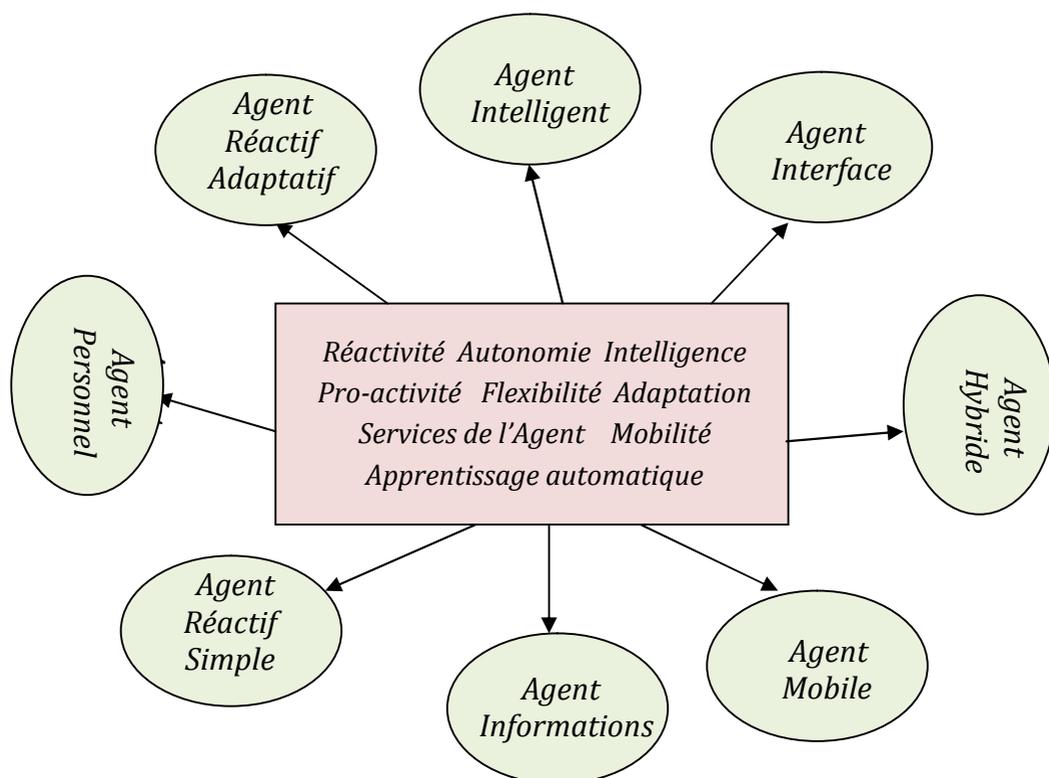


Figure 3.2: Une typologie d'Agents.

3.3.1. Agent Réactif Simple

C'est un Agent qui se caractérise par la capacité de réagir rapidement à des problèmes simples; il agit ou réagit en réponse à des stimulus de l'environnement, c'est donc un mécanisme du genre stimulus-réponse. Ainsi, ce type d'Agent, correspond à l'Agent réactif défini au chapitre 1, appelé aussi Agent reflexe simple ou Agent comportemental [Brooks, 1991b; Jennings et al., 1998; Russell et Norvig, 2006,]. Le comportement, d'un Agent réactif simple, est basé sur des règles simples de type situation-action permettant de réagir aux changements de l'environnement. L'Agent réactif simple

ne possède pas de modèle interne de son environnement. Ce type d'Agent est surtout caractérisé par les propriétés de réactivité et d'autonomie, et il n'organise aucun souvenir explicite des expériences et ne possède aucune capacité d'apprentissage. Il ne coordonne pas ses actions dans le temps et ne crée pas des éléments conceptuels pour interpréter ou comprendre son environnement. La seule capacité importante, pour ce type d'Agent, est d'associer des entrées sensorielles à des sorties motrices; ainsi, pour un Agent réactif simple, il n'y a que le présent et les perceptions sensorielles instantanées.

Pour les exemples d'application d'Agent réactif simple, ils font généralement partie de la robotique automatique simple. Pour les Agents logiciels, les Agents réactifs simples peuvent être utilisés dans la conduite de processus industriels simples; mais il ya relativement peu de nombre d'Agent individuel réactif simple. Les Agents réactifs simples sont utilisés dans les SMA surtout pour l'émergence de comportement intelligent à travers leurs interactions.

3.3.2. Agent Réactif Adaptatif

L'Agent réactif Adaptatif, représente une catégorie d'Agent réactif qui adapte son comportement à son environnement. Ce type d'Agent tire profit de son expérience et des mauvaises et bonnes actions réalisées dans le passé. Il possède donc un comportement dynamique dépendant de l'état actuel de l'environnement ainsi que de son passé. D'où la nécessité de techniques d'apprentissage pour adapter dynamiquement le comportement de ce type d'Agent. Les propriétés caractéristiques d'un Agent réactif adaptatif sont: la réactivité, l'autonomie, l'adaptation et l'apprentissage automatique; cette dernière propriété lui permet de s'adapter à l'environnement et d'avoir une flexibilité réactive de son comportement. Contrairement à un agent réactif simple, l'Agent réactif adaptatif organise un souvenir des expériences et coordonne ses actions dans le temps. Il crée des éléments conceptuels pour interpréter et comprendre son monde. Pour un Agent réactif adaptatif il n'y a pas que le présent et les perceptions sensorielles instantanées, le passé compte beaucoup pour en tirer des enseignements.

La robotique automatique comporte des applications d'Agent réactif Adaptatif, un exemple est l'Agent robot qui se déplace dans un environnement afin d'effectuer certaines tâches avec possibilité d'éviter les obstacles.

3.3.3. Agent Intelligent

Un Agent intelligent, cognitif ou délibératif, tel qu'il a été défini au chapitre 1, possède la capacité de résoudre des problèmes complexes en maintenant une représentation interne de son environnement et un état mental explicite [Bratman, 1987; Cohen et Levesque, 1988 ; Rao et Georgeff, 1991; Shoham, 1993; Wooldridge et Jennings, 1995]. Ce type d'Agent est ainsi capable de raisonner sur les connaissances qu'il maintient; il peut examiner ses objectifs, planifier ses actions et éventuellement agir sur ses plans.

Un Agent intelligent peut être un Agent logiciel ou même un robot automatique utilisé dans plusieurs domaines tels que:

- L'aviation (contrôle et planification du trafic aérien).
- L'industrie et le commerce (Agent intelligent pour le commerce électronique).
- Le militaire (Agent intelligent d'aide au pilotage d'engins, pour le guidage de missiles et d'avions de chasse).
- L'enseignement assisté par ordinateur (Agent intelligent basé sur les profils des apprenants).....

Nous pensons qu'un Agent intelligent doit posséder un haut degré d'autonomie, de pro-activité, de flexibilité et surtout d'intelligence, d'apprentissage et d'adaptation. Le comportement intelligent et adaptatif, de l'Agent cognitif, est caractérisé par une flexibilité proactive. Un agent intelligent doit établir une représentation de la dynamique de son interaction avec l'environnement sous la forme d'un modèle interne et être capable à faire évoluer ce modèle, par apprentissage automatique, pour raisonner à partir de ce modèle. Les caractéristiques d'un Agent intelligent, lui permettent d'être capable d'organiser l'expérience vécue en construisant des concepts abstraits, de réaliser des anticipations et de planifier des actions futures.

3.3.4. Agent Interface ou Personnel

Un Agent interface est généralement destiné à l'accomplissement de tâches au profit d'utilisateurs ou d'autres systèmes. La plus part des Agents interfaces sont des « Personal Software Agent: PSA » dits aussi Agents ou assistants Personnels, ayant pour but l'adaptation aux besoins et préférences de leurs utilisateurs [Maes et Schneiderman, 1994; Negroponte, 1995]. L'Agent interface est plus général qu'un PSA, dans le sens où il peut être utilisé par plusieurs utilisateurs ou avoir un environnement autre que l'utilisateur humain, mais en pratique les Agents interfaces sont généralement des PSA. Les tâches à déléguer, à un Agent Personnel, sont nombreuses; c'est en quelque sorte une secrétaire qui agit au profit de son manager selon les connaissances dont elle dispose. Ce type d'Agent doit donc s'adapter, de façon continue, aux habitudes et intérêts de son utilisateur. Le problème est: comment concevoir des PSA possédant cette habilité d'adaptation? Une approche d'adaptation consiste à fournir, à l'Agent Personnel, une spécification explicite des préférences et intérêts de l'utilisateur. Il est évident que cette approche n'est pas intéressante, puisqu'elle ne permet pas une évolution automatique et continue des connaissances du PSA; d'où l'apport de l'apprentissage automatique pour ce type d'Agent.

Un Agent interface ou un PSA sont caractérisés par: la pro-activité, la flexibilité, l'intelligence, l'autonomie (partielle ou plus ou moins totale) et surtout l'apprentissage et l'adaptation.

Les applications des Agents interfaces et des Agents Personnels sont nombreuses, entre autres la planification des agendas et des meetings selon les préférences des usagers, l'analyse des offres et l'achat selon les intérêts des clients (qui peuvent être des systèmes) ainsi que le filtrage personnalisé des informations électroniques.

3.3.5. Agent d'informations

Les Agents d'informations sont spécialisés dans La recherche d'informations et son filtrage. Un Agent de recherche dispose de la connaissance de diverses sources d'informations, qui peut inclure le type d'informations disponibles à chaque source et comment y accéder.

Le filtrage d'informations peut se faire à l'aide de mot clés, mais un Agent de filtrage qui a pour but l'adaptation aux besoins de ses usagers nécessite le recours à l'apprentissage automatique. Et dans ce dernier cas l'Agent de filtrage sera un Agent Personnel.

L'Agent d'informations est surtout caractérisé par les propriétés d'autonomie et de pro-activité, mais si ses recherches d'informations doivent être adaptées à son usager, les propriétés d'apprentissage et d'adaptation seront importantes.

Un Agent d'informations peut être utilisé pour le suivi des changements possibles des informations vitales pour un traitement donné. Tel qu'un Agent de suivi des changements dans les conditions météorologiques pour l'aviation, ou un Agent de recherche et de filtrage d'informations sur le net.

3.3.6. Agent Mobile

Un Agent mobile possède une capacité de mobilité, donc une habilité à se déplacer de site en site en ayant conscience de son déplacement. Il est défini comme une entité autonome œuvrant généralement au nom d'un usager ou d'une application, et représentant une unité complète d'exécution. Les Agents mobiles doivent pouvoir être multi-plate-formes, en se déplaçant sur le réseau pour accomplir des tâches sans que l'utilisateur ait le moindre contrôle sur eux [Jennings et al., 1998; Wooldridge et Jennings 1995]. Un Agent mobile peut changer de site pour une recherche locale d'informations ou pour un accès à une ressource critique, et puis revenir à son site initial. L'avantage des Agents mobiles n'est pas fonctionnel, il est plutôt pratique, en effet ces Agents aident à améliorer les performances par une meilleure utilisation des ressources.

L'apprentissage automatique n'est pas une propriété primordiale pour les Agents mobiles c'est plutôt, la mobilité, l'autonomie, la réactivité ou la pro-activité. Mais sachant que pour les Agents mobiles la sécurité est d'une grande importance, le recours à l'apprentissage automatique permet de renforcer la sécurité de ces Agents.

Un domaine d'application intéressant pour les Agents mobiles est la gestion des réseaux; les opérateurs de ces derniers doivent disposer de connaissances étendues sur les divers

réseaux qui peuvent exister afin de les prendre en charge et de les contrôler. Une telle gestion a besoin de la collection d'une grande quantité de données. Les Agents mobiles représentent un outil idéal pour le rassemblement des données puis leur analyse.

3.3.7. Agent Hybride

Un Agent hybride regroupe plusieurs caractéristiques. Donc c'est une combinaison de deux ou plusieurs types d'Agents parmi les précédents. Ainsi un Agent hybride peut être un Agent personnel, ou un Agent d'informations, qui est aussi un Agent mobile. Donc, l'apprentissage automatique peut être une caractéristique importante pour un Agent hybride.

A partir des propriétés de chacun des types d'Agents présentés ci-dessus, les différents types d'Agents Apprenants dégagés sont illustrés par la figure 3.3 suivante.

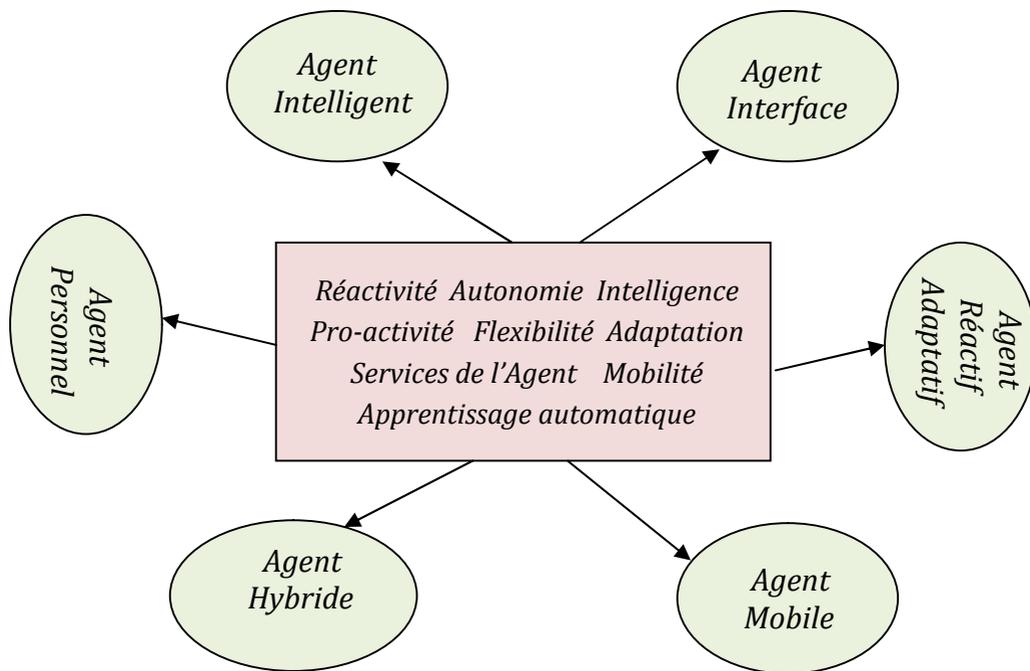


Figure 3.3: Les différents types d'Agents-Apprenants.

3.3.8. Systèmes basés Agent-Apprenant

Sachant qu'un système basé Agent est un système informatique composé d'un seul Agent, et comme dans notre travail, nous nous intéressons aux systèmes basés Agent avec apprentissage automatique; nous allons donc définir ce type de systèmes.

Un système basé Agent-Apprenant est un système informatique composé d'un seul Agent avec des capacités d'apprentissage, donc d'un seul Agent Apprenant. Un système basé Agent-Apprenant doit être capable:

- *de percevoir son environnement.*
- *d'apprendre à partir de son environnement.*
- *d'un comportement adaptatif.*
- *d'une résolution de problèmes complexes.*

Le type de l'Agent composant un système basé Agent-Apprenant peut être l'un des types d'Agents de la figure 3.3; ainsi il peut être un:

- *Agent Réactif Adaptatif.*
- *Agent Intelligent.*
- *Agent Interface.*
- *Agent Personnel.*
- *Agent Hybride.*
- *Agent Mobile.*

Nous avons considéré l'Agent Mobile parmi les Agents Apprenants pour l'apport de l'apprentissage automatique au renforcement de la sécurité au niveau de ce type d'Agent. Nous présenterons dans la section 3.4.2.2, l'apprentissage le plus approprié pour l'Agent Mobile.

Un système basé Agent-Apprenant est surtout caractérisé par les propriétés suivantes:

- *L'autonomie.*
- *L'intelligence.*
- *La flexibilité.*
- *L'adaptation.*
- *L'apprentissage automatique.*

Pour résumer les origines et les propriétés d'un système basé Agent-Apprenant, nous proposons la représentation graphique suivante:

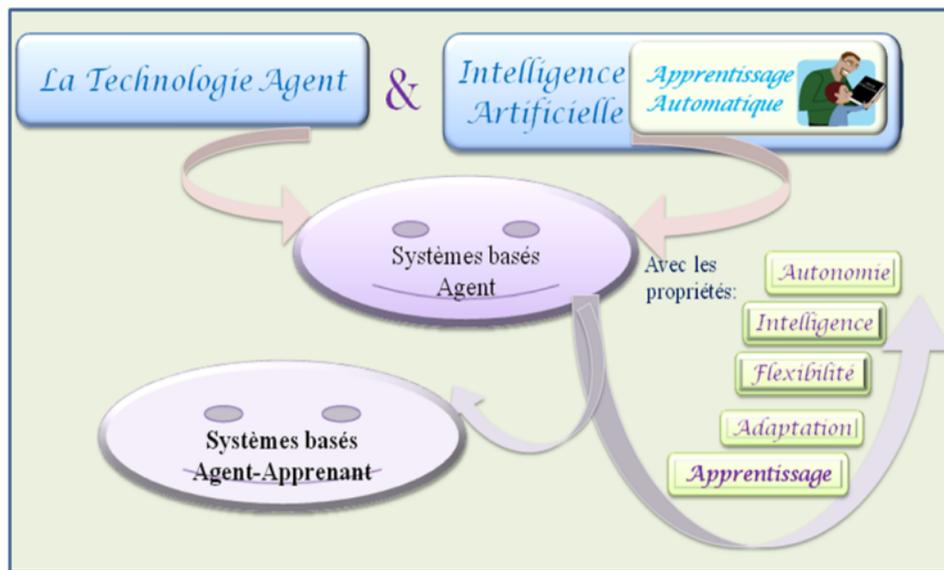


Figure 3.4: Caractéristiques des systèmes basés Agent-Apprenant.

Maintenant que les différents types d'Agents-Apprenants sont déterminés et que les approches de l'apprentissage automatique sont définies, avec leurs méthodes les plus représentatives, la question est pour chaque type d'Agents-Apprenants quelles sont les approches et les méthodes d'apprentissage les plus appropriées? La section suivante a pour objectif de répondre à cette question.

3.4. Mise en Correspondance entre Agents-Apprenants et les Approches/Méthodes d'Apprentissage

L'objectif final de ce chapitre est de présenter la mise en correspondance établie entre les différents types d'Agents-Apprenants déterminés d'une part et les approches/méthodes d'apprentissage automatique, étudiées au chapitre 2, d'autre part. Mais avant d'entamer cet objectif, il est primordial de définir ce que doit apprendre les systèmes basés Agent-Apprenant.

Avant tout, un système basé Agent peut-il adapter son comportement et résoudre des problèmes complexes sans faire recours à l'apprentissage automatique? Un système basé Agent peut présenter un comportement adapté sans l'avoir appris. Il peut résoudre ses problèmes (intelligemment), simplement à partir d'un ensemble d'instructions préétablies. Toutefois, pour que ce soit possible, il est nécessaire que toutes les stratégies de résolution soient connues par les concepteurs. Et que la quantité de situations possibles dans l'environnement soit suffisamment petite pour que le comportement, du système basé Agent, puisse être spécifié et défini, de manière complète, au préalable.

Ainsi, construire un système basé Agent adapté à l'environnement exige que les développeurs maîtrisent les connaissances nécessaires pour résoudre les problèmes que

ce système va rencontrer au fil de son existence. Il est évident, que pour des problèmes réels, il ya de nombreux cas où il est difficile ou même impossible de prévoir des solutions pour toutes les situations. Ces problèmes exigent que les systèmes basés Agent apprennent à travers l'expérience afin de s'adapter à des situations imprévues.

Donc afin d'adapter son comportement et résoudre des problèmes complexes, un système basé Agent nécessite le recours à l'apprentissage automatique. Par conséquent un système basé Agent-Apprenant peut apprendre, de façon autonome, par observations et expériences et à partir d'un cycle d'interactions avec son environnement en:

- *Construisant et reconstruisant des hypothèses sur les phénomènes qu'il observe.*
- *Identifiant des régularités et des similarités entre différentes situations et données.*
- *Développant des concepts abstraits couvrants plusieurs situations.*
- *Faisant correspondre le comportement aux situations par adaptation des actions.*

Ainsi, l'apprentissage automatique dans un système basé Agent peut être:

- *Un apprentissage de politique et stratégie de contrôle pour adapter des actions à des situations.*
- *Un apprentissage de connaissances pour une résolution de problèmes complexes.*

L'apprentissage de stratégie de contrôle permet d'apprendre de nouvelles structures d'interaction avec l'environnement, autrement dit c'est l'apparition et l'acquisition de nouveaux comportements plus adaptés à ce dernier.

Un apprentissage de connaissances pour la résolution de problèmes complexes correspond à une construction automatique de connaissances générales, par découverte automatique de toutes sortes de formes remarquables au niveau des données et des situations rencontrées.

Maintenant, nous allons considérer chaque type d'Agent-Apprenant, déterminé précédemment, pour définir son type d'apprentissage et le mettre en correspondance avec les approches et méthodes de l'apprentissage automatique les plus appropriées.

3.4.1. Agent Réactif Adaptatif et Approches/Méthodes d'Apprentissage

Un Agent réactif adaptatif fait un apprentissage de politique et stratégie de contrôle, il adapte ses actions aux états de son environnement. Pour ce type d'Agent, nous proposons une approche d'apprentissage par renforcement; en effet l'Agent réactif adaptatif doit effectuer l'action la mieux appropriée à l'état actuel de son environnement, il adapte ainsi son comportement. Ce type d'Agent est caractérisé par un haut degré d'adaptation dynamique et c'est l'apprentissage par renforcement qui peut renforcer cette

adaptation, en se basant sur la maximisation de la récompense retournée par l'environnement.

L'Agent réactif adaptatif doit adopter un apprentissage de stratégie de contrôle lui permettant d'apprendre une politique d'actions visant à atteindre ses objectifs, c'est-à-dire, la maximisation des situations effectivement positives au cours de sa vie. Et ainsi, parmi les approches d'apprentissage, étudiées au chapitre 2, l'approche par renforcement est la plus appropriée et celle qui correspond le mieux à l'Agent réactif adaptatif.

Avec ce type d'apprentissage, l'Agent réactif adaptatif apprend une politique d'actions lui permettant de choisir l'action qui le fait adapter à son environnement et lui permet d'être mieux récompensé. L'apprentissage à partir de la récompense ou la pénalité et des actions passées permet à ce type d'Agent de modifier son comportement pour sélectionner la meilleure action. Quand à la technique d'apprentissage à utiliser pour un Agent réactif adaptatif, la méthode la plus représentative de l'approche par renforcement, est le Q-learning. Cette dernière permet d'apprendre une stratégie et une politique de contrôle assurant un comportement adaptatif pour ce type d'Agent.

Un apprentissage par renforcement, contrairement à l'apprentissage inductif et déductif, permet à l'Agent réactif adaptatif d'apprendre par sa propre expérience donc sans disposer d'une source d'exemples qui désignent les comportements corrects ou désirables. Ce type d'Agents apprend donc par essais et erreurs une bonne politique d'actions par l'expérimentation et l'expérience dans un environnement quelconque en se basant sur les renforcements de ce dernier.

Nous pensons que l'efficacité de l'apprentissage par renforcement dépend essentiellement des deux paramètres suivants, qui ont été définis au chapitre 2:

1. *La fonction de renforcement.*
2. *Le dilemme exploration/exploitation.*

Pour le premier point, nous proposons de concevoir la fonction de renforcement tel que les récompenses initiales ne pénalisent pas trop l'Agent réactif adaptatif, afin de lui permettre d'apprendre de ses erreurs et ensuite d'évoluer. Ainsi la fonction de renforcement doit être temporellement progressive, en assignant des récompenses incrémentales pour permettre l'évolution, l'auto-amélioration et l'adaptation de l'Agent.

Pour pallier le problème du dilemme « exploitation/exploration » de l'espace de recherche, une balance entre l'exploration et l'exploitation de l'espace de situation-action doit être atteinte. Nous pensons que les algorithmes génétiques permettent d'atteindre cette balance puisqu'ils permettent d'exploiter la solution couramment disponible et continuent à explorer l'espace, de situation-action, à la recherche d'une solution meilleure. Aussi l'utilisation de certaines heuristiques liées au domaine de l'application de l'Agent réactif

adaptatif peut aider dans la réduction de la taille de l'espace de situation-action et ainsi réduire la complexité de la recherche dans cet espace et assurer une balance entre l'exploration et l'exploitation.

3.4.2. Systèmes basés Agent-Apprenant et Approches/Méthodes d'Apprentissage

Nous présentons, ici, la mise en correspondance des Agents Apprenants avec les approches/méthodes d'apprentissage en excluant les Agents réactifs adaptatifs, que nous avons déjà traités. L'apprentissage automatique dans un Agent réactif adaptatif diffère de l'apprentissage dans les autres types d'Agents-Apprenants; ainsi nous traitons, ces derniers, ensemble dans cette section.

Les systèmes basés Agent-Apprenant correspondant aux types suivants:

- *Agent Intelligent;*
- *Agent Interface;*
- *Agent Personnel;*
- *Agent Mobile;*
- *Agent Hybride;*

sont des systèmes qui font recours à l'apprentissage automatique pour la résolution de problèmes complexes. Ainsi, ces systèmes nécessitent un apprentissage de connaissances, autrement dit une construction automatique de connaissances; d'où leur différence avec un système basé Agent-réactif adaptatif. Pour, ce dernier type d'Agents ne disposant pas de représentation interne, seule l'apprentissage par renforcement peut aider dans l'adaptation du comportement.

Ainsi, pour une résolution de problèmes complexes, les systèmes basés Agent Apprenant, (donc le type de l'Agent Apprenant correspond à l'un des types énumérés dans cette section), nécessitent une approche d'apprentissage permettant la construction automatique de connaissances. Par conséquent, pour ces systèmes basés Agent Apprenant, les approches d'apprentissage automatique requises sont:

- *Approche Déductive.*
- *Approche Inductive.*
- *Approche Hybride.*

Pour décider d'une approche, parmi les trois citées ci-dessus, nous proposons un facteur clé; ce dernier permet d'opter pour l'une des trois approches. Ce facteur est la disponibilité des:

- *Connaissances du domaine.*
- *Exemples d'entraînement.*
- *Connaissances du domaine et Exemples d'entraînement.*

Donc, selon la disponibilité des connaissances du domaine ou des exemples d'entraînement ou des connaissances du domaine et des exemples d'entraînement ensemble, l'approche d'apprentissage à adopter, pour un système basé Agent Apprenant, est respectivement: l'approche déductive, l'approche inductive ou l'approche hybride.

3.4.2.1. Systèmes basés Agent-Apprenant et l'Approche Déductive

Nous trouvons l'approche déductive, dite aussi analytique, bien appropriée à un système basé Agent-Apprenant pour un domaine disposant d'une théorie parfaite prédéfinie, dite aussi connaissances préalables. A notre avis, la véracité de l'approche analytique permet un apprentissage de connaissances avec un haut degré de fiabilité et ne nécessite pas de tests des connaissances apprises. Cette véracité est due à la déduction, le principe d'inférence de cette approche, qui se fonde sur des axiomes et produit des résultats tautologiques. La déduction préserve la vérité en partant de connaissances établies au niveau de la théorie du domaine pour en déduire des connaissances plus efficaces et plus opérationnelles.

Un domaine crucial, où les connaissances préalables sont d'une grande importance est celui du Génie Civil et plus précisément du génie para-séismique. En effet, les experts en matière de génie civil disposent d'une grande théorie de règles normalisées pour le contrôle des constructions avant et après leurs réalisations. En effet ces règles sont utilisées d'une part, par les ingénieurs et experts en génie civil pour la conception d'une structure (en béton ou en acier) à partir d'une architecture bien définie. Ensuite cette structure est soumise à un ensemble de charges, de forces et de contraintes pour le calcul et la vérification par rapport aux règles para-séismiques. D'autres part, cette théorie de règles normalisées est aussi exploitée pour la réparation des pathologies des constructions, causées par l'usure ou les séismes.

En Algérie, et surtout après le séisme de Boumerdes en 2003 et les dégâts causés par le manque de prise en compte des règles para-séismiques, le contrôle technique des constructions avant et en cours de leurs réalisations est devenu obligatoire. En effet toute réalisation nécessite un permis de construire, délivré par les services communaux après approbation d'un dossier d'étude d'architecture et de génie civil, préalablement contrôlé techniquement et visé par l'organisme de Contrôle Technique des Constructions (le CTC). Ainsi, pour ce secteur vital, la mauvaise exploitation ou l'application partielle de la théorie de règles normalisées disponibles peuvent entraîner d'immenses dégâts matériels et humains. Certes il existe des systèmes (logiciels) de calcul et de vérification de structures utilisés par les ingénieurs et les experts en génie civil, tels que : le SAP (Structures Analysis Program) et l'ETABS (Elastic Three dimensional Analysis of Building Structures). Mais l'inconvénient est que ces systèmes sont basés sur des méthodes de calcul et des règles plus au moins anciennes. Même si ces normes sont toujours valides, elles restent non évolutives et par conséquent les systèmes de calcul et de vérification de

structures les utilisant ne peuvent prendre en considération de nouveaux paramètres. Par conséquent, le recours à l'apprentissage automatique d'une part et aux Agents intelligents d'autre part permet de disposer de systèmes basés Agent-Apprenant pour l'exploitation intelligente et complète de la théorie disponible de ce domaine critique. Un apprentissage guidé par les connaissances préalables et augmenté par les expériences réelles est d'une grande utilité et apportera une aide considérable aux experts en génie civil, surtout pour:

- *La conception semi automatique de structures selon les règles para-séismiques;*
- *Le calcul et la vérification de structures par rapport aux connaissances disponibles et celles apprises;*
- *L'analyse des pathologies et des dommages des constructions causés par les séismes et l'apprentissage automatique de connaissances para-séismiques.*
- *La prédiction des réparations nécessaires et faisables pour les constructions endommagées.*

Ainsi une approche d'apprentissage déductive est adéquate à ce domaine disposant d'une théorie assez riche et fondée sur des bases mathématiques et physiques rigoureuses. Nous généralisons que pour tout système basé Agent-Apprenant à construire pour un domaine disposant d'une théorie préétablie et bien fondée, l'approche déductive est la plus appropriée.

3.4.2.2. Systèmes basés Agent-Apprenant et l'Approche Inductive

Maintenant, pour les domaines d'application ne disposant pas d'une théorie prédéfinie, qui est le cas le plus fréquent des applications réelles, nous proposons l'approche inductive pour la construction automatique de connaissances. Cette approche nécessite un grand ensemble d'exemples d'entraînement, nous pensons même que si ces derniers ne sont pas initialement disponibles, les systèmes basés Agent Apprenant peuvent construire ces exemples d'entraînement. Les systèmes basés Agent sont équipés de capteurs qui sont sensés observer l'environnement et capter tout événement ou action dans ce dernier; ainsi ces systèmes peuvent:

- *acquérir directement les exemples d'entraînement ou;*
- *construire des exemples d'entraînement par observation ou;*
- *extraire, des exemples d'entraînement, à partir des bases de données existantes.*

En effet les exemples de base pour l'apprentissage peuvent être directement fournis par l'environnement ou construits suite à des observations et perceptions de ce dernier. Pour la première source de la base d'apprentissage (les exemples d'entraînement), nous fournissons comme exemple le domaine biologique, où cette base peut être tirée directement des essais de laboratoires et des expériences précédentes. Pour les domaines ne disposant pas de base d'apprentissage, les données captées par perception peuvent être fournies à des générateurs qui aident à construire des exemples d'entraînement. Ces

propositions seront mieux illustrées avec le modèle général, établi pour les systèmes basés Agent Apprenant, que nous présenterons au chapitre 4; et plus précisément avec l'instanciation de ce modèle au chapitre 5 par un domaine où les exemples d'entraînement sont construits automatiquement.

L'approche inductive est, à notre avis, largement applicable à un grand éventail d'applications pratiques et dont les systèmes basés Agent peuvent tirer profit et apprendre à résoudre des problèmes complexes pour des domaines variés et multiples. Nous avons indiqué au chapitre 1 que la croissance intensive de l'Internet et sa prolifération immense ont donné naissance à de multiples applications d'Agents. Les avancées technologiques actuelles, surtout en télécommunication et en outils électroniques intégrant des capacités de traitement automatique avancées nécessitent de plus en plus des systèmes à base d'Agents Apprenants.

A cette époque de l'information et de la télécommunication, les types d'Agents Apprenants les plus fréquents sont les Agents Intelligents, puis les Agents Interfaces et plus particulièrement les Agents Personnels. Ces derniers assistent les usagers dans l'accomplissement de plusieurs tâches, et tentent d'atteindre un haut degré d'adaptation et de personnalisation. Les Agents Personnels nécessitent à notre avis une approche d'apprentissage inductive. En effet, pour un objectif de personnalisation, les théories de domaines ne peuvent exister. Et c'est l'approche inductive qui doit se charger, à partir d'exemples d'entraînement, de la construction automatique de connaissances et de modèles, représentant les profils des usagers.

Les Agents Interfaces ou Personnels peuvent accomplir différentes tâches, entre autres la recherche d'informations personnalisée sur l'Internet. Pour un système basé Agent supportant cette recherche personnalisée la difficulté réside d'une part dans les thèmes variés qui pourraient intéresser l'utilisateur et d'autre part dans les changements des intérêts eux même. Par conséquent, le système basé Agent doit apprendre des profils d'intérêts multiples de l'utilisateur. L'apprentissage inductif permet de construire ces profils et aussi de prendre en charge les changements des intérêts de l'utilisateur; l'approche inductive assure la construction de ces profils par induction à partir d'observations et sans connaissances préalables.

Ainsi un système basé Agent Apprenant, pour la recherche d'informations personnalisée, peut proposer des suggestions d'informations susceptibles d'intéresser l'utilisateur. Ce système opère un suivi du comportement de l'utilisateur et tente d'anticiper ses centres d'intérêts en effectuant une exploration autonome des liens basée sur les profils appris et enrichis de façon continue.

Les applications des Agents Interfaces, qui jouent le rôle de véritables « maîtres d'hôtel digitaux » ou de « secrétaires virtuelles », sont nombreuses. Ces Agents

augmentés par les capacités de l'apprentissage inductif, apprennent des modèles évolutifs des usagers pour les soulager et les remplacer dans la prise en charge de toutes sortes de tâches comme la planification des meetings, de voyages et de vacances, le shopping sur le Web, le filtrage d'informations ou.....

Le filtrage personnalisé d'informations électroniques est un domaine intéressant d'application des systèmes basés Agent Apprenant, qui peut utiliser divers supports. En effet, en plus des stations de travail et des ordinateurs personnels, d'autres supports sont possibles, entre autres: les téléphones portables et les radios à bords des véhicules, qui présentent des informations textuelles, visuelles ou sonores filtrées et adaptées à l'utilisateur. Les systèmes intégrés dans ces supports utilisent des protocoles d'accès à l'Internet pour récupérer les informations; puis se basent sur l'apprentissage automatique, généralement inductif pour la personnalisation et la construction des profils des usagers. Ces systèmes basés Agent Apprenant doivent posséder des capteurs puissants leur permettant de prendre en charge des entrées sous formes: textuelles, empruntes digitales ou signaux sonores. Aussi les effecteurs, surtout pour la radio personnelle, doivent avoir la capacité de transformer les informations filtrées en signal audio. Par conséquent ces systèmes doivent faire un recours massif à l'apprentissage automatique, pas uniquement pour la construction des profils des usagers, mais aussi pour la reconnaissance de formes et le traitement de signal audio.

Des Agents mobiles affrontant les problèmes d'attaques, au cours de leur déplacement, doivent se protéger et disposer d'un haut degré de sécurité. Pour ce type d'Agents, l'apprentissage automatique permet d'améliorer considérablement leur sécurité. En effet, la détection d'attaques nécessite qu'un grand nombre de données soient collectées et enregistrées afin d'être analysées. La détection d'anomalie, définit un modèle d'activité normale en construisant un profil de l'activité utilisateur; et toute déviation de la norme établie est considérée comme anormale. Ainsi, le modèle d'activité normale est construit par apprentissage automatique; et comme ce dernier se base sur un grand nombre de données collectées, alors l'approche d'apprentissage appropriée est inductive.

3.4.2.3. Systèmes basés Agent-Apprenant et l'Approche Hybride

En réalité, il existe des domaines, où une théorie prédéfinie est disponible mais il est connu qu'elle ne couvre pas tous les exemples possibles; il s'agit d'une théorie incomplète ou approximative. Nous trouvons que la médecine est un bon exemple pour ce type de domaine. En effet si on considère une spécialité donnée en médecine, tels que: la médecine interne, l'infectieux, la pédiatrie, etc. Et que nous voulons développer un système basé Agent-Apprenant, pour l'aide au diagnostic médical, nous allons certainement trouver une théorie médicale prédéfinie; mais il est possible qu'elle ne puisse couvrir et répondre à tous les symptômes, bilan d'analyse et radiographie avancés possibles. Ce domaine ne peut

disposer d'une théorie parfaite; nous rappelons que le chapitre 2 a défini ce type de théorie. Aussi, il est tout à fait logique qu'une théorie établie depuis un certain temps ne puisse satisfaire des cas récents, et ainsi nous trouvons que l'apprentissage inductif doit participer pour combler l'incomplétude des théories imparfaites.

Donc, pour ce domaine nous proposons une approche d'apprentissage hybride associée à des Agents intelligents équipés, éventuellement de capacités de traitement d'images et de reconnaissance de formes. Ces capacités permettent au système basé Agent-Apprenant de prendre en considération l'imagerie médicale dans l'aide au diagnostic. Avec cet exemple illustratif, nous généralisons notre mise en correspondance en proposant pour les systèmes basés Agent Apprenant, disposants de théories de domaines approximatives, une approche d'apprentissage hybride combinant les principes des deux approches déductive et inductive.

Une approche purement inductive ou purement déductive peut être augmentée par un apprentissage par renforcement, est sera une approche hybride : inductive-renforcement ou déductive-renforcement. Cette hybridation permet de mettre en correspondance l'une de ces deux approches hybrides avec des systèmes basés Agent-Apprenant, nécessitant une construction automatique de connaissances pour des résolutions de problèmes ainsi qu'un apprentissage de stratégies de contrôle. Ces deux approches hybrides peuvent être mises en correspondance avec des Agents intelligents ou hybrides supportant des applications de jeux.

En effet, les jeux sont variés, certains possèdent des connaissances établies au préalable et l'apprentissage consiste à les transformer sous une forme plus opérationnelle et à apprendre une stratégie du gain d'une partie du jeu. Ce type nécessite une approche déductive-renforcement permettant de généraliser, à partir des connaissances préalables, pour traiter la situation en cours et d'apprendre une politique optimale qui fait converger le système vers le gain de la partie en cours. D'autres domaines de jeux ne possèdent pas de connaissances préalables, ainsi l'apprentissage doit se baser sur l'essai et l'erreur et sur la récompense associée au gain d'une partie. Aussi les expériences et les situations de jeux passées sont des exemples d'entraînement qui peuvent être utilisés pour résoudre la situation en cours. Ainsi, pour des systèmes basés Agent-Apprenant, supportant ce dernier type de jeux, l'approche d'apprentissage la plus adéquate est l'approche hybride: inductive-renforcement.

Par conséquent, pour des Agents intelligents ou hybrides, possédant une représentation mentale et devant adapter leurs comportements à leurs environnements, une approche hybride est souhaitable. Cette approche permet de combiner plusieurs méthodes: synthétique, analytique et par renforcement. La mise en œuvre, de l'approche hybride, est plus ou moins difficile car elle peut nécessiter l'intégration de plusieurs formalismes de

représentation de connaissances et de différents mécanismes d'apprentissage dans un système basé Agent-Apprenant.

Une approche d'apprentissage hybride utilisée par des Agents intelligents ou Agents hybrides leur permet d'avoir des comportements proactifs et d'opérer de manière intelligente et autonome. Un domaine intéressant d'application de ces types d'Agents, qui doivent intégrer de hautes capacités d'apprentissage est le pilotage automatique ou l'aide au pilotage d'engins, tels que des missiles, des avions de chasse et des véhicules ordinaires sur les autoroutes. En général, le domaine de la robotique automatique, nécessitant des systèmes avec des représentations de l'environnement et une commande intelligente qui peut s'adapter aux changements de ce dernier, est un large champs d'application des Agents intelligents intégrant de hautes capacités de l'apprentissage automatique via une approche hybride.

Donc, concernant le choix de l'approche inductive ou déductive, pour une construction automatique de connaissances par un système basé Agent-Apprenant, nous concluons par ce tableau illustratif.

Type d'apprentissage Données disponibles	Approche d'apprentissage Inductive	Approche d'apprentissage Déductive	Approche d'apprentissage Hybride
<i>Théorie de Domaine Parfaite</i>		*	
<i>Théorie de Domaine Approximative</i>			*
<i>Exemples d'enTraînement (ET)</i>	*: {ET} = Une base Etendue	*: Quelques ET	*: {ET} = Une base modérée

Table 1: Les entrées des approches d'apprentissage automatique.

3.4.2.4. Systèmes basés Agent-Apprenant et Méthodes d'Apprentissage

Une fois l'approche d'apprentissage déterminée pour un système basé Agent-Apprenant, il restera à choisir une méthode d'apprentissage. A ce niveau, le choix devient plus difficile à cause de la grande variété des méthodes d'apprentissage, en particulier pour l'approche inductive. Nous avons présenté, au chapitre 2, plus de méthodes représentatives de l'approche inductive que de méthodes représentatives pour chacune des autres approches.

Si pour un système basé Agent-Apprenant, l'approche choisie est analytique (déductive), alors le choix d'une méthode déductive est plus simple que celui des méthodes inductives.

En effet, d'une part les méthodes déductives ne sont pas nombreuses, et d'autre part les méthodes analytiques partagent un principe commun. Ce principe, présenté au chapitre 2, est: l'Explanation Based Learning (EBL). Dans l'EBL, les connaissances du domaine sont utilisées pour analyser et expliquer comment chaque exemple d'entraînement satisfait le concept cible. Donc la généralisation est faite selon un raisonnement logique préservant la vérité des déductions. Il faut noter que l'apprentissage basé explication, permet une généralisation rationnelle des exemples en se servant de la théorie du domaine. Un aspect important de l'EBL est sa capacité à formuler des caractéristiques nouvelles non explicites dans les exemples à traiter. L'algorithme le plus connu dans la littérature et le plus représentatif de l'EBL est Prolog-EBG, présenté au chapitre 2.

Si un système basé Agent-Apprenant nécessite une approche inductive, alors quelle méthode d'apprentissage choisir parmi la pléthore de techniques existantes? A ce niveau nous guidons le choix, d'une méthode d'apprentissage inductive, par les critères suivants:

1. *La richesse du formalisme de la représentation des connaissances associé à la méthode.*
2. *L'efficacité du processus d'apprentissage, utilisé par la méthode en question.*
3. *La facilité d'interprétation, des connaissances apprises, par des humains.*

Le degré d'importance de chacun des critères précédents, dépend de l'application, mais nous les avons énumérés dans l'ordre d'importance qui nous semble le plus logique, surtout pour le premier critère. En effet l'expressivité du formalisme de représentation et son adéquation pour exprimer les connaissances à apprendre, pour un domaine donné, est un facteur crucial dont dépend le choix de la technique d'apprentissage. Cette dernière doit être basée sur un formalisme permettant de représenter convenablement les connaissances de l'application. Pour le deuxième et le troisième critère, l'ordre d'importance dépend du problème à traiter. Pour certains problèmes, l'efficacité du processus d'apprentissage est plus importante que la facilité d'interprétation de ses résultats, pour d'autres c'est l'inverse.

Ainsi, pour des systèmes basés Agent-Apprenant, avec des connaissances qui peuvent être représentées par l'un des formalismes suivants:

- *Paires d'attribut/ valeur;*
- *Règles if then;*
- *Frames ou schémas;*

nous proposons, respectivement les méthodes suivantes présentées au chapitre 2:

- *Les arbres de décision et les algorithmes génétiques.*
- *La programmation logique inductive.*
- *L'apprentissage à base de cas et d'analogie.*

Ces méthodes sont en grande partie employées pour: l'adéquation de leur formalisme de représentation de connaissances avec plusieurs types d'applications, l'interprétation facile de leurs résultats et aussi leur efficacité. Il faut noter que l'apprentissage à l'aide de la programmation logique inductive est basé sur l'utilisation conjointe des exemples d'entraînement et des connaissances préalables (théorie du domaine). Cette méthode combine entre le principe inductif, pour généraliser à partir des exemples d'entraînement, et le principe déductif pour la consistance des règles apprises avec les connaissances préalables. Pour des domaines, où les connaissances nécessitent une représentation à base de prototypes, l'apprentissage à base de cas et d'analogie serait adéquat. En effet cette dernière méthode utilise un formalisme de représentation de connaissances basé sur les frames ou les schémas bien appropriés pour représenter et exprimer des connaissances sous forme de prototypes.

Pour des Agents intelligents (logiciels ou matériels) avec une perception sophistiquée à l'aide de capteurs automatiques (tels que les cameras, les capteurs de sons) et des données discrètes ou à valeurs réelles, nous proposons un apprentissage par Réseaux de Neurones Artificiels (RNA). La structure du réseau de neurones artificiels dépend de la complexité du problème à représenter. Les RNA sont adaptés à l'apprentissage à partir de données discrètes ou réelles éventuellement bruitées, c'est-à-dire incomplètes, à l'interprétation de scènes visuelles, à la reconnaissance de la voie, mais leurs résultats sont difficiles à interpréter et à expliquer. Ainsi, les RNA correspondent bien à des systèmes basés Agent-Apprenant utilisés, entre autres, pour la reconnaissance de formes ou de son, un domaine assez riche en application, surtout dans l'aide à:

- La prospection géologique et minière et l'interprétation des images des forages.
- L'interprétation d'images satellitaires ou médicales.
- L'interprétation d'enregistrement vocal et la parole humaine.

Les systèmes basés Agent-Apprenant mis en correspondance avec une approche hybride, utiliseront des méthodes hybrides qui ne sont pas nombreuses; le choix d'une technique, pour un domaine donné, dépend principalement de la nature des connaissances à apprendre. Les critères présentés, au début de cette section, peuvent guider ce choix. Les réseaux KBANN, présentés au chapitre 2, permettent de construire des réseaux de neurones à partir d'une théorie du domaine. Ils sont basés sur une méthode hybride permettant l'intégration des connaissances du domaine avec les RNA inductifs. L'extraction de règles à partir du réseau final facilite l'interprétation des résultats de cette méthode, mais les passages effectués entre les différentes phases de cette méthode et le temps qu'ils impliquent peuvent contraindre son choix. En partant du fait que la programmation logique inductive combine entre les exemples d'entraînement et les connaissances préalables, nous pouvons, par analogie au principe des réseaux KBANN, déduire que la PLI peut être considérée comme une méthode hybride. En effet, ces deux

méthodes se basent sur un principe inductif de généralisation et un principe déductif pour la consistance des exemples d'entraînement avec les connaissances préalables du domaine.

3.5. Conclusion

Le premier objectif de ce chapitre était la détermination des types d'Agents concernés par l'apprentissage automatique. Ainsi, un ensemble de critères de classification a été défini pour aboutir à une typologie d'Agents. A partir de cette dernière, nous avons dégagé les différents types d'Agents Apprenants. La mise en correspondance entre ces derniers et les approches/méthodes d'apprentissage, étudiées au chapitre 2, a constitué le second et le but final de ce chapitre. Pour atteindre cette finalité, nous avons défini au préalable la notion de système basé Agent-Apprenant avec ses propriétés principales et les deux types d'apprentissage qu'il peut mener.

Les systèmes basés Agent-Apprenant réactif adaptatif ont été mis en correspondance avec l'apprentissage par renforcement. Ce dernier est le plus apte à apprendre des stratégies de contrôle, pour une adaptation du comportement, sans données et connaissances disponibles. Pour pallier certaines difficultés, au niveau de l'apprentissage par renforcement, des propositions ont été émises. Celles ci concernent la bonne conception de la fonction de renforcement et la balance entre l'exploration et l'exploitation de l'espace de situation-action qu'un Agent réactif adaptatif tente d'atteindre.

Une construction automatique de connaissances peut être basée sur une approche déductive, inductive ou hybride. Le facteur clé pour mettre en correspondance respectivement l'une de ces trois approches avec un système basé Agent Apprenant est la disponibilité d'une théorie de domaine, d'exemples d'entraînement ou les deux ensembles. L'approche déductive est bien appropriée à un système basé Agent-Apprenant pour un domaine disposant d'une théorie prédéfinie et parfaite. Les méthodes analytiques peu nombreuses et partageant un principe commun facilitent le choix de la technique d'apprentissage. L'approche inductive est trouvée la plus appropriée aux systèmes basés Agent Apprenant, pour des domaines ne disposant pas d'une théorie prédéfinie. Pour orienter le choix d'une méthode inductive, des critères ont été présentés. Pour des Agents intelligents ou hybrides, possédant une représentation mentale et devant adapter leurs comportements à leurs environnements, une approche/méthode hybride est souhaitable. Cette approche permet de combiner plusieurs méthodes: inductive, analytique et par renforcement.

Afin d'apporter un cadre général et une assise conceptuelle à l'apprentissage automatique dans un Agent, un ensemble de concepts et un modèle général et générique, pour les systèmes basés Agent-Apprenant, sont proposés. Ce modèle ainsi que sa base conceptuelle feront l'objet du chapitre suivant.

Chapitre 4

Modèle Général et Générique de Systèmes basés-Agent Apprenant

4.1. Introduction

Après l'établissement d'une mise en correspondance entre les Agents Apprenants et les approches/méthodes d'apprentissage automatique; nous présentons, dans ce chapitre, un modèle général et générique pour les systèmes basés-Agent Apprenant. Par, les propriétés de généralité et de généricité, de ce modèle, nous visons à disposer d'un pattern qui peut être réutilisé par différents domaines d'applications et pouvant couvrir différentes approches et méthodes de l'apprentissage automatique.

Notre modèle se propose, donc, comme un cadre général qui identifie les fonctionnalités principales des systèmes basés Agent-Apprenant. Ce modèle, conçu avec une réflexion centrée sur l'apprentissage automatique générique, permet d'identifier les différents composants génériques ainsi que leurs relations d'interdépendance. Ce modèle est indépendant de tout domaine, mais il peut être instancié par différentes applications.

Dans ce chapitre nous commençons par définir les activités de modélisation, qui nous ont servi de base pour la construction de notre modèle. Ce dernier a été établi selon une démarche incrémentale de définitions, de décompositions et de raffinements. Nous présentons aussi un ensemble de concepts, liés à l'apprentissage automatique, qui

constitue une base et une assise conceptuelle pour le modèle proposé et permettant de renforcer ses propriétés de généralité et de généricité.

4.2. Pourquoi un Modèle de Systèmes basés-Agent Apprenant?

Pourquoi construire un modèle? Pour répondre à cette question il faut préalablement définir la notion de modélisation. Cette dernière est l'activité qui consiste à construire des modèles; c'est l'une des deux principales composantes de la démarche scientifique à savoir: la modélisation et l'expérimentation [Amblard et Phan, 2006; Zeigler, 1999]. Les chercheurs consacrent beaucoup de temps pour construire, tester, comparer et réviser des modèles; donc, définir la notion de modèle c'est définir la clé de ces activités. D'après [Treuil et al., 2008; Zeigler, 1999], un modèle est une construction abstraite qui permet de comprendre le fonctionnement d'un système de référence en répondant à une question qui le concerne. Ainsi, la construction d'un modèle signifie la production d'une représentation simplifiée du système à étudier.

Après avoir défini la notion de modèle, nous pouvons répondre à la question initiale, « Pourquoi construire un modèle? ». C'est pour bâtir et fournir une représentation ou une construction abstraite du système en étude. Concernant notre travail, la modélisation des systèmes basés Agent-Apprenant est équivalente à la construction d'un modèle de système général basé Agent-Apprenant générique.

La mise en correspondance établie entre les Agents Apprenants et les approches/méthodes d'apprentissage automatique, nous a permis d'aboutir aux constatations suivantes:

- *Un Agent Apprenant est fortement lié à son domaine d'application.*
- *Il existe une grande variété des domaines d'application des Agents Apprenants.*
- *Il existe une forte relation entre les connaissances à apprendre, par un Agent Apprenant, et le domaine d'application.*

Et pour surmonter cette dépendance ainsi que la spécificité des systèmes existants, qui fait que ces derniers sont des applications dédiées et non réutilisables; nous avons construit un modèle pour les systèmes basés Agent-Apprenant qui possède les caractéristiques suivantes:

- *Général et indépendant de tout domaine.*
- *Peut être instancié par différentes applications.*
- *Générique et couvre toutes les approches et méthodes de l'apprentissage automatique.*
- *Complet et couvre tous les composants pour assurer l'apprentissage au niveau d'un système basé Agent.*

Ces caractéristiques, constituent la réponse à la question suivante: «Pourquoi un modèle de systèmes basés-Agent Apprenant? ». Elles permettent d'avoir un modèle général et générique pouvant être instancié par différentes applications et couvrant toutes les approches et les méthodes de l'apprentissage automatique. Ce modèle, conçu avec une réflexion centrée sur l'apprentissage automatique générique et sur l'indépendance de tout domaine d'application, permet d'identifier les différents composants nécessaires, pour avoir un système général basé Agent-Apprenant générique, ainsi que leurs relations et dépendances. Aussi ce modèle complet peut être étendu et instancié par différents domaines d'application. L'instanciation consiste à réutiliser le modèle proposé avec ses composants génériques, constituant la base des systèmes basés-Agent Apprenant, et l'étendre par les spécificités de l'application. Pour construire ce modèle, nous nous sommes basés sur les activités de la modélisation, que nous allons définir à présent.

4.3. Principales Activités de la Modélisation

Pour mener une modélisation, plusieurs auteurs, entre autres [Amblard et Phan, 2006; Zeigler, 1999;] considèrent qu'il faut la décomposer en plusieurs activités qui consistent à:

1. *Définir les questions que l'on se pose sur le système (en étude).*
2. *Définir ce qu'est le système, les éléments le composant et leurs relations, en fonction des questions posées.*
3. *Décomposer le système et modéliser chacune de ses parties.*

Notons que la modélisation peut inclure aussi l'utilisation de langages ou formalismes pour exprimer et spécifier les modèles de façon plus ou moins formelle. Notre travail n'est pas concerné par cette phase de modélisation et vise surtout à créer et à établir une construction abstraite qui permet de comprendre le fonctionnement d'un système général basé Agent-Apprenant générique. Et par conséquent construire un modèle en produisant une représentation simplifiée de ce système. Une fois ce modèle établi, l'utilisation de langages ou formalismes pour sa description et sa spécification est l'une des perspectives de notre travail.

Donc, construire un modèle général pour les systèmes basés Agent-Apprenant doit passer par les activités précédentes indépendamment de toute application et de tout domaine. Ainsi le modèle constitue un moule et un pattern à partir duquel différentes applications peuvent être instanciées. Pour bâtir notre modèle, nous avons mené les activités de modélisation en commençant par se poser les questions suivantes:

1. *Qu'est ce qu'un système basé-Agent Apprenant ?*
2. *Qu'est ce qu'un système général basé-Agent Apprenant ?*
3. *Qu'est ce qu'un système basé-Agent Apprenant générique?*

La première question est la plus primordiale, elle a trouvé partiellement sa réponse dans le chapitre 1 puis complètement dans le chapitre 3. Nous rappelons ici qu'un système *basé-Agent Apprenant* est un système informatique composé d'un Agent unique avec des capacités d'apprentissage automatique.

La seconde question est centrée sur la *généralité* et elle signifie, pour notre travail, indépendance, du *système général* basé-Agent Apprenant, de tout domaine d'application. Ainsi, les fonctionnalités, de ce *système général*, sont définies pour un cadre global non lié à un domaine particulier.

La troisième question concerne l'*Agent Apprenant générique*, elle vise donc la *généricité* de l'apprentissage automatique au niveau de l'Agent Apprenant. Dans cette perspective, notre modèle doit représenter un Agent Apprenant capable de couvrir toutes les approches de l'apprentissage automatique, présentées au chapitre 2 de cette thèse.

Ainsi, nous avons abouti à la notion de système général basé Agent-Apprenant générique, qui est équivalent aux systèmes basés Agent-Apprenant.

La seconde activité, a consisté à : « bien définir ce qui est un système général basé Agent-Apprenant générique, les éléments le composants et leurs relations, en fonction des questions posées ». A ce niveau il a fallu déterminer les fonctionnalités principales d'un système basé-Agent Apprenant tout en visant la généralité et la généricité. Ces fonctionnalités permettent de bien définir le système et de comprendre son fonctionnement. Ensuite les éléments composants le système ont été déterminés progressivement en procédant à la décomposition du système global et la modélisation de chacune de ses parties. L'établissement, de la seconde et la troisième, activités de la modélisation et par conséquent la construction d'un modèle de système général basé-Agent Apprenant générique, que nous nommons aussi modèle général et générique de systèmes basés-Agent Apprenant, font l'objet des sections suivantes de ce chapitre.

4.4. Fonctionnalités Principales d'un Système Général basé Agent-Apprenant Générique

La définition de ce qui est un système général basé Agent-Apprenant générique revient à définir les fonctionnalités principales que doit supporter un tel système. Puis définir progressivement les éléments composants le système et les rôles qu'ils doivent assumer, ainsi que leurs relations et interdépendances. Le terme progressivement indique que la construction du modèle est incrémentale, nous commençons alors par présenter les fonctionnalités.

Guidé par notre objectif de modèle général et générique, nous avons défini les fonctionnalités générales de systèmes basés Agent-Apprenant, indépendamment de tout

domaine d'application et de toute méthode d'apprentissage automatique. Ainsi, ce modèle doit énumérer les fonctionnalités noyaux qui doivent exister, dans chaque système basé Agent-Apprenant. De tels systèmes doivent être caractérisés, principalement, par des capacités d'apprentissage automatique de connaissances et des mécanismes de prise de décision. En effet ces systèmes tentent d'imiter les capacités de l'humain pour: percevoir l'environnement, apprendre des connaissances ou des stratégies de contrôle, raisonner et prendre des décisions concernant les problèmes à résoudre et agir en conséquence. Ainsi, indépendamment de tout domaine d'application, un système général basé Agent-Apprenant doit assumer les fonctionnalités suivantes:

- *Perception continue de son environnement.*
- *Acquisition des connaissances observées.*
- *Apprentissage de connaissances ou de stratégies de contrôle.*
- *Raisonnement et Action.*

Ces fonctionnalités sont assumées par les éléments composants le système général basé Agent-Apprenant, constituant la vue globale du modèle, qui est illustrée par la figure suivante:

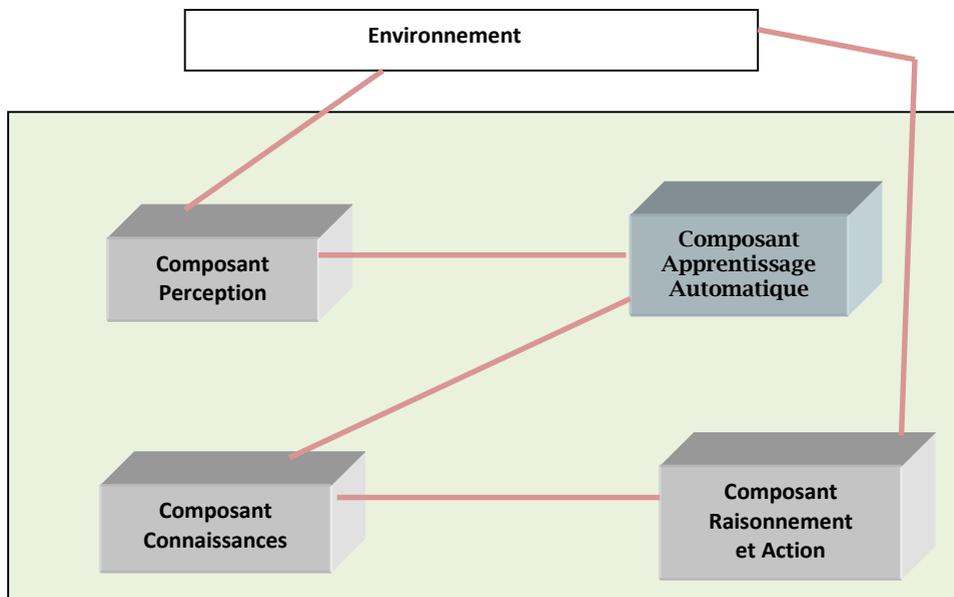


Figure 4.1: Vue globale du modèle général.

La Perception continue de l'environnement, par un système basé Agent-Apprenant, est le processus de recueil des informations perçues et observées par les capteurs et les moyens dont dispose ce système. La Perception permet à un système basé Agent d'observer son environnement et de suivre ses changements et son évolution; et par conséquent le système peut réagir à temps aux changements qui affectent le perçu.

L'environnement pour un système basé Agent est d'une grande importance, en effet l'Agent est par définition situé dans un environnement avec lequel il interagit; l'ensemble des définitions d'Agent, présentées au chapitre 1, incluent la notion d'environnement.

Si le système basé Agent est en plus Apprenant, alors l'environnement est de plus en plus important. C'est la source des changements auxquels le système basé Agent-Apprenant est sensé s'adapter. La Perception continue permet de capter toute action ou événement qui peuvent apporter un plus au processus de l'apprentissage automatique; l'acquisition des connaissances observées dans le domaine d'application, constitue la source des données pour l'apprentissage automatique. En effet les méthodes inductives démarrent d'une base d'apprentissage; les méthodes déductives nécessitent une acquisition des connaissances préalables et l'apprentissage par renforcement se base sur le signal de renforcement retourné par l'environnement. Ainsi la fonctionnalité de perception, des systèmes basés Agent-Apprenant, permet:

- *d'acquérir les données et exemples brutes observés.*
- *d'acquérir les connaissances préexistantes du domaine.*
- *d'acquérir les signaux de renforcement.*

Une fois les connaissances perçues sont acquises, par le système, des mécanismes d'apprentissage automatique doivent être utilisés afin d'apprendre et de construire automatiquement des connaissances ou des stratégies de contrôle. Les connaissances apprises enrichissent de façon continue une base de connaissances. Cette dernière est l'élément clé des raisonnements et prise de décisions établies par le système basé Agent-Apprenant pour des résolutions de problèmes complexes. Ainsi, l'apprentissage automatique continu permet, au système, d'atteindre un haut degré d'intelligence et une meilleure résolution de problèmes. Ceci conduit à une adaptation à l'environnement et une amélioration de la performance du système, suite à ses bonnes actions et prise en charge des problèmes à résoudre. En effet les règles de comportement, du système basé Agent Apprenant, sont construites dynamiquement, par apprentissage automatique, au fur et à mesure que l'environnement change. Ainsi, la capacité du système à avoir un comportement flexible et adaptatif, aux changements de son environnement, est en évolution continue. L'autonomie est certainement améliorée par l'apprentissage automatique, en effet les connaissances et les stratégies apprises aident le système à résoudre les problèmes et à s'adapter de façon autonome. Les fonctionnalités précédentes permettent aux systèmes basés Agent-Apprenant une amélioration continue des capacités suivantes:

1. *Autonomie.*
2. *Intelligence.*
3. *Adaptation à l'environnement.*
4. *Flexibilité.*
5. *Performance.*

Avant de poursuivre la description de la construction et le raffinement du modèle, des systèmes basés Agent-Apprenant, nous allons présenter un ensemble de concepts, liés à l'apprentissage automatique, qui ont été définis comme support et assise de ce modèle.

4.5. Concepts de base du Modèle Général et Générique de Systèmes basés Agent-Apprenant

Pour renforcer notre approche et disposer d'une base conceptuelle pour notre modèle, nous nous sommes basés sur une définition précise et plus ou moins formelle de l'apprentissage automatique, présentée au chapitre 1. Cette définition, émise par le chercheur Tom Mitchell, chef d'équipe à l'université Carnegie Million qui travaille massivement sur l'apprentissage automatique, stipule que:

« Un programme informatique apprend à partir de l'expérience E par rapport à une classe de tâches T et une mesure de performance P , si sa performance à l'exécution de tâches de T , mesurée par P , s'améliore avec l'expérience E »¹ [Mitchell, 1997; Mitchell, 2006].

Cette définition relativement aux autres définitions de l'apprentissage automatique, dont les principales ont été présentées au chapitre 1, renferme des éléments bien déterminés et bien précis. Certes la majorité des définitions convergent vers l'amélioration de la performance, mais celle rappelée ci-dessus spécifie clairement les éléments clés pour l'apprentissage. En effet, les éléments T , E et P sont des facteurs formels à déterminer pour un système apprenant. Tom Mitchell affirme que: « pour bien concevoir un système apprenant pour un domaine donné, les trois éléments: T , E et P sont à identifier et à définir à la phase de spécification et de conception du système à développer » [Mitchell, 1997; Mitchell, 2006]. Pour bien comprendre les éléments: T , E et P , nous présentons les deux exemples suivants.

Exemples

1. Un Système de Reconnaissance de Caractère

Pour un système apprenant, du domaine de la reconnaissance de formes, qui permet de reconnaître automatiquement des lettres manuscrites. Donc un O.C.R (Optical Character Recognition), les différents éléments du problème d'apprentissage selon la définition formelle précédente sont:

- T : reconnaître et classer des lettres manuscrites décrites par des images, le système apprenant doit être capable de faire correspondre à chaque image la lettre associée (une image donnée est un a, b, c ou... z);

¹ "A Computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E " [Mitchell, 1997; Mitchell, 2006].

- **P** : la proportion de lettres correctement reconnues;
- **E** : une base de données contenant des lettres sous forme d'images avec la lettre de l'alphabet que représente chaque image.

2. Un Système de Conduction Automatique de Véhicule

Si on considère, l'exemple d'un robot de conduction automatique de véhicule [Mitchell, 1997], alors pour ce système apprenant les facteurs T, E et P sont:

- **T** : la classe de tâches, que le robot doit assurer, est la conduction automatique d'un véhicule sur une autoroute en utilisant des capteurs visuels;
- **P** : pour ce système, la mesure de performance P est la distance moyenne parcourue par le robot avant une erreur de conduction;
- **E** : la base des expériences ou des exemples d'apprentissage est une séquence d'images et de commandes de conduite, enregistrées durant l'observation d'un conducteur humain.

La base conceptuelle, du modèle de systèmes basés Agent-Apprenant, doit renforcer les propriétés de généralité et de généricité visées par ce modèle, qui doit être :

1. *Général, donc indépendant de tout domaine d'application.*
2. *Générique en couvrant les différentes approches de l'apprentissage automatique.*

La première propriété, nous permet d'établir un modèle qui peut être réutilisé par différents domaines d'application, par instanciation et extension du modèle général. La deuxième propriété, nous permet de construire un modèle qui intègre différentes approches d'apprentissage automatique, ainsi il peut être instancié par différents types d'Agents et différents problèmes d'apprentissage.

Alors, pour former l'assise conceptuelle de notre modèle, les éléments : T, E et P, de la définition formelle de l'apprentissage automatique, ont été repris. Et sachant qu'ils concernent la conception d'un système apprenant donné; nous avons généralisé et redéfini ces éléments, selon les propriétés visées par notre modèle. Aussi, nous avons étendu et complété ces éléments par d'autres concepts, qui ont été induits à partir des:

- Fonctionnalités générales, que nous avons définies pour les systèmes basés Agent Apprenant, décrites dans ce chapitre.
- Etude, des approches et méthodes d'apprentissage automatique, présentée au chapitre 2.
- Domaines d'applications, multiples et variés des systèmes basés Agent Apprenant, constatés et présentés au chapitre 1.
- Mise en correspondance, établie entre les types d'Agents Apprenants et les approches/méthodes d'apprentissage automatique, décrite au chapitre 3.

Ainsi, les concepts suivants ont été réunis et définis pour former un support conceptuel à notre modèle général et générique de systèmes basés Agent Apprenant:

1. *La classe des tâches.*
2. *Une mesure de performance.*
3. *Les exemples d'entraînement.*
4. *Les connaissances préalables.*
5. *La représentation des connaissances.*
6. *Une approche /méthodes d'apprentissage.*

Maintenant, selon les objectifs de généralité et de généricité visés par notre modèle de systèmes basés Agent-Apprenant, nous allons définir chacun de ces concepts.

4.5.1. La Classe des Tâches

Pour développer un système Apprenant, généralement nous commençons à partir de son domaine d'application afin de recenser et définir les tâches T que le système doit prendre en charge [Michalski et al., 1993; Mitchell, 1997]. En effet l'apprentissage est fortement lié au domaine de l'application et le but du système Apprenant est d'assumer automatiquement les tâches du domaine avec une amélioration continue des performances. Pour les systèmes basés Agent-Apprenant, les domaines d'applications sont variés. Cette diversité, évoquée au chapitre 1, s'accroît avec les avancées technologiques, surtout en matière de réseau (Internet), de large volume de données et de connaissances électroniques. Par conséquent des outils automatiques, de prise en charge de ces applications variées, sont de plus en plus nécessaires. Pour que notre modèle prenne en considération cette diversité et afin d'être général et indépendant de tout domaine d'application, nous avons prévu un *composant général, dit application*. Ce dernier assume, en collaboration avec d'autres composants, la prise en charge des tâches du domaine, et dont la spécification et la définition sont différées à la phase d'instanciation du modèle général.

$$C = \sum T / T \in D \quad \text{avec } C : \text{classe des tâches.}$$

T : les différentes tâches.
 D : domaine d'application.

Ainsi la classe des tâches C , dans le modèle général, correspond à un composant abstrait qui sera instancié à la définition d'un domaine D et la spécification de ses tâches T .

4.5.2. Une Mesure de Performance

Mesurer la performance d'un système, revient généralement à mesurer son temps de réponse; en effet l'acception la plus courante des tests de performance est celle dans laquelle les tests ont pour objectif de mesurer les temps de réponse d'un système en

fonction de sa sollicitation. *Mais, pour un système apprenant, mesurer la performance revient à mesurer la qualité du service offert par le système qui se base, pour sa réalisation, sur l'apprentissage automatique* [Michalski et al, 1993; Mitchell, 1997; Mitchell, 2006].

Concernant la mesure de performance, nous pensons que ce concept est un élément crucial dans un système à base d'apprentissage automatique. En effet le but du système apprenant est d'assumer les tâches du domaine avec un haut degré de performance; par conséquent nous devons avoir un contrôle continu sur cet élément crucial, par une mesure de performance P bien définie. Mais cette mesure dépend de l'application et ne peut pas être définie en général [Mitchell, 1997; Russell et Norvig, 2006]; le concepteur d'un système apprenant spécifique doit définir au préalable une mesure de performance qui permet d'évaluer la prise en charge des tâches T assumées par le système. Ainsi notre modèle général, doit avoir *un évaluateur de performance*, dont le rôle est l'évaluation de la qualité du service offert par un système donné. De façon générale, nous spécifions la mesure de performance P à évaluer par:

$$P = \text{une fonction des variables du système ayant une relation avec les résultats obtenus concernant la prise en charge des tâches } T$$

4.5.3. Les Exemples d'Entraînement

Pour apprendre automatiquement des connaissances, un système doit avoir des expériences dites aussi des exemples d'entraînement [Cornuéjols et Miclet, 2002; Michalski et al, 1993]. L'ensemble E des exemples d'entraînement est l'un des trois éléments de la définition formelle de Tom Mitchell [Mitchell, 1997; Mitchell, 2006] sur lequel se base l'apprentissage automatique. Nous avons présenté, au chapitre 2, l'approche inductive, de l'apprentissage automatique, dont les méthodes nécessitent souvent un grand ensemble d'exemples d'entraînement pour pouvoir induire des concepts généraux. Aussi au chapitre 3, nous avons constaté le nombre important des domaines d'application des systèmes apprenants inductifs, représentant les cas les plus fréquents des applications des Agent-Apprenant. L'approche déductive utilise aussi un certain nombre d'exemples d'entraînement, par conséquent, ces exemples, quelque soit leur nombre, représente une source importante pour le processus de construction de connaissances. Le modèle que nous proposons intègre la prise en charge des exemples d'entraînement et leur production par un composant dit *générateur des exemples d'entraînement* E .

$$E = \{ e / e \text{ est un exemple d'entraînement} \}$$

Les exemples d'entraînement $\{e\}$ sont: fournis directement par l'environnement, acquis par une observation et perception à travers les capteurs d'un système basé Agent, extraits à partir d'une base de données ou produits automatiquement.

4.5.4. Les Connaissances Préalables

Ils existent des applications où il ne s'agit pas d'apprendre uniquement à partir des exemples d'entraînement, mais aussi à partir des connaissances préalables du domaine. La mise en correspondance, établie entre les systèmes basés Agent-Apprenant et les approches/méthodes d'apprentissage automatique présentée au chapitre 3, a montré que pour ces applications, l'approche d'apprentissage la plus appropriée est soit déductive, soit hybride (déductive-inductive). En effet ils existent des domaines où un savoir cognitif préétabli et bien structuré est disponible; si ce savoir, dit aussi la théorie du domaine, est parfait, alors l'approche d'apprentissage déductive est la plus convenable. Par contre si cette théorie de domaine est approximative, alors l'approche hybride déductive-inductive est la plus adéquate. Par conséquent, partons des objectifs visés par notre modèle, de systèmes basés Agent-Apprenant, nous augmentons les éléments (T, E et P) de la définition formelle de l'apprentissage automatique, par un autre concept: «*Les connaissances préalables du domaine* » comme un élément complémentaire aux exemples d'entraînement. Notre but est de disposer d'une base conceptuelle assez complète pour pouvoir couvrir les deux objectifs de généralité et de généricité de notre modèle. Ainsi, La théorie du domaine, appelée aussi le background *B*, avec :

$$B = \text{connaissances préalables,}$$

est un concept très important pour les applications, où l'apprentissage doit être basé sur des connaissances prédéfinies parfaites ou approximatives.

4.5.5. La Représentation des Connaissances

Un autre concept, qui est d'une grande importance pour tout système apprenant destiné à résoudre des problèmes complexes, est la détermination des connaissances à apprendre et leur mode de représentation. En effet la criticité de cet élément découle du fait que toutes les prises de décision ultérieures sont établies sur la base des connaissances apprises. Ainsi, pour tout système apprenant devant être capable de construction automatique de connaissances, nous devons au préalable:

- *Bien définir et spécifier les connaissances à apprendre et leur nature;*
- *Choisir un mode ou un langage de représentation qui permet de bien exprimer la nature des connaissances à apprendre, et de capter leur syntaxe et sémantique.*

Un formalisme de représentation des connaissances est décrit par un langage *L*, tel que:

$$L = \text{Syntaxe} + \text{Sémantique} + \text{Règles d'Inférences}$$

Les différents langages de représentation peuvent être regroupés dans l'une ou l'autre de ces deux grandes classes:

1. Les approches connexionnistes.
2. Les approches symboliques.

Les modes de représentation des connaissances connexionnistes sont inspirées des études sur le cerveau humain [Clément, 1989; Hopfield, 1982]. Cette représentation est fondée sur les neurones artificiels et leurs connexions et a donné naissance aux réseaux connexionnistes; le chapitre 2 a présenté l'apprentissage automatique à partir des structures connexionnistes, et le chapitre 3 a présenté des applications d'Agents pour lesquels ils sont bien adaptés. Les formalismes de représentation des connaissances symboliques les plus connus sont: les descriptions par liste d'attributs/valeurs; la logique d'ordre 0, d'ordre 1, la logique floue et d'autres logiques, les règles de production; les réseaux sémantiques, les frames, les schémas, la Représentation des Connaissances par Objets (R.C.O), la représentation hybride: Règles et Objets et les ontologies [Guarino et Giaretta, 1995; Haton, 1990; Igor, 1987 ; Kayser, 1997; Lavrac et Dzeroski, 1994; Masini et al., 1989; Minsky, 1975; Shirky, 2005].

Les formalismes de représentation, utilisés pour les systèmes à base de connaissances, les plus répandus sont : la logique, les règles de production et la R.C.O. La logique et les règles de production représentent les connaissances par des descriptions symboliques, proches du langage naturel humain, et ainsi les représentations sont facilement interprétables, d'où leur large utilisation, mais leur expressivité est limitée pour représenter les structures de contrôles complexes. La R.C.O est un formalisme qui permet de capturer la structure et le comportement des entités à modéliser et elle constitue une base pour la représentation par ontologies, qui est de nos jours, des plus recommandées. Une ontologie est une conceptualisation d'un domaine sous forme d'entités, d'attributs, de relations et d'axiomes, compréhensibles par les humains [Guarino et Giaretta, 1995; Shirky, 2005]. Les ontologies sont employées, dans les systèmes à base de connaissances, pour définir les termes et les relations de base dans le vocabulaire, du domaine à supporter; aussi bien que les règles pour combiner les termes et les relations pour des extensions du vocabulaire. Le formalisme de représentation des connaissances, pour un système apprenant est fortement lié au mécanisme d'apprentissage à utiliser, concept qui fait l'objet de la section suivante.

4.5.6 Une Approche /Méthodes d'Apprentissage

Maintenant, nous arrivons au concept le plus critique « *Une approche/Méthodes d'apprentissage* », sur lequel se base la qualité de l'apprentissage automatique et ainsi celle du système basé Agent-Apprenant. Nous l'avons nommé « *Approche/Méthodes d'apprentissage* » pour le fait que, pour un système basé Agent-Apprenant, le choix doit porter d'abord sur l'approche d'apprentissage puis sur une ou deux méthodes appartenant à l'approche choisie. Le choix de deux méthodes peut se faire afin d'opter pour la plus adéquate. Au chapitre 2, nous avons dégagé quatre approches de l'apprentissage automatique:

1. *Déductive.*
2. *Inductive.*

3. *par Renforcement.*

4. *Hybride.*

Chacune de ces approches comporte un certain nombre de méthodes d'apprentissage, dont les plus représentatives ont été présentées au chapitre 2. Notre modèle est sensé couvrir ces différentes approches ainsi que les méthodes qui leurs appartiennent. Et par conséquent garantir, aux systèmes basés Agent-Apprenant, des principes et des mécanismes leur permettant d'apprendre et de construire des connaissances ou des stratégies de contrôle, pour une meilleure résolution des problèmes à supporter et une bonne adaptation à l'environnement.

Les concepts suivants:

- *les exemples d'entraînement;*
- *les connaissances préalables;*
- *la représentation des connaissances;*
- *une approche /méthodes d'apprentissage;*

une fois instanciés, vont coopérer pour assurer un apprentissage automatique et permettre aux systèmes basés Agent-Apprenant, de bien supporter, chacun, sa « *classe des tâches* » et par conséquent améliorer la performance et s'auto adapter.

Une fois, la base conceptuelle, de notre modèle général et générique, définie; nous avons procédé à la décomposition et le raffinement de la vue globale du modèle.

4.6. Composants du Modèle des Systèmes basés Agent-Apprenant

A partir des fonctionnalités générales énumérées précédemment, de la vue globale du modèle et des concepts qui ont été définis à la section 4.5, nous avons abouti à un modèle général et générique de systèmes basé Agent-Apprenant, illustré par la figure 4.2 et constitué par l'ensemble des composants suivants:

- *Composant Interface.*
- *Composant Application.*
- *Composant Perception.*
- *Composant Bases d'Apprentissage.*
- *Composant Apprentissage automatique.*
- *Composant Raisonnement.*
- *Composant Connaissances.*
- *Composant Evaluation Performance.*

Nous tenons à préciser que nous utilisons le terme composant pour désigner les éléments ou les modules qui composent ou constituent un système général basé Agent-Apprenant générique. Nous utilisons une représentation graphique, où les composants sont représentés par des boîtes fermées et les liens entre boîtes représentent leurs entrées/sorties ainsi que les relations de dépendances entre composants. Au raffinement d'un composant, sa boîte est ouverte pour laisser apparaître les sous composants et les éléments issus de ce raffinement. Nous utilisons aussi la notation de classe et de cardinalité pour représenter l'aspect de généricité du modèle.

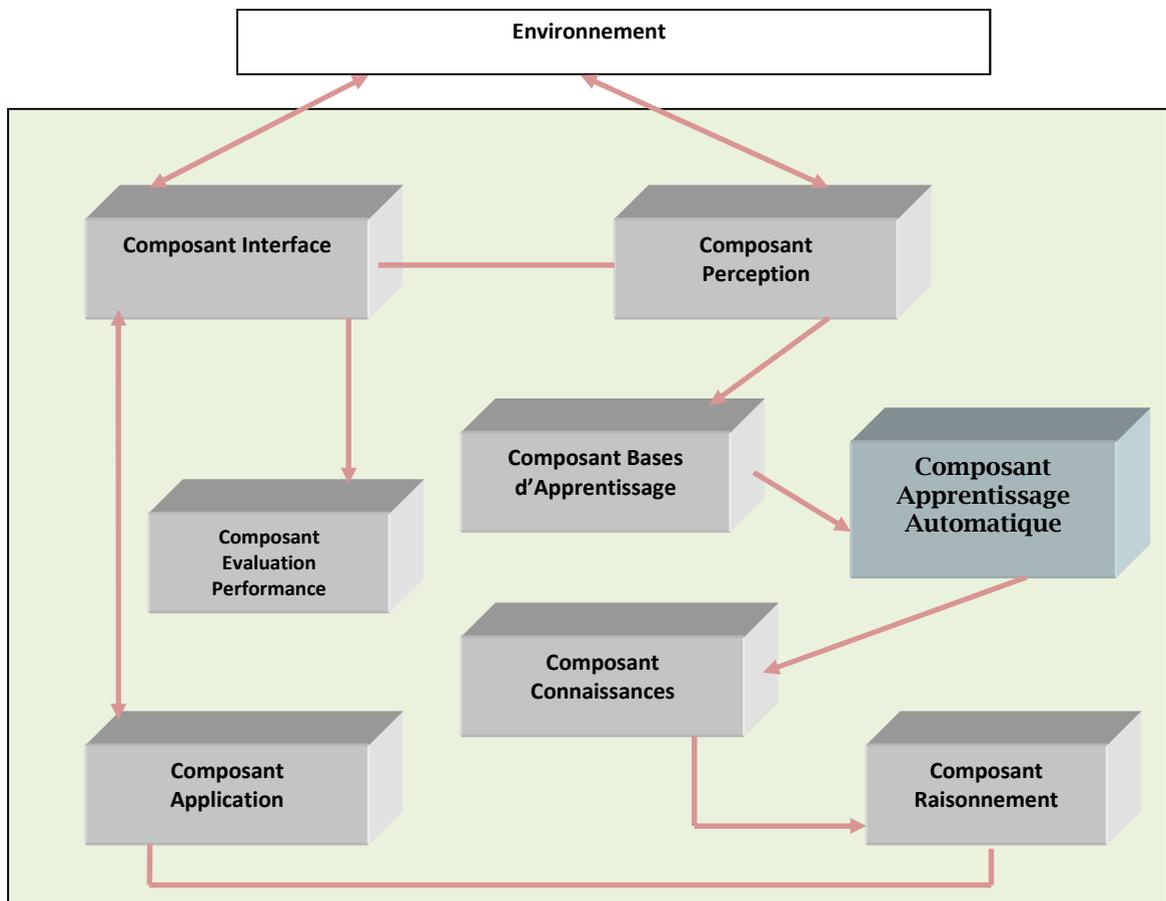


Figure 4.2: Composants du modèle général et générique de systèmes basés Agent-Apprenant.

4.6.1. Composant Interface

L'Interface est le composant d'interaction avec l'environnement pour lui permettre d'accéder aux services offerts par le système. Le plus important pour ce module, surtout si le système basé Agent-Apprenant est un logiciel, est l'ergonomie ainsi que la convivialité. Pour la robotique, l'interface peut être liée à des pièces électroniques, tels que des bras de manipulation d'objets ou autres. Ce composant est lié aux effecteurs d'un système basé Agent-Apprenant permettant, à ce dernier, d'interagir avec l'environnement.

4.6.2. Composant Application

Le module Application inclut la classe des tâches qu'un système basé Agent-Apprenant doit supporter ainsi que les autres services offerts par le système. Ainsi ce module est une boîte fermée dans le modèle général et il n'est décomposé qu'à l'instanciation de ce modèle. Tel qu'il est illustré par la figure 4.2, ce composant est en relation avec le module Raisonnement; qui est le responsable des décisions concernant la prise en charge de la classe des tâches du module d'application.

Le composant Application correspond à la partie actions d'un système basé Agent, présenté au chapitre 1 au niveau de la figure 1.1. En effet ce module permet de mettre en œuvre et d'exécuter les services, offerts par le système, ainsi que les actions décidées, par le composant de raisonnement. D'où la relation, illustrée par la figure 4.2, de dépendance entre le composant Application et le composant Raisonnement; qui correspondent à la décomposition du module Raisonnement et Action, Au niveau de la vue global du modèle.

4.6.3. Composant Perception

Le composant Perception a pour but de veiller à la perception et l'observation continues de l'environnement afin de capter toute action ou événement survenus dans ce dernier. Le composant Perception est d'une grande importance pour un système basé Agent, et la perception est une de ses fonctionnalités principales. En effet, un système basé Agent, est en relation avec son environnement, ainsi il doit le percevoir et l'observer afin d'agir en conséquence.

Pour un système basé Agent-Apprenant, le composant Perception est encore plus important, il permet de guetter et capter toute action ou événement qui peuvent apporter un plus au processus de l'apprentissage automatique. En effet toutes les données, constituantes des bases de l'apprentissage sont acquises par ce composant, et ainsi la perception et les processus qu'elle déclenche collaborent pour améliorer la performance d'un système basé Agent-Apprenant et sa capacité d'adaptation.

4.6.4. Composant Bases d'Apprentissage

L'apprentissage automatique nécessite toujours une base d'apprentissage pour démarrer son processus de construction de connaissances ou de stratégie de contrôle. Et comme notre modèle vise à être générique, le composant, Bases d'Apprentissage, est constitué, tel qu'il est illustré par la figure 4.3 suivante, de plusieurs éléments.

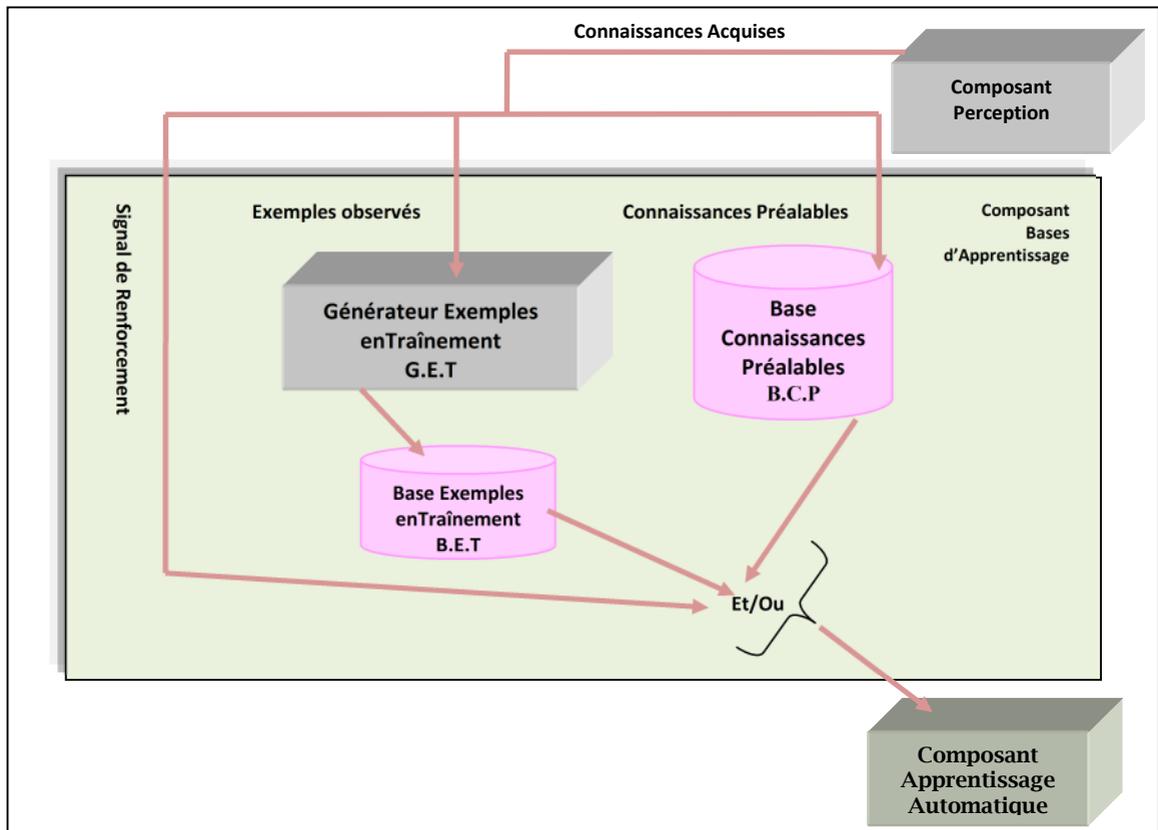


Figure 4.3 : Eléments du composant bases d'apprentissage.

Pour être générique, le modèle proposé doit couvrir les différentes approches de l'apprentissage automatique et par conséquent le composant Bases d'Apprentissage doit regrouper toutes les entrées possibles du module d'apprentissage, à savoir:

1. *Base d'exemples d'entraînement.*
2. *Base de connaissances préalables.*
3. *Signaux de renforcement.*

Ces éléments peuvent coexister comme ils peuvent être exclusifs, ceci est illustré sur la figure 4.3 par les opérateurs et/ou indiquant que deux ou trois entrées peuvent être combinées ou séparées. Tout dépend du domaine d'application et des approches d'apprentissage qui lui sont appropriées. Pour une approche d'apprentissage inductive, une Base d'Exemples d'enTraînement: B.E.T est nécessaire; en effet l'induction part d'un large volume de données spécifiques pour généraliser et apprendre des connaissances générales. Pour construire cette base, nous avons prévu un sous-composant dit «Générateur d'Exemples d'enTraînement: G.E.T». Le G.E.T reçoit en entrée les données observées, par le module Perception, pour les traiter et générer la B.E.T. Une fois cette base construite, le G.E.T se charge de sa maintenance et son enrichissement continu par les résultats des traitements des données perçues.

La Base des Connaissances Préalables (B.C.P) est à utiliser pour les applications disposant de théories de domaines préétablies. Cette base constitue l'entrée principale de l'apprentissage déductif, qui est le plus approprié aux domaines à connaissances préalables parfaites. La B.C.P est aussi une entrée pour l'apprentissage hybride, approprié aux domaines à connaissances préalables approximatives.

La troisième entrée dite: signal de renforcement correspond aux applications pour lesquelles l'apprentissage par renforcement est adéquat. Cette donnée est acquise par le module Perception et utilisée directement comme entrée du composant Apprentissage Automatique, qui est le sujet de la section suivante.

4.6.5. Composant Apprentissage Automatique

Le module Apprentissage Automatique, composant critique du modèle, reçoit la sortie du composant Bases d'Apprentissage, pour apprendre de nouvelles connaissances ou des stratégies de contrôle. Tel qu'il est illustré par la figure 4.2, la sortie du composant Apprentissage enrichit la base des connaissances qui est la source pour les prises de décisions nécessaires à la résolution des problèmes complexes. Le composant Apprentissage du modèle général et générique ne prédéfinit aucune approche ni méthode d'apprentissage, mais diffère la détermination de ces paramètres à l'étape d'instanciation du modèle. Ainsi ce composant peut être basé sur une ou plusieurs approches d'apprentissage; le digramme de classe suivant illustre la généricité des approches du composant Apprentissage Automatique du modèle proposé.

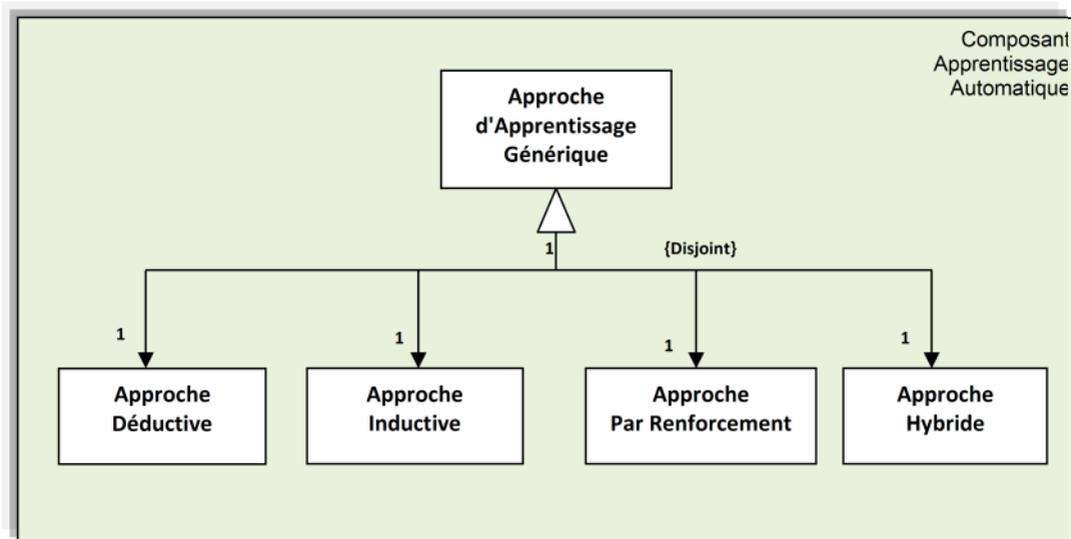


Figure 4.4 : Généricité des approches d'apprentissage du modèle proposé.

L'approche d'apprentissage, du composant principal de notre modèle, peut être spécialisée par l'une des quatre approches suivantes:

1. Approche Déductive.
2. Approche Inductive.

3. Approche par Renforcement.
4. Approche Hybride.

La spécialisation, schématisée sur la figure 4.4, possède une propriété «*disjoint*» indiquant que l'approche générique peut être spécialisée par une et une seulement des quatre approches. Cette notation nous permet de distinguer les différentes approches d'apprentissage tout en permettant leur combinaison dans l'approche hybride.

Toutes ces approches ont été présentées au chapitre 2 et utilisées dans la mise en correspondance décrite au chapitre 3; ici nous synthétisons leur principe et leurs méthodes. L'approche déductive est basée sur des théories de domaines; son but est la déduction, à partir d'une théorie, de connaissances plus opérationnelles. L'apprentissage déductif convient aux applications où des connaissances préalables parfaites sont disponibles; les méthodes utilisant l'explication forment la base de cette approche.

L'approche inductive est basée sur un processus de généralisation ou d'induction à partir d'une base riche d'exemples d'entraînement. Les méthodes de cette approche sont variées, les plus connues sont: l'apprentissage à base d'arbre de décision, les réseaux de neurones artificiels, les algorithmes génétiques, l'apprentissage à base de cas et la programmation logique inductive. Les méthodes inductives sont plus utilisées que les méthodes déductives, parce qu'il est possible de produire des exemples d'entraînement; mais pas de théorie de domaines.

L'apprentissage par renforcement est basé sur la maximisation d'un signal de renforcement retourné par l'environnement au système, appelé la récompense ou la pénalité. La méthode la plus connue, pour cette approche est le Q-learning. L'approche hybride permet de combiner deux ou trois approches ensemble. L'apprentissage déductif-inductif est souhaitable quand une théorie d'un domaine est disponible mais elle n'est pas parfaite; dans ce cas l'apprentissage déductif peut être augmenté par l'apprentissage inductif à partir des exemples d'entraînement. Le diagramme de classe suivant, illustre la généralité des méthodes, du composant Apprentissage Automatique.

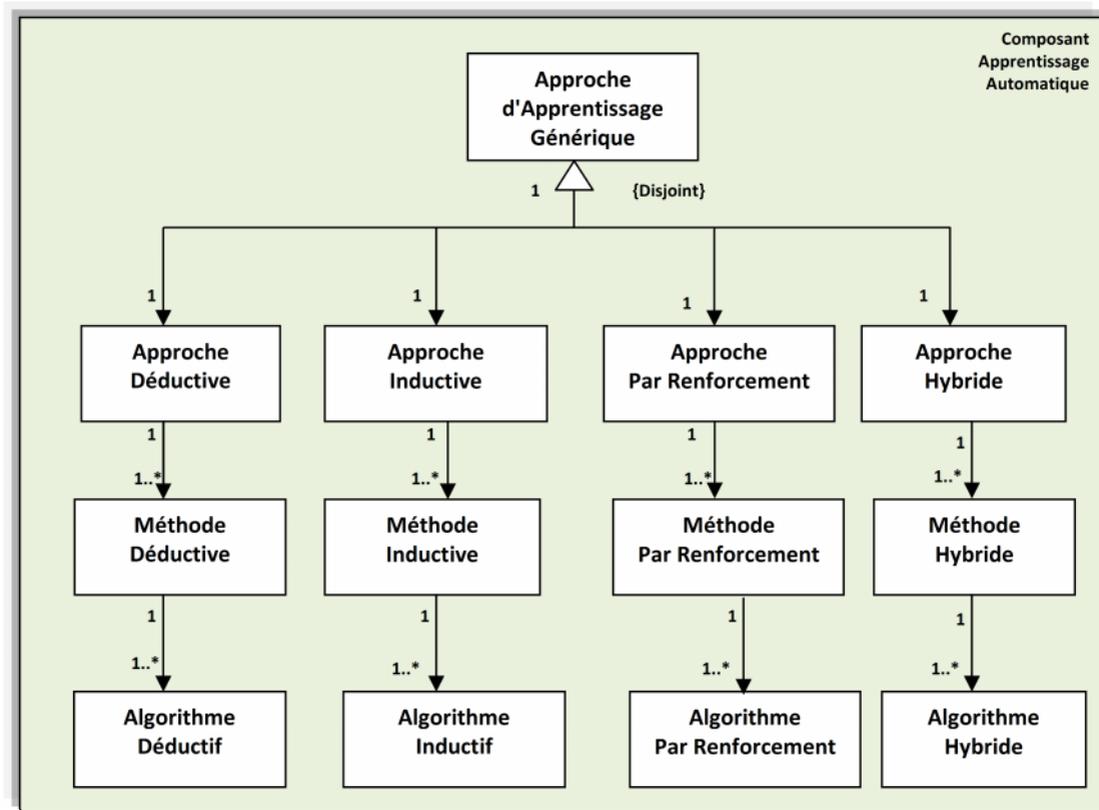


Figure 4.5 : Généricité des méthodes d'apprentissage du modèle proposé.

A l'instanciation du modèle générique, une approche puis une ou deux méthodes d'apprentissage sont à choisir; nous pensons que deux facteurs importants doivent être considérés:

1. *La nature de l'application du système basé Agent-Apprenant.*
2. *La disponibilité des exemples d'entraînement et/ou la disponibilité de la théorie du domaine.*

Ainsi le modèle proposé, pour les systèmes basés Agent-Apprenant, ne prédéfinit aucun des deux paramètres suivants:

- Formalisme de représentation des connaissances.
- Mécanisme d'apprentissage.

Ces deux paramètres sont différés à l'instanciation du modèle et l'expérimentation, ils dépendent du domaine d'application et de la nature des connaissances

4.6.6. Composant Connaissances

Ce composant, tel qu'il est illustré par la figure 4.2 constitue la sortie du module Apprentissage et l'entrée principale du module Raisonnement. En plus d'une Base de Connaissances, ce composant comporte aussi, tel qu'il est montré par la figure 4.6 suivante, un sous module Maintient Cohérence. Vu l'importance des connaissances pour

la résolution de problèmes, ce sous module se charge de la maintenance de la Base des Connaissances (B.C). Le maintien de la cohérence de la B.C peut, à notre avis, être automatique ou semi-automatique. Mais pour plus de fiabilité et de véracité des connaissances, nous préférons une maintenance semi-automatique. Nous avons illustré cette préférence, sur la figure 4.6, par l'interaction existante entre le composant des Connaissances et l'environnement. Un cognitivien ou un expert, dans le domaine d'un système basé Agent-Apprenant, peut aider le système dans la maintenance de la cohérence de la B.C. En effet, il est plus prudent de ne pas confier complètement, au système, cette tâche critique.

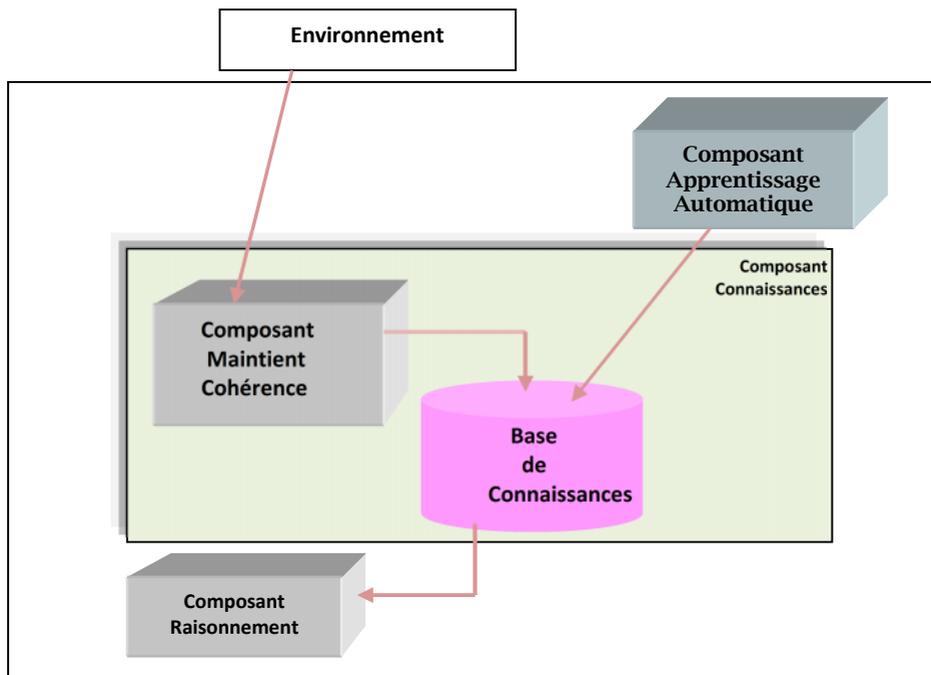


Figure 4.6 : Composant connaissances.

Pour la représentation des connaissances de la B.C, le modèle proposé ne prédéfinit aucun formalisme, mais diffère sa définition à l'instanciation. Le formalisme de représentation des connaissances dépend de la nature de ces dernières; ainsi il dépend du domaine d'application. C'est un facteur important, qui conditionne l'efficacité de l'apprentissage. En effet les deux paramètres: formalisme de représentation des connaissances et méthode d'apprentissage sont fortement liés; d'où leur détermination à la phase d'instanciation du modèle générique. La mise en correspondance, présentée au chapitre 3, entre les systèmes basés Agent-Apprenant et les approches/méthodes d'apprentissage permet de guider le choix, des deux paramètres précédents, à la phase d'instanciation du modèle général et générique.

4.6.7. Composant Raisonement

Ce composant assure le raisonnement et les décisions pour une prise en charge intelligente de la classe des tâches à supporter par un système basé Agent-Apprenant.

Après une prise de décision, par le composant Raisonement, ces tâches sont exécutées par le module Application, d'où la relation, montrée par la figure 4.2 entre ces deux composants.

L'intelligence et l'adaptation ne sont pas des propriétés d'un composant isolé, mais elles émergent de l'interaction entre le système basé Agent apprenant et son environnement, ainsi que des interactions entre les différents composants au sein du système. En effet, le composant Raisonement, du modèle général et générique, collabore avec les autres composants du système pour une bonne prise de décision. Cette dernière est le résultat des interactions du système basé Agent apprenant avec son environnement, via les composants Interface et Perception. Ainsi que, la collaboration des processus déclenchés suite à ces interactions, principalement le composant Apprentissage Automatique. En effet, ce dernier assure la construction et l'apprentissage des connaissances et des stratégies de contrôle formant la base et la source des prises de décision du système.

Le mode de raisonnement du composant des prises de décision dépend des formalismes de représentation des connaissances, et comme ces derniers sont variés, les modes de raisonnement sont aussi multiples. Le raffinement du composant de raisonnement ne fait pas partie de notre travail, mais nous notons qu'un travail de thèse de doctorat au sien de l'équipe « Génie Logiciel et Intelligence Artificielle » du laboratoire LIRE, a été réalisé [Kazar, 2005]. Ce travail permet à un Agent d'adapter son raisonnement au formalisme de représentation, des connaissances, utilisé par l'Agent à raisonnement adaptable. Au chapitre instantiation du modèle proposé, nous présenterons deux systèmes basés Agent-Apprenant pour deux domaines bien définis; où les composants de raisonnement seront détaillés avec les mécanismes d'inférences adéquats aux formalismes de représentation des connaissances utilisés.

4.6.8. Composant Evaluation Performance

Le composant d'évaluation de la performance d'un système basé Agent-Apprenant, n'a pas d'effet direct sur le comportement du système et ne fait pas partie de ses composants de base; sur la figure 4.2, ce module se trouve isolé des autres constituants du modèle. Le composant Evaluation Performance est prévu pour les concepteurs et les usagers d'un système, afin d'estimer et d'évaluer directement sa performance, par des mesures quantitatives. Mais en réalité les usagers peuvent se rendre compte de la performance d'un système donné par constatation de la qualité des services offerts. Les mesures quantitatives peuvent aider les concepteurs et les cognitiens d'un système basé Agent-Apprenant à prolonger l'entraînement du système. Une prolongation de l'entraînement d'un système Apprenant lui permet d'apprendre plus de connaissances, et par conséquent d'améliorer ses performances. Sinon, les concepteurs auront à réviser leurs choix concernant particulièrement: les formalismes de représentation des connaissances et surtout les mécanismes d'apprentissage automatique. A l'instanciation du modèle

générique, au chapitre suivant, ce composant sera instancié par des fonctions d'évaluation de la performance relatives aux domaines d'applications.

4.7. Conclusion

Ce chapitre a présenté le modèle général et générique, que nous avons proposé pour les systèmes basés Agent-Apprenant. Ce modèle a été construit selon les activités principales de la modélisation, qui ont guidé son établissement. Aussi un ensemble de concepts liés à l'apprentissage automatique, a été défini comme une assise conceptuelle pour ce modèle. Les éléments formels d'une définition de base de l'apprentissage automatique sont la source d'inspiration de cette assise conceptuelle; en effet, ces éléments ont été redéfinis et étendus par d'autres concepts pour renforcer les objectifs de généralité et de généricité du modèle proposé. Les fonctionnalités principales de systèmes basés Agent-Apprenant, ont été définies indépendamment de tout domaine d'application et de toute approche ou méthode d'apprentissage automatique. Ensuite à partir de ces fonctionnalités ainsi que de la base conceptuelle, les composants du modèle ont été déterminés. La définition des composants a été guidée par les objectifs de généralité et de généricité du modèle construit. Ces deux objectifs permettent au modèle d'être instancié par différents domaines d'applications et de couvrir toutes les approches et méthodes de l'apprentissage automatique.

Les composants généraux et génériques du modèle, ont été définis avec un raffinement de ceux correspondant à l'apprentissage automatique. En effet, le composant Bases d'Apprentissage fournissant les entrées nécessaires à l'apprentissage a été décomposé et raffiné en sous composants. L'aspect générique du composant Apprentissage Automatique a été explicité par la présentation des classes génériques et leurs spécialisations. Pour l'importance des connaissances apprises, un sous composant pour la maintenance de la cohérence de ces dernières est prévu. Le composant Evaluation Performance peut, entre autre, aider les cognitivistes d'un système basé Agent-Apprenant à prolonger son entraînement initial afin d'améliorer ses performances futures.

L'instanciation consiste à réutiliser le modèle proposé avec ses composants génériques, constituant la base des systèmes basés Agent-Apprenant, et l'étendre par les spécificités de l'application. Ainsi le développement d'un système basé Agent-Apprenant spécifique sera facilement réalisé, en effet la conception d'un tel système sera calquée sur le modèle général constituant un pattern générique, puis étendue aux spécificités du domaine. Par conséquent les phases d'analyse et de conception, qui ont un grand effet sur la fiabilité et la complétude des systèmes à développer, seront aisément et efficacement menées. L'instanciation du modèle proposé fera l'objet du chapitre suivant.

Chapitre 5

Instanciation du Modèle Général et Générique: Expérimentations et Etudes de Cas

5.1. Introduction

Nous avons construit notre modèle général et générique pour les systèmes basés Agent-Apprenant dans le but de proposer un «pattern» qui peut être étendu et instancié par différents domaines d'application. Le modèle proposé, présenté au chapitre 4, a énuméré les composants noyaux qui doivent être présents dans chaque système basé Agent-Apprenant, indépendamment de tout domaine d'application. L'instanciation du modèle générique, par un domaine donné, préserve ses composants noyaux et étend le modèle par les spécificités de l'application. La base conceptuelle, du modèle, que nous avons définie au chapitre 4, fait aussi partie de l'instanciation. Nous rappelons ici les concepts qui forment cette base:

- *La classe des tâches.*
- *Une mesure de performance.*
- *Les exemples d'entraînement.*
- *Les connaissances préalables.*
- *La représentation des connaissances.*
- *Une approche /méthodes d'apprentissage.*

Ces concepts constituent les paramètres d'instanciation du modèle projeté sur un domaine donné. Ces éléments sont en étroite relation et correspondent à plusieurs aspects d'un même Agent Apprenant. Leur instanciation et leur définition, pour un domaine donné, sont faites progressivement et parallèlement avec celles des composants du système basé Agent Apprenant en construction.

Dans un but d'exploitation et de réutilisation du modèle proposé, pour les systèmes basés Agent Apprenant, celui-ci a fait l'objet d'instanciations et d'expérimentations. Le chapitre en cours présente deux études de cas qui constituent aussi un moyen de mise en œuvre et de validation du modèle établi. La première instanciation concerne la prise en charge intelligente du courrier électronique, un moyen qui devient de plus en plus largement répandu. La seconde s'intéresse à un système basé Agent Apprenant pour le diagnostic des pathologies des plantes.

5.2. Instanciation du Modèle Général et Générique par le Domaine des E-mails

Les recours massifs à l'Internet, avec un coût d'utilisation sans cesse en diminution ont généré une croissance exponentielle du volume d'informations électroniques drainées. L'étendue de la diffusion de ces informations est devenue très large, et une conséquence de cette explosion est la difficulté de manipuler, de traiter ou d'exploiter toutes ces données. En effet ces tâches s'avèrent de plus en plus pénibles et fastidieuses, c'est le cas du courrier électronique, un moyen de communication devenu incontournable au vu de l'efficacité et de la rapidité qui le caractérisent. Les E-mails ont facilité les communications, mais leur nombre, leur source ainsi que leur contenu ont transformé notre comportement vis-à-vis de ces messages.

L'accès réitéré à une boîte e-mail, la lecture ou l'envoi de messages électroniques sont des manipulations consommatrices de temps, il s'avère donc essentiel de mettre en place des outils automatiques afin de soulager les usagers du net et économiser un facteur précieux et critique qui est le temps. Dans cet objectif, nous avons développé un système pour la notification et la prise en charge intelligente des E-mails. Ce système a été instancié à partir de notre modèle général et générique de systèmes basés Agent-Apprenant. Dans la suite de ce chapitre, nous allons nommer ce système l'I.M.N.H pour « **I**ntelligent **M**ails **N**otification and **H**andling». L'instanciation du modèle générique par le domaine des E-mails a engendré l'architecture de l'I.M.N.H, qui fait l'objet de la section suivante.

5.2.1. Architecture du Système basé Agent-Apprenant pour la Prise en Charge Intelligente des E-mails

L'instanciation du modèle général et générique par une application de prise en charge intelligente du courrier électronique a été établie à l'aide d'une succession

d'extensions et de raffinements du modèle général et sa projection sur le domaine des E-mails; le résultat de cette instanciation est l'architecture illustrée par la figure 5.1 suivante.

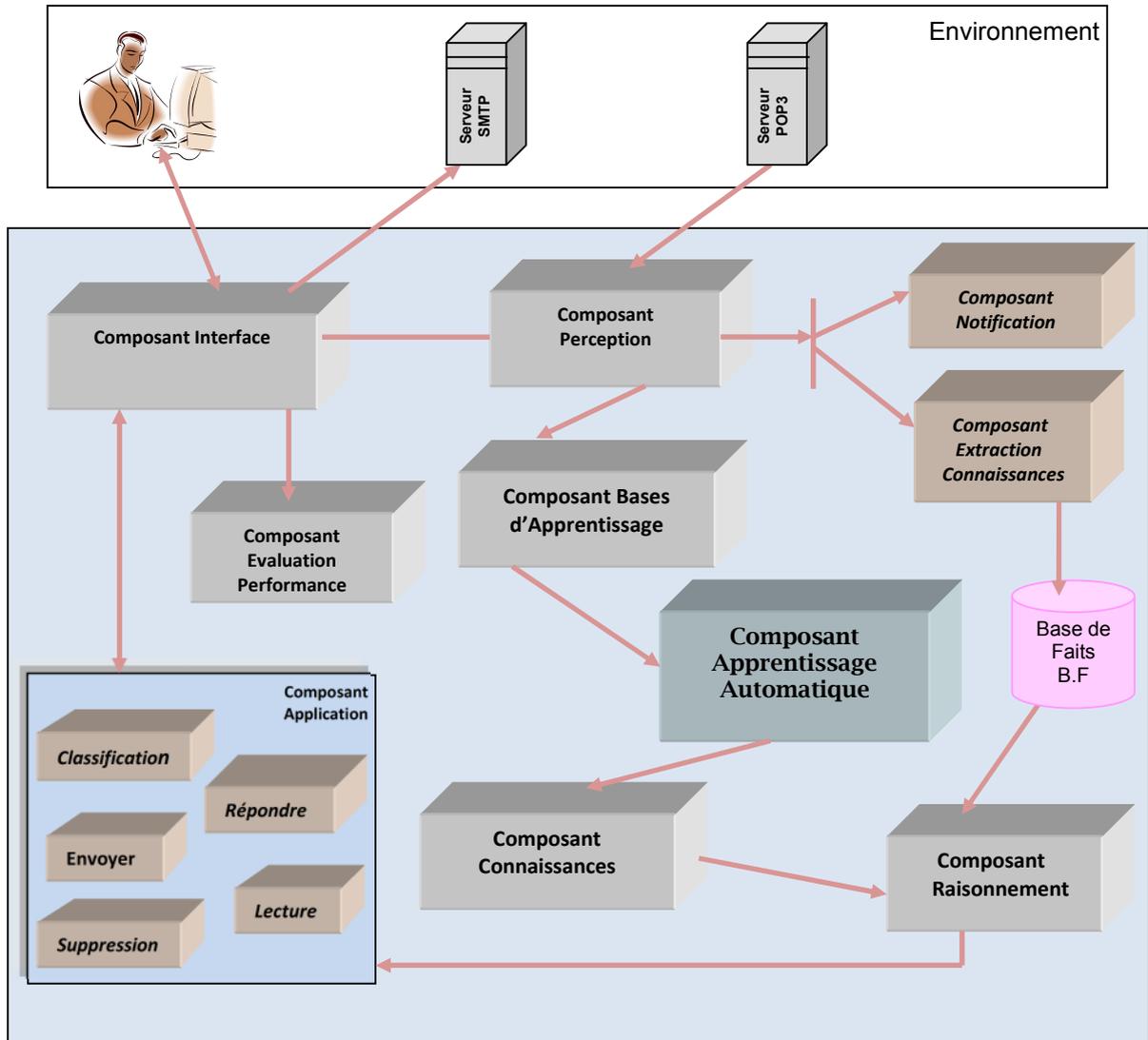


Figure 5.1: Architecture de l'Agent-Apprenant de prise en charge des E-mails.

D'après la figure 5.1, l'architecture de l'Agent-Apprenant, pour la prise en charge intelligente des E-mails, est une instanciation du modèle général et générique, présenté au chapitre 4. En effet les composants noyaux du modèle sont toujours présents dans cette architecture, avec des extensions et des raffinements relatifs au domaine de l'application qui ont été intégrés afin de créer une instance du modèle général, spécifique à l'application envisagée. Nous commençons par instancier l'environnement de ce système Apprenant; ensuite, nous présentons les différents composants ou modules de l'I.M.N.H avec l'instanciation des concepts de base qui leur sont associés.

5.2.1.1. Instanciation de l'Environnement

L'environnement, du système basé Agent-Apprenant de prise en charge intelligente du courrier électronique, est constitué, tel qu'il est illustré par la figure 5.1, des trois éléments suivants:

1. *L'utilisateur de l'I.M.N.H.*
2. *Le serveur POP3.*
3. *Le serveur SMTP.*

5.2.1.1.1. Usager de l'I.M.N.H

L'utilisateur de l'I.M.N.H, illustré par l'icône graphique de la figure 5.1, constitue l'utilisateur de ce système; la prise en charge intelligente des E-mails se fait pour un usager donné, celui qui utilise cette application. Ainsi le système développé est un Agent-Apprenant Personnel, son objectif final est la satisfaction de son usager par une adaptation et une personnalisation continue. Ces deux dernières habilités sont assurées par un processus d'apprentissage automatique, dont l'utilisateur est la source des données d'entraînement.

5.2.1.1.2. Serveurs POP3 et SMTP

Le système basé Agent-Apprenant, construit pour la prise en charge intelligente des E-mails, doit avoir un certain degré d'autonomie, dans l'accès, la récupération et l'envoi des E-mails, de son usager, indépendamment d'un autre mailing système (tels que Yahoo, Hotmail, Gmail, ...). Cette autonomie permet à l'I.M.N.H d'être complet et par conséquent évite à l'utilisateur le recours à d'autres systèmes de messagerie. Pour atteindre cet objectif, nous avons intégré dans l'architecture du système des E-mails deux protocoles de communication: POP3 et SMTP, illustrés par la figure 5.1.

POP3 est l'abréviation de «Post Office Protocol» avec sa version 3 [Myers et Rose, 1996], c'est un protocole de communication qui assure la réception des E-mails situés sur un fournisseur de messagerie électronique. SMTP est l'abréviation de «Simple Mail Transfer Protocol» [Postel, 1982] qui assure le transfert des E-mails vers un serveur de messagerie électronique, tels que Yahoo, Hotmail, Gmail, ... Le serveur de messagerie est à définir par l'utilisateur de l'I.M.N.H au début de son utilisation, et il peut être modifié si l'utilisateur désire changer sa boîte E-mails. Les deux protocoles de communication sont utilisés par le système via un ensemble de commandes qui assurent la connexion, la vérification de la boîte, la réception et l'envoi d'E-mails.

5.2.1.2. Composant Interface

Ce Composant ou module fait interface avec l'environnement du système en assurant l'interaction avec:

- Le serveur SMTP pour l'envoi des E-mails.
- L'utilisateur du système, pour lui offrir des services de manipulation d'E-mails.

- Le composant Perception, qui observe l'utilisateur pour une acquisition de connaissance.
- Le composant Application, qui met en œuvre les services offerts par le système.
- Le composant d'évaluation de la performance de l'I.M.N.H.

Nous présenterons quelques écrans de ce module dans les sections suivantes.

5.2.1.3. Composant Perception

Le composant Perception, tel qu'il a été défini par le modèle général et générique, à pour but de veiller à la perception et l'observation continues de l'environnement afin de capter toute action ou événement survenus dans ce dernier. Ainsi, pour l'I.M.N.H, ce composant permet d'observer l'utilisateur et le percevoir de façon continue afin de guetter et de capter n'importe quelle action, effectuée par cet usager. Aussi, le module Perception est à l'écoute du serveur POP3 pour la détection de l'arrivée de nouveaux E-mails, qui sont systématiquement notifiés à l'utilisateur. Par conséquent, en plus de l'observation et l'écoute continues de l'utilisateur et du serveur POP3, le composant Perception, tel qu'il est illustré par la figure 5.1, est chargé d'invoquer:

1. *Un composant Notification.*
2. *Un composant Extraction Connaissances.*

L'observation continue de l'utilisateur, par le composant Perception, permet de capter toutes les données formant l'entrée du composant Bases d'Apprentissage; d'où l'importance du composant Perception pour un système basé Agent-Apprenant.

5.2.1.4. Composant Notification

Il est devenu nécessaire pour les usagers d'Internet d'accéder plusieurs fois dans la même journée à leurs boîtes électroniques pour vérifier l'arrivée de nouveaux E-mails. De ce fait avoir un service de notification, qui de façon autonome et sans aucune intervention humaine nous informe au sujet de ces nouveaux E-mails reçus, est d'une grande importance. Dans cette perspective, notre système de prise en charge intelligente des E-mails est un Agent autonome, qui recueille automatiquement les nouveaux E-mails et opère la notification en temps réel. Cette capacité est assurée par l'intégration, dans l'I.M.N.H, du protocole de communication POP3 qui reçoit les E-mails situés sur le serveur de messagerie électronique de l'utilisateur. Ce service important informe systématiquement l'utilisateur au sujet de la réception de nouveaux E-mails, tel qu'il est illustré par la fenêtre ci dessous, même si la session de travail de l'utilisateur concerne une autre application.

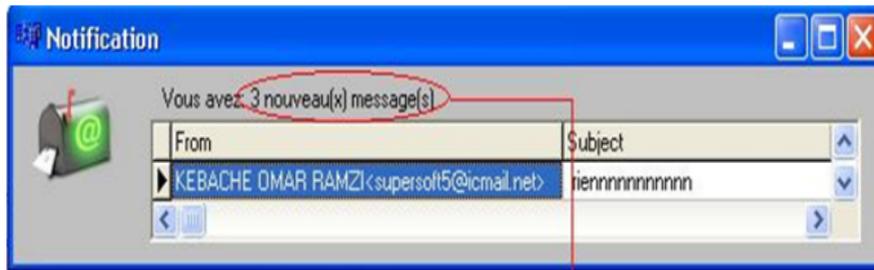


Figure 5.2: La notification de nouveaux E-mails au niveau de l'I.M.N.H.

5.2.1.5. Composant Extraction Connaissances

Afin de traiter automatiquement les nouveaux E-mails, par l'I.M.N.H, une extraction des connaissances à partir des E-mails reçus est nécessaire; ainsi, l'architecture, déduite du modèle général, doit aussi comporter un composant Extraction Connaissances. Ce module est basé sur des algorithmes d'élimination des mots communs ou mots vides à partir des E-mails reçus (tels que: le, la, les, de, des, un, une, à, pour...), pour permettre l'extraction de:

- *L'identificateur d'un E-mail.*
- *La partie « de » ou « From » d'un E-mail.*
- *et le « Sujet » d'un E-mail.*

5.2.1.6. Base de Faits

Les connaissances principales extraites de chaque nouvel E-mail, sont sauvegardées dans une base temporaire, donc les deux attributs: « From » et « Sujet » ainsi que leurs valeurs pour chaque E-mail reçu. La Base de Faits forme une entrée pour le module de raisonnement qui l'utilise durant son processus de prise de décision.

5.2.1.7. Composant Bases d'Apprentissage

Tel que nous l'avons présenté au chapitre 4, l'entrée du composant Bases d'Apprentissage du modèle général peut être instanciée par l'une des trois possibilités suivantes ou par leur combinaison:

1. *Des exemples observés.*
2. *Des signaux de renforcement.*
3. *Des connaissances préalables.*

Pour la prise en charge intelligente des E-mails, le système a pour objectif final d'atteindre un haut degré de personnalisation en imitant les habitudes et les pratiques de l'utilisateur, concernant sa prise en charge de ses E-mails. Ainsi le système doit capter toutes les manipulations d'E-mails, de la part de l'utilisateur, pour pouvoir apprendre à partir de celles-ci; ces manipulations constituent les *exemples observés* qui aident l'I.M.N.H à construire des profils de son usager.

L'entrée « *signaux de renforcement* », utilisée avec un apprentissage par renforcement, n'a pas de sens pour le domaine des E-mails et le type d'Agent Apprenant correspondant à l'I.M.N.H. En effet, ce système est basé sur un Agent Personnel, qui a pour rôle principal l'apprentissage et la construction automatique de connaissances pour une prise en charge intelligente des E-mails d'un usager donné.

Pour les *connaissances préalables*, il est clair que nous ne pouvons pas disposer de telles connaissances pour une prise en charge des E-mails. En effet, aucune connaissance n'est préétablie pour ce domaine. Les connaissances sont à construire automatiquement, par l'I.M.N.H, et elles dépendent des manipulations, préférences et habitudes de l'usager de ce système. Ainsi, L'entrée du composant Bases d'Apprentissage pour l'I.M.N.H est formée uniquement des *exemples observés*, tel qu'il est illustré par la figure 5.3 suivante.

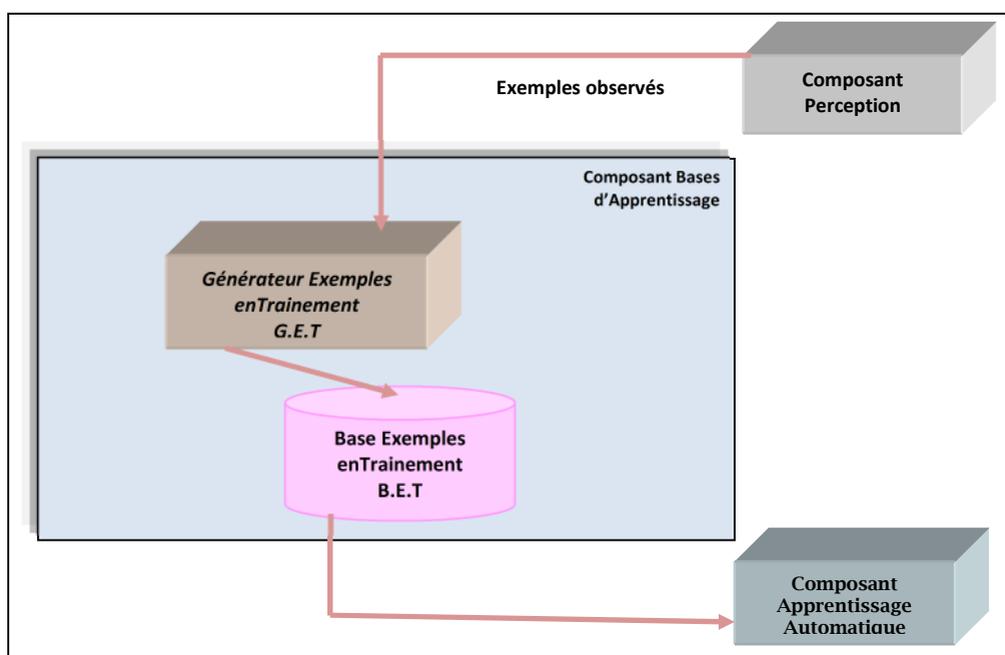


Figure 5.3 : Composant bases d'apprentissage de l'I.M.N.H.

Les *exemples observés* correspondent aux manipulations des E-mails, par l'usager, captées par le composant Perception. Par conséquent, l'I.M.N.H doit disposer d'un Générateur d'Exemple d'enTraining (G.E.T) pour construire l'ensemble *des exemples d'entraînement*. Cet élément crucial, faisant partie des concepts du modèle général, forme la base d'entraînement du processus d'apprentissage automatique pour une prise en charge intelligente des E-mails. Le G.E.T, de l'I.M.N.H, assure la construction automatique d'un ensemble d'exemples d'entraînement, par des combinaisons des différents attributs et des actions associées au niveau des exemples observés. Pour une prise en charge intelligente des E-mails d'un usager donné, la génération automatique des exemples d'entraînement, est possible. D'une part, les actions sur les E-mails n'ont pas d'effet néfaste, et d'autre part l'usager peut toujours refuser des décisions, de l'I.M.N.H, s'il le désire. Aussi, pour un Agent Apprenant Personnel qui se base sur les manipulations

de l'utilisateur, pour l'apprentissage de ses profils, seuls les exemples observés peuvent participer à la génération de la B.E.T.

5.2.1.8. Composant Application

Ce module se compose des services, illustrés par l'architecture de l'I.M.N.H, de prise en charge des E-mails et leur manipulation. Ces services correspondent, en partie, à l'instanciation du concept « *classe des tâches* » du modèle générique. En réalité ce concept fut le premier à être instancié pour partir d'une définition et d'une spécification précise des tâches que ce système basé Agent-Apprenant doit supporter. En effet, pour tout système Apprenant, il faut commencer par recenser les tâches qui lui sont confiées; mais ici, les instanciations des concepts sont présentées conjointement à celles des composants auxquels ils sont associés. De ce fait, concernant l'I.M.N.H, la classe de tâches énumérées pour la prise en charge intelligente des E-mails est:

- *Lecture.*
- *Classification.*
- *Suppression.*
- *Répondre.*

Le composant Application offre aussi un service d'envoi de courrier permettant à l'utilisateur d'envoyer des E-mails, personnellement à travers l'I.M.N.H et sans recourir à d'autres serveurs de messagerie. Les prises de décision concernant les tâches à entreprendre de façon autonome, par l'I.M.N.H, sont améliorées continuellement selon les entraînements et l'apprentissage de ce système basé Agent-Apprenant. Les services du composant Application sont invoqués selon deux manières:

1. D'une part, ils sont invoqués automatiquement par le système, selon les décisions du module Raisonnement. Les E-mails sont, ainsi, pris en charge intelligemment selon les inférences de ce module; d'où le lien entre les deux composants: Application et Raisonnement.
2. D'autre part, pour assurer une certaine liberté à l'utilisateur et garantir l'apprentissage de ses profils, l'utilisateur peut également invoquer les services précédents, s'il le souhaite; il peut donc consulter et manipuler personnellement ses E-mails, ceci est illustré par les deux écrans suivants de l'I.M.N.H. Cette manière d'invoquer des services constitue un apport primordial pour l'apprentissage automatique. En effet, la manipulation personnelle des E-mails par l'utilisateur permet au système, qui est en perception continue, d'acquérir les pratiques, les habitudes et les préférences de son utilisateur concernant la prise en charge des E-mails.

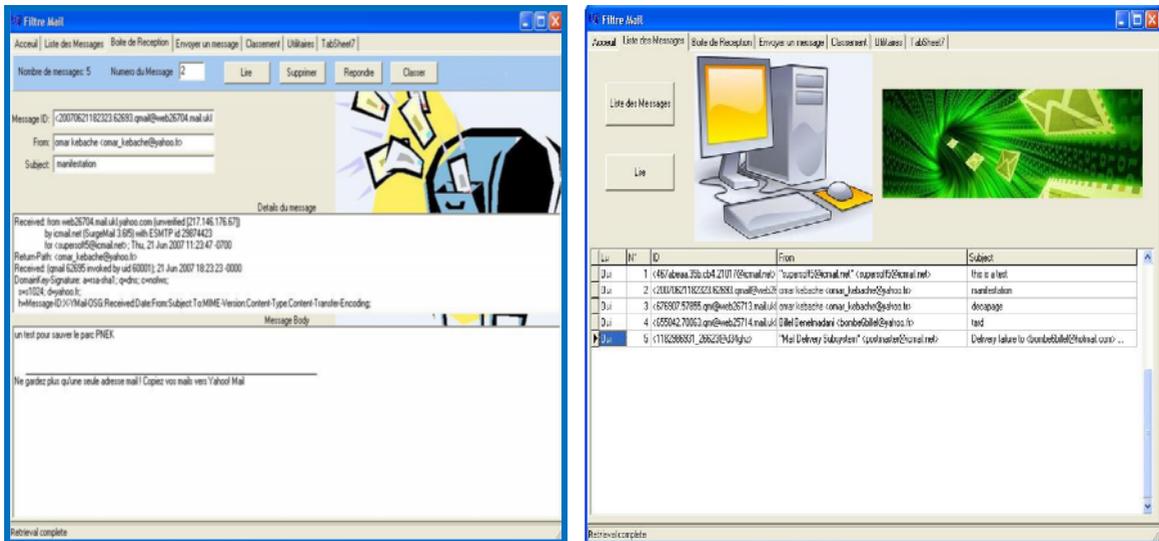


Figure 5.4: Ecrans du système de prise en charge des E-mails.

Le composant Raisonnement est le responsable des prises de décision concernant la prise en charge intelligente des E-mails; pour expliciter son fonctionnement, nous présentons au préalable l'entrée principale de ce module correspondant au composant suivant.

5.2.1.9. Composant Connaissances

Pour l'I.M.N.H, le Composant Connaissances (C.C) est principalement constitué d'une Base de Connaissances-E-mails (B.C E-mails) décisive et importante. Cette dernière est construite puis enrichie de façon incrémentale par le composant d'apprentissage automatique. En partant du composant des connaissances, du modèle général, et tel qu'il est illustré par la figure 5.5, le C.C comporte aussi un sous-composant de maintien de la cohérence. Ce dernier est un simple module de vérification, permettant à l'utilisateur d'inspecter la bonne signification des connaissances. L'élément crucial du C.C, la B.C E-mails, qui se compose des connaissances apprises représentées, pour l'I.M.N.H, par un ensemble de règles de production. Ainsi, le concept « *représentation des connaissances* » du modèle général et générique a été instancié, pour le domaine des E-mails, par le formalisme des règles de production qui s'adapte bien à ce domaine. Et pour la relation étroite entre le formalisme de représentation des connaissances et le mécanisme d'apprentissage, l'adéquation de ce formalisme à l'application des E-mails sera détaillée à l'instanciation et la description du composant d'apprentissage.

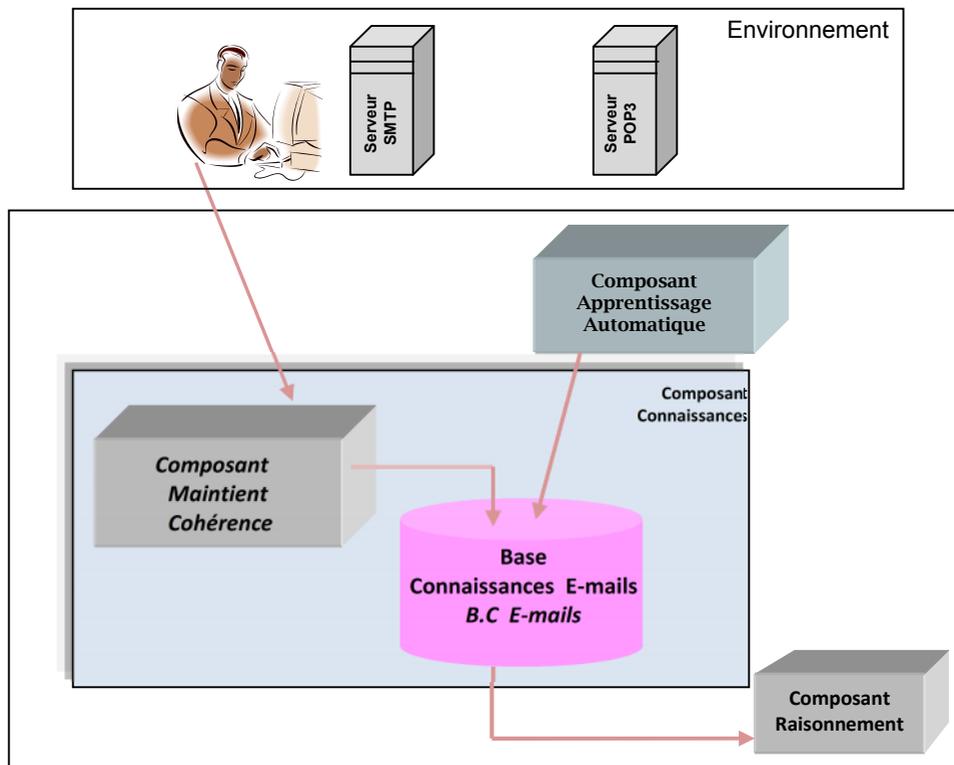


Figure 5.5: Composant connaissances de l'I.M.N.H.

Les connaissances apprises, par l'I.M.N.H, représentées par un ensemble de règles de production sont de la forme:

Si *From* = la valeur I et *Sujet* = valeur J alors *action* K avec DC
 "From et Sujet" sont les attributs des E-mails, l'action K est l'opération à effectuer intelligemment par le système sur un nouvel E-mail. DC est le degré de confiance d'une règle, c'est un scalaire représentant l'appréciation (le feedback) de l'utilisateur sur la décision intelligente entreprise par le système basé Agent-Apprenant, en particulier par son composant de raisonnement.

5.2.1.10. Composant Raisonnement

L'instanciation en cours, du modèle général, étant un système basé-Agent Apprenant Personnel; qui de façon autonome et intelligente manipule les E-mails de son usager en invoquant les services du module Application, illustrés par l'architecture de l'I.M.N.H et ses écrans précédemment présentés. Avec ce système basé-Agent Apprenant Personnel, en parallèle à la notification de nouveaux E-mails, un processus de prise de décision est lancé pour prendre en charge les E-mails reçus. Ainsi, pour manipuler intelligemment les E-mails, à leur réception ils sont systématiquement notifiés à l'utilisateur. D'autre part et de manière simultanée, le système opère une extraction des connaissances significatives pour invoquer le module Raisonnement. En se basant sur les connaissances apprises, au niveau de la B.C E-mails, et la base de faits (B.F), des décisions intelligentes sont

automatiquement prises pour la manipulation des E-mails reçus. Les décisions concernent les opérations à effectuer sur les nouveaux E-mails. La figure 5.6 présente le fonctionnement du composant Raisonement pour la prise en charge intelligente des E-mails.

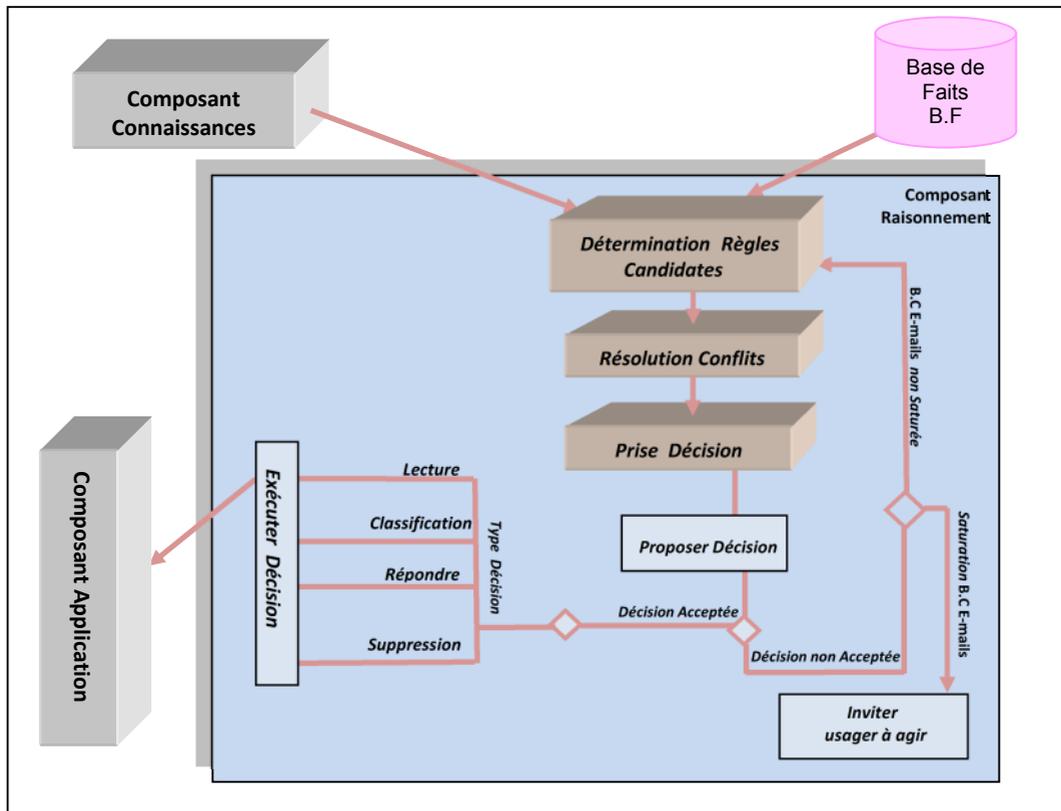


Figure 5.6: Fonctionnement du composant de raisonnement de l'I.M.N.H.

Le moteur d'inférence de ce système intelligent emploie un raisonnement déductif [Igor, 1987; Russel et Norvig, 2006] selon un chaînage avant ou un raisonnement guidé par les données, avec ses trois phases suivantes:

1. *Détermination de règles candidates ou de règles applicables.*
2. *Résolution de conflits entre les règles candidates.*
3. *Prise de décision.*

Pour la résolution de conflits, nous avons employé la stratégie ou l'heuristique suivante: "choix de la règle du plus haut degré de confiance"; ce qui renforce l'adaptation de l'I.M.N.H aux préférences de son usager. L'opération à exécuter, éventuellement, sur un nouvel E-mail est la partie «Action» de la règle choisie. Les règles de production sont de la forme : " Si conditions Alors Action", avec Action une opération qui peut être une lecture, une classification, une réponse ou une suppression de l'E-mail reçu. Cette action est réellement exécutée, par le composant Application, si et seulement si l'utilisateur accepte la prise de décision (donc l'inférence suggérée); autrement une autre règle de la B.C E-mails est essayée jusqu'à la satisfaction de l'utilisateur ou à la saturation de cette base (aucune

règle ne peut être appliquée). Dans ce dernier cas l'utilisateur est invité à effectuer l'action qu'il désire.

Pour accélérer le processus de raisonnement, notre système effectue un tri préalable de la base des connaissances; ce tri met la règle de plus haut degré de confiance à la tête de cette base. Par conséquent, la complexité de la phase de résolution de conflits est réduite au maximum et les inférences suggérées sont ainsi également accélérées.

5.2.1.11. Composant Apprentissage Automatique

Nous arrivons au composant le plus critique, en effet l'I.M.N.H étant instancié à partir de notre modèle générique des systèmes basés Agent-Apprenant, le module Apprentissage Automatique est le composant noyau de ce modèle. Pour ce composant nous commençons par instancier deux concepts importants:

1. *La représentation des connaissances.*
2. *L'approche/méthodes d'apprentissage.*

Nous évoquons aussi, pour ce composant, les concepts des «*connaissances préalables*» et celui des «*exemples d'entraînement*». D'où l'importance du composant d'apprentissage où quatre concepts du modèle générique sont traités, nous présentons donc l'instanciation de ces concepts, pour ce système, avant de décrire son processus d'apprentissage.

Pour l'I.M.N.H, il s'agit d'apprendre et de construire automatiquement des connaissances pour une prise en charge intelligente du courrier électronique d'un usager donné. Ainsi, l'élément à déterminer au préalable correspond aux connaissances à apprendre et le formalisme de leur représentation. Pour l'I.M.N.H, la connaissance à apprendre reflète les habitudes de l'utilisateur, ses pratiques et ses préférences au sujet de la manipulation des E-mails. Par exemple le système apprend que pour un E-mail reçu «*from*» une personne ou une organisation donnée doit être lu d'urgence. Cet apprentissage est une généralisation à partir des observations et perceptions de l'usager. Les manipulations des E-mails, par l'usager, sont captées et utilisées pour générer une *Base d'Exemples d'entraînement* (B.E.T), qui correspond à la source du processus d'apprentissage. La B.E.T est l'entrée du composant Apprentissage Automatique qui lui permet de pouvoir généraliser et apprendre des connaissances générales et personnelles de manipulation des E-mails, représentant des profils de l'usager.

En partant de la mise en correspondance établie entre les systèmes basés Agent-Apprenant et les approches/méthodes de l'apprentissage automatique, présentée au chapitre 3, l'apprentissage, pour l'I.M.N.H doit être inductif et par conséquent le concept «*approche d'apprentissage*» du modèle générique, est instancié ici par une approche d'apprentissage inductive. Nous ne disposons pas d'une théorie du domaine, parce qu'il n'y a aucune connaissance prédéfinie pour la manipulation des E-mails. Chacun a ses préférences et

ses habitudes pour prendre en charge ses E-mails, et pour la même personne les préférences peuvent changer au cours du temps, par conséquent pour le domaine des E-mails, le concept de «*connaissances préalables*» n'a pas d'instanciation.

En se basant sur l'étude, présentée au chapitre 2, l'approche inductive inclue une pléthore de méthodes d'apprentissage. Il fallait donc choisir entre différentes techniques pour instancier le concept «*méthodes d'apprentissage*». A partir des critères, présentés au chapitre 3, permettant de guider le choix, d'une méthode d'apprentissage inductive, nous avons opté pour la méthode des arbres de décision inductifs. Et plus précisément un algorithme, basé sur cette méthode, dit Induction Decision Tree ID3 de Quinlan [Quinlan, 1986; Nilsson, 1996] a été utilisé pour l'instanciation du concept «*méthodes d'apprentissage*», pour l'I.M.N.H. Ce choix est motivé en grande partie par la nature de l'application de la prise en charge des E-mails et des connaissances à apprendre et leur représentation. Pour cette application, il s'agit des attributs « From », « Sujet » d'un E-mail et « Action » (lecture, classification, répondre ou suppression) à effectuer sur cet E-mail. Les éléments « From » et « Sujet » ainsi que « Action » sont naturellement représentés par des attributs et leurs valeurs. Exemple:

From = Laboratoire LIRE *Sujet* = Séminaire *Action* = Lecture

Et comme la méthode des arbres de décision et, par conséquent, l'algorithme ID3 sont basés sur une approche d'apprentissage inductive d'une part, et d'autre part sur une représentation symbolique des connaissances avec attribut/valeurs [David, 2006; Nilsson, 1996]; nous les avons trouvés appropriés pour l'apprentissage au niveau de l'I.M.N.H. Aussi à partir d'ID3, nous pouvons générer, pour les connaissances apprises, des règles de production, un formalisme qui s'adapte bien à la représentation des connaissances apprises pour cette application. En effet, les connaissances à apprendre sont naturellement représentées par des règles similaires aux règles de production, qui en plus sont aisément interprétées par l'utilisateur du système. En outre ID3 est un procédé simple et efficace de classification et de généralisation [Quinlan, 1986; Nilsson, 1996].

Ainsi, pour la prise en charge intelligente des E-mails, l'I.M.N.H se base essentiellement sur un apprentissage des pratiques, habitudes et préférences de son usager. Le composant Perception est en observation continue de l'utilisateur afin de guetter et capter toute action de ce dernier. Le processus d'apprentissage de l'I.M.N.H, illustré par la figure 5.7 suivante, a pour entrée la Base des Exemples d'Entraînement (*B.E.T*) du composant Bases d'Apprentissage et pour sortie les règles de production générées à partir de l'arbre appris. Ces dernières enrichissent la base de connaissances, qui est la source des inférences intelligentes assumées par le composant Raisonnement.

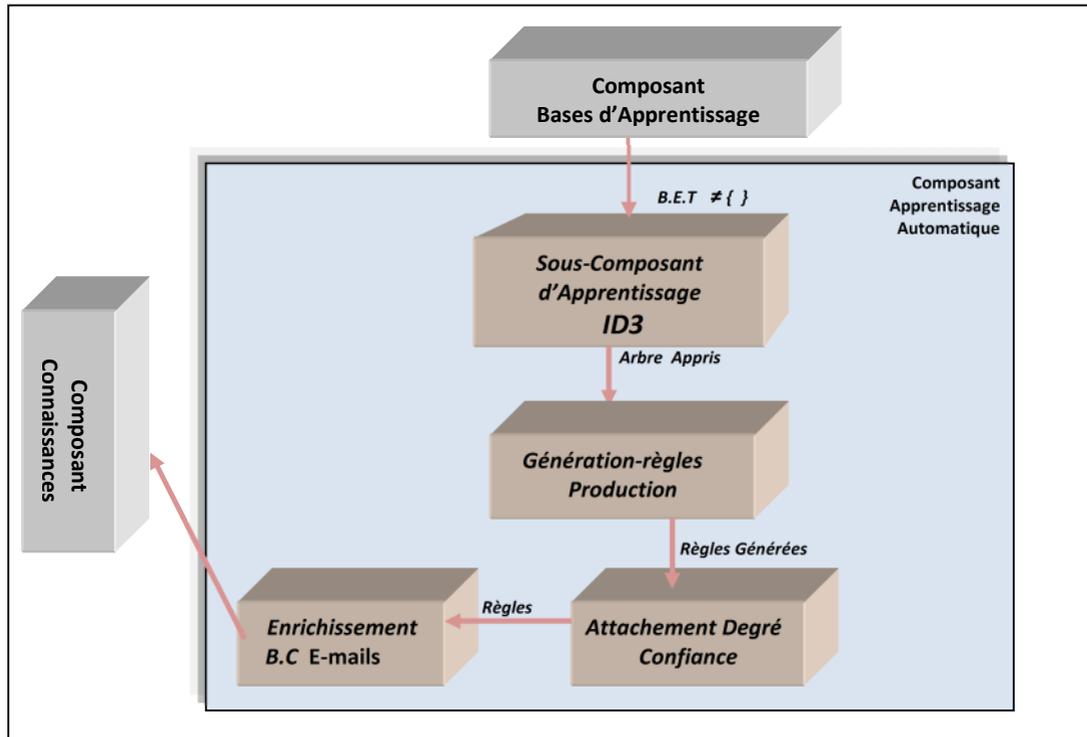


Figure 5.7: Composant d'apprentissage de l'I.M.N.H.

Maintenant nous explicitons le principe du sous composant d'apprentissage ID3.

5.2.1.11.1. Principe de Fonctionnement du Sous Composant ID3

Partant d'une base d'exemples d'entraînement ($B.E.T$), le sous composant d'apprentissage ID3 induit un arbre de décision en le construisant de haut en bas tout en sélectionnant, à chaque nœud, l'attribut qui classe le mieux les éléments de la $B.E.T$. L'arbre de décision est construit récursivement, ID3 utilise une structure de données bien adaptée à son principe de fonctionnement; en effet, les arbres sont naturellement récursifs. Pour le choix du meilleur attribut, ID3 emploie la mesure de l'entropie de Shannon [Quinlan, 1986; Nilsson, 1996], qui estime la pureté d'une collection d'exemples d'entraînement, quand ils sont partitionnés selon l'attribut sélectionné. Le meilleur attribut a la valeur d'entropie minimale parmi celles calculées; ainsi, à chaque étape de récursion dans ID3, l'attribut choisi avec son incertitude minimale, assure une bonne classification des exemples d'entraînement.

Avec ce procédé de sélection, ID3 permet d'avoir un arbre de décision concis et suffisant, en effet il ne suffit pas de traiter les attributs, des exemples d'entraînement ($B.E.T$), séquentiellement. Au contraire, toute la richesse de cet algorithme consiste à choisir judicieusement les attributs d'éclatement pour aboutir, par le chemin le plus court et nécessaire, au plus grand nombre d'exemples de la même classe. L'utilisation de l'entropie minimale, par ID3, permet de construire l'arbre de décision le plus efficace et le plus

petit, en suivant le principe "small is beautiful" [Quinlan, 1986; David, 2006]. Ce principe de simplicité garantit l'efficacité de l'algorithme ID3.

Nous présentons les formules de l'entropie, après le pseudo-code suivant, illustrant la fonction récursive d'ID3. Le paramètre de cet algorithme est la base $B.E.T = \{e/e = \text{exemple d'entraînement}\}$; et le type de cette fonction est un nœud de l'arbre appris, indiquant le nœud racine retourné par cet algorithme.

Fonction ID3 (B.E.T): arbre appris

```

Début
  Si B.E.T = {} (* {} est l'ensemble vide *)
  Alors
    Return Null
  Sinon
    Si  $\forall e \in B.E.T$  et e fait partie de la même classe
    alors
      Return un nœud contenant tous les exemples
    Sinon
      A  $\leftarrow$  attribut avec entropie minimale (* l'attribut A sera utilisé pour le partitionnement *)
      Vals  $\leftarrow$  liste des valeurs possibles pour A
      Pour aj  $\in$  Vals faire
        Part [aj]  $\leftarrow$  exemples qui ont aj comme valeur pour A
        Fils [aj]  $\leftarrow$  ID3 ( Part[aj] )
      Fin pour
      Return racine de l'arbre appris.
    Fin si
  Fin si
Fin.

```

Pour le partitionnement courant et pour chaque attribut non utilisé, la fonction ID3 calcule son incertitude ou sa valeur d'entropie. L'incertitude sur une classe C (parmi n classes) d'un exemple e avec la valeur aj pour un attribut A , est notée $Incert$; elle est calculée par Eq. (1):

$$Incert (C / aj) = - \sum_{i=1}^n (freq (ci / aj) * \log_2 freq (ci / aj)) \quad \text{Eq. (1)}$$

Dans Eq. (1), $freq$ est la fréquence ou la proportion d'exemples ayant la valeur aj pour l'attribut A , et $Incert (C/aj)$ est l'entropie de l'attribut A , tel que aj est sa valeur. Pour l'I.M.N.H, nous avons $n = 4$, le nombre de classes possibles: classification, lecture, répondre et suppression d'E-mails. Chaque attribut peut avoir différentes valeurs, c'est le cas des attributs « From » et « Sujet » dans un ensemble d'exemples d'entraînement; par conséquent, nous calculons l'incertitude moyenne sur la classe C pour toutes les valeurs de l'attribut A . L'entropie moyenne sur la classe C d'un attribut A avec m valeurs possibles est calculée par Eq. (2) :

$$Incert(C / A) = - \sum_{j=1}^m freq(aj) * \sum_{i=1}^n (freq(ci / aj) * \log_2 freq(ci / aj)) \quad \text{Eq. (2)}$$

Ainsi, l'attribut A , sélectionné pour le partitionnement des exemples, est celui dont l'incertitude est la plus petite valeur calculée par Eq. (2). La mesure d'entropie permet à ID3 d'apprendre le plus court arbre [Quinlan, 1986] que nous convertissons en un ensemble de règles de production.

5.2.1.11.2. Génération des Règles de Production

Après la construction d'un arbre de décision, le composant Apprentissage Automatique de l'I.M.N.H invoque un processus de génération de règles de production, tel qu'il est illustré par la figure 5.7. Les décisions d'un arbre sont toujours liées à une séquence de parcours, et chaque chemin dans l'arbre correspond à une règle de production de la forme: "Si *conditions* Alors *conclusion*". Exemple:

Si From=*Conférence ISA* et Sujet=*Acceptation* Alors *Lecture*

Ainsi l'arbre est converti en un ensemble de règles qui lui sont équivalentes, en produisant une règle pour chaque chemin de la racine jusqu'à une feuille. Les connaissances apprises par l'I.M.N.H sont alors représentées par des règles de production, un formalisme qui s'adapte bien à cette instanciation. Les connaissances relatives aux E-mails sont naturellement exprimées sous forme de règles, d'une part et d'autre part, ce formalisme permet:

- *La compréhension, par l'usager, des connaissances apprises, ainsi leur fiabilité peut être vérifiée et par conséquent la véracité du système est améliorée.*
- *Le raisonnement déductif, à partir des règles générées, qui se trouve approprié aux inférences de l'I.M.N.H.*

Nous arrivons maintenant à la présentation de l'instanciation du concept, «*une mesure de performance*» associé au composant de performance que nous avons différé jusqu'à la fin des présentations des différentes instanciations pour le fait que l'amélioration de la performance représente la finalité d'un système apprenant.

5.2.1.12. Composant Evaluation Performance

L'instanciation en cours concerne les concepts de base et les composants de notre modèle général et générique des systèmes basés Agent-Apprenant. La base conceptuelle de ce modèle comporte des éléments formels de la définition de l'apprentissage automatique, émise par Tom Mitchell, et avant de définir la performance de l'I.M.N.H, nous rappelons cette définition.

« *Un programme informatique apprend à partir de l'expérience E par rapport à une classe de tâches T et une mesure de performance P , si sa performance à l'exécution de tâches de T , mesurée par P , s'améliore avec l'expérience E » [Mitchell, 1997; Mitchell, 2006].*

Ainsi la performance d'un système apprenant est le facteur clé à améliorer à l'aide de l'apprentissage automatique. Rappelons aussi la spécification générale, que nous avons associée à la mesure de performance P , au niveau des définitions des concepts du modèle proposé.

P = une fonction des variables du système ayant une relation avec les résultats obtenus concernant la prise en charge des tâches T

Pour l'I.M.N.H, nous avons instancié ce facteur par une mesure adéquate, qui correspond au degré de satisfaction de l'utilisateur par rapport à la prise en charge intelligente des E-mails par ce système. Ainsi la performance, pour l'I.M.N.H, est le taux de satisfaction de l'utilisateur concernant les prestations offertes par ce système, c'est-à-dire les inférences intelligentes satisfaites, autrement dit les suggestions du système acceptées par l'utilisateur. Par conséquent, ce concept a été instancié par une mesure P définie par le quotient du nombre de suggestions acceptées sur le nombre d'inférences totales.

$$P = \text{Nbr Sug-Sat} / \text{Nbr Sug-Tot}$$

Avec :

Nbr Sug-Sat = Nombre de Suggestions Satisfaites.

Nbr Sug-Tot = Nombre de Suggestions Totales.

Ces deux variables, de l'I.M.N.H, sont en relation avec les résultats obtenus concernant la prise en charge des tâches T, instanciée par la prise en charge intelligente des E-mails. En effet, ces variables correspondent aux nombre de suggestions totales, proposées par l'I.M.N.H, et le nombre de suggestions acceptées par l'utilisateur. Et la fonction P correspond au quotient de ces deux variables.

L'amélioration de la performance de l'I.M.N.H est corrélée avec la satisfaction de l'utilisateur et la mesure P s'améliore au cours du temps par apprentissage automatique. La prise en charge intelligente des E-mails, s'améliore graduellement en fonction des entraînements du système et de l'apprentissage des connaissances représentant les pratiques, habitudes et préférences de l'utilisateur. Ces dernières constituent l'expérience E qui aide l'I.M.N.H à améliorer sa performance pour la prise en charge des tâches T, correspondants aux opérations à effectuer sur les E-mails reçus.

L'évaluation de la performance de l'I.M.N.H a consisté en une série de tests d'évaluation de la mesure P définie ci dessus. Les graphes suivants présentent des séries d'évaluation de la performance, P , de l'I.M.N.H.

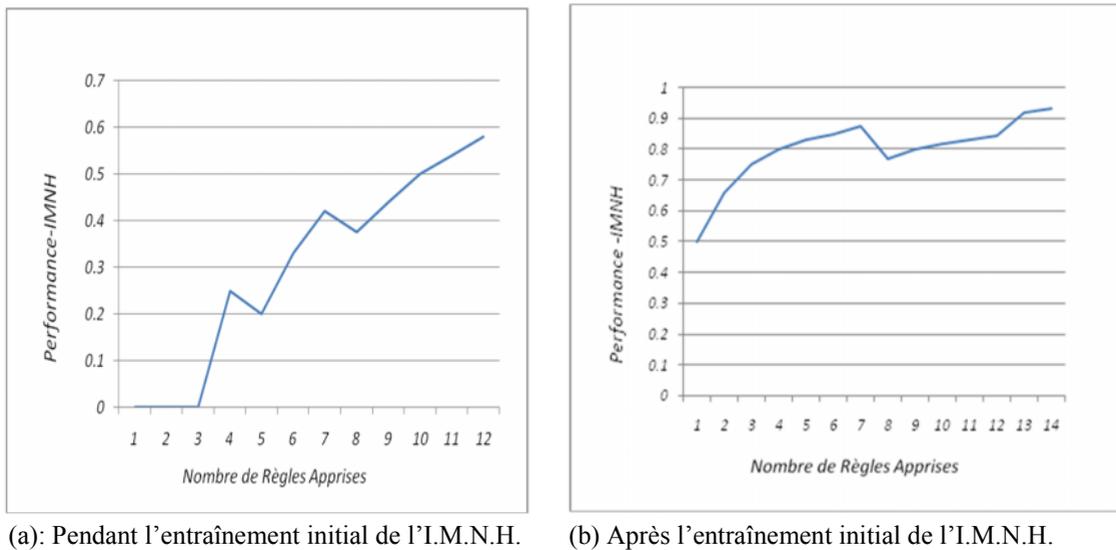


Figure 5.8: Graphes des tests de la performance de l'I.M.N.H.

Comme la performance, de l'I.M.N.H, dépend des connaissances apprises, représentant les profils de l'utilisateur pour la prise en charge des E-mails; alors elle a été évaluée en fonction des règles apprises. La courbe (a) illustre les variations de la performance pendant l'entraînement initial de l'I.M.N.H; alors que la courbe (b) présente les variations de la performance après cet entraînement, donc à la phase d'exploitation de l'I.M.N.H. A partir des graphes (a) et (b), nous constatons que la mesure P est initialement minimale, mais elle s'accroît proportionnellement avec le nombre de règles apprises par l'I.M.N.H, durant son entraînement initial. La courbe (b) illustre une nette amélioration de la mesure de performance, P de l'I.M.N.H; puisque ce dernier a appris des connaissances et il continue à le faire. Ainsi la prise en charge des E-mails, par ce système apprenant, gagne progressivement la satisfaction de l'utilisateur. La satisfaction de ce dernier reste importante et dans certain cas, elle est altérée, ceci correspond à l'inclinaison de la courbe (b) durant sa progression. Cette situation correspond généralement à des changements de certaines pratiques, habitudes et préférences de l'utilisateur. D'où l'apport de l'apprentissage automatique pour la prise en considération de ces changements et l'amélioration de la performance de l'I.M.N.H.

Après avoir présenté l'instanciation des différents concepts et des différents composants, du modèle général et générique, pour l'application de la prise en charge intelligente des E-mails; nous passons à la présentation de la deuxième instanciation et un autre système basé Agent-Apprenant.

5.3. Instanciation du Modèle Général et Générique par le Domaine des Pathologies des Plantes

Le modèle général et générique a été proposé, pour faciliter la construction de systèmes basés Agent-Apprenant par son exploitation et sa réutilisation. Cette deuxième instanciation concerne une application de diagnostic des pathologies des plantes, qui consiste à établir des diagnostics concernant les maladies qui peuvent affecter les plantes, pour pouvoir les traiter convenablement. En réalité, il existe un nombre important de plantes, par conséquent un nombre important de pathologies existe aussi. Les experts en pathologies des plantes prennent les décisions selon les symptômes observés sur la plante affectée. Chaque pathologie possède un traitement spécifique qui doit être utilisé selon l'état de chaque plante avec certaines quantités. Comme exemple de maladies, prenons les deux plantes suivantes affectées respectivement par les pathologies, nommées *Pestalozzia guepini* et *Acariens* [Legros, 2004], illustrées par la figure 5.9 suivante.



Figure 5.9: Deux pathologies de plantes.

Cette deuxième instanciation a engendré un système basé Agent-Apprenant, pour le diagnostic des pathologies des plantes, qui a été implémenté sous la plateforme JADE. L'utilisation d'une plateforme à base d'Agents, pour nous JADE est d'un double intérêt:

1. D'une part, la plateforme supporte la définition de comportements variés d'Agents et offre des classes avec des méthodes prédéfinies permettant de définir et de gérer ces comportements.
2. D'autre part, JADE facilite l'extension de cette instanciation vers un système multi-Agents de diagnostic des pathologies des plantes, à utiliser dans un centre agricole par plusieurs usagers. La plateforme JADE supportera alors les interactions entre les différents Agents du système; ce deuxième objectif est l'une des perspectives pour cette instanciation.

Pour la première instanciation, le recours à une plateforme à base d'Agents n'était pas nécessaire, en effet le système basé Agent-Apprenant de prise en charge intelligente du courrier électronique est un Agent Personnel à utiliser par un usager unique.

5.3.1. La Plateforme JADE

JADE est une plate-forme multi-Agents développée entièrement en JAVA, qui a comme but de faciliter le développement des applications Agents conformant aux normes FIPA97 [Bellifemine et al., 1999]. Cette dernière, établit les règles normatives qui permettent à une société d'Agents d'inter opérer. Les documents FIPA [FIPA, 2006] décrivent le modèle de référence d'une plate-forme multi-Agents où ils identifient les rôles des Agents clés nécessaires pour la gestion de la plateforme. JADE met en œuvre les spécifications de base de la FIPA et possède trois modules principaux nécessaires aux normes de cette dernière.

- DF «Director Facilitator»: le DF permet à la plateforme d'exécuter des applications multi-domaines.
- ACC «Agent Communication Channel » gère la communication entre Agents, assurée par les messages ACL.
- AMS «Agent Management System» supervise l'enregistrement des Agents, leur authentification, leur accès et l'usage de la plateforme.

La plate forme JADE consiste en une collection de conteneurs représentant des environnements d'exécution. Chaque conteneur contient un ou plusieurs Agent(s) et chaque Agent est constitué de plusieurs classes. JADE fournit un grand nombre de classes, dans des packages, avec des méthodes prédéfinies qui implémentent le comportement des Agents qu'elle crée. Un Agent doit être capable de prendre en charge plusieurs tâches, éventuellement concurrentes, en réponse à différents événements. Pour rendre efficace la gestion des tâches, chaque Agent peut être composé de plusieurs comportements. Les tâches sont implémentées par des instances de classes qui étendent la classe `jade.core.behaviours.Behaviour` [Bellifemine et al., 1999]. Un comportement est défini à laide de la classe *Behaviour*; et pour l'implémenter, le développeur doit définir un ou plusieurs objets de cette classe. Tout objet de type *Behaviour* dispose d'une méthode *action()* qui constitue le traitement à effectuer, ainsi qu'une méthode *done()* qui vérifie si le traitement est terminé. JADE offre d'autres méthode pour gérer les objets de la classe *Behaviour*, telles que: *Reset()* permettant de réinitialiser un comportement, *onStart()* et *onEnd()* définissant les opérations à effectuer avant et après l'exécution de la méthode *action()*. Un Agent, au niveau de JADE, peut avoir plusieurs types de comportements:

1. Simples.
2. Planifiés.
3. Composés.

Les comportements simples peuvent être décrits par un comportement à une seule passe dit « OneShotBehaviour », un comportement cyclique ou répétitif dit « CyclicBehaviour » ou

un comportement générique, à définir par le programmeur, dit « GenericBehaviour ». Les comportements planifiés sont mis en œuvre par le comportement périodique dit « TickerBehaviour » et le comportement en réveil dit « WakerBehaviour » permettant d'exécuter respectivement des tâches périodiquement ou après expiration d'un temps d'attente. Les Comportements composés, comme leur nom l'indique, permettent la composition de comportements. La composition permet d'associer des comportements composites aux tâches complexes. La classe CompositeBehaviour incluse dans le package jade.core.behaviours permet de créer des tâches complexes en composant des sous comportements. Les comportements composites peuvent correspondre à un comportement séquentiel « SequentialBehaviour », à un comportement d'un automate à états finis, « FSMBehaviour » ou à un comportement parallèle « ParallelBehaviour ».

Le « SequentialBehaviour » permet d'exécuter en séquence une série de tâches possédant chacune un comportement bien défini. Le « FSMBehaviour » permet de définir des tâches selon un automate d'états finis et le « ParallelBehaviour » offre la possibilité de tâches concurrentes et parallèles. Ainsi, la plate-forme JADE est riche en classes et méthodes prédéfinies supportant différents types de comportements d'Agent.

Pour la création d'un Agent sous JADE, il faut étendre la classe jade.core.Agent, récupérer l'identificateur de l'Agent (l'AID) puis démarrer des comportements (behaviours) à l'aide de la méthode *setup()*. En résumé la plate-forme JADE offre une bibliothèque de classes et de méthodes pour la création d'Agents et la définition de leurs comportements. L'interface suivante illustre l'Agent-Apprenant pour le diagnostic des pathologies des plantes, nommé Agriculteur, implémenté sous la plateforme JADE.

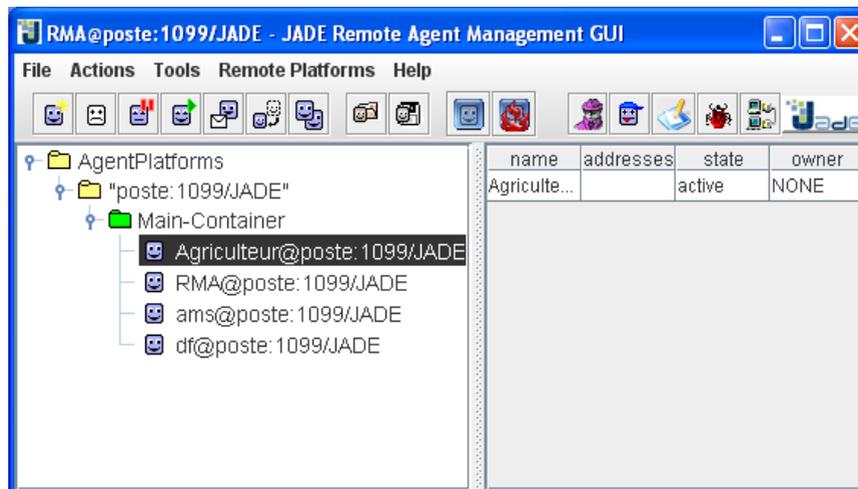


Figure 5.10: Agent-Apprenant de diagnostic des pathologies des plantes sous JADE.

Nous passons maintenant à la structure interne et l'architecture du système basé Agent Apprenant, développé pour le domaine des pathologies des plantes.

5.3.2. Architecture du Système basé Agent-Apprenant pour le Diagnostic des Pathologies des Plantes

En agriculture on dispose d'une masse volumineuse de connaissances préalables concernant les pathologies des plantes; ces connaissances font partie du savoir accumulé des experts en pathologies agricoles et même des simples agriculteurs, intéressés par la santé et le bon état de leurs plantes. L'accès à ces connaissances est facilité par la coopération des experts et des agriculteurs détenant ce savoir et aussi par leur disponibilité sur des supports variés [Legros, 2004; Truffaut, 2010], entre autres dans des bases de données agricoles. Un système basé Agent-Apprenant permet d'acquérir ces connaissances pour mieux les exploiter et les enrichir par, apprentissage automatique, de nouveaux cas et de nouvelles expériences. L'instanciation, du modèle général et générique, par ce domaine, aide et facilite la construction de ce système basé Agent-Apprenant. Ce système, basé sur les connaissances disponibles et l'apprentissage automatique, permet d'établir des diagnostics et de prendre des décisions concernant les pathologies affectant les plantes, en fonction des symptômes présentés. Les raffinements du modèle général et générique, des systèmes basés Agent-Apprenant, et son extension, par les spécificités du domaine des pathologies des plantes, ont engendré l'architecture suivante.

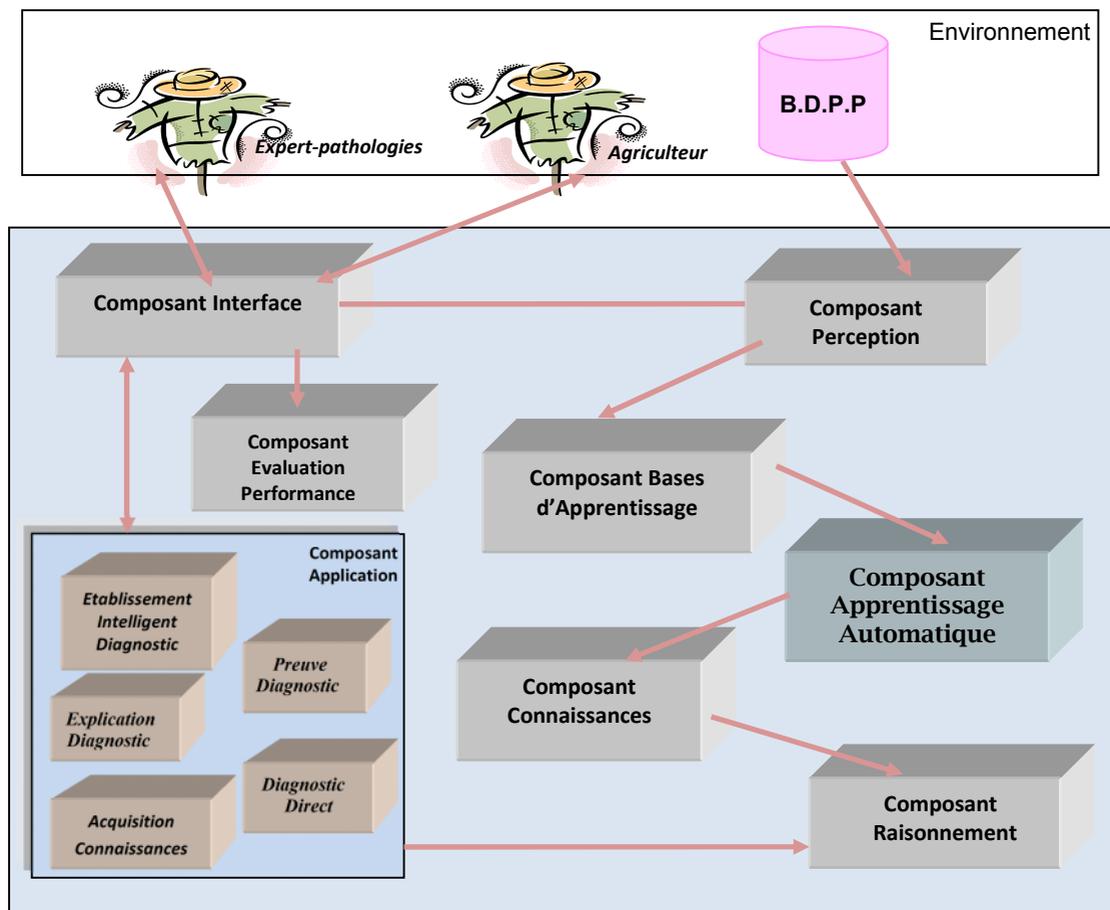


Figure 5.11: Architecture du système pathologies-plantes.

5.3.2.1. Instanciation de l'Environnement

L'environnement du système basé Agent-Apprenant pour le diagnostic des pathologies des plantes, tel qu'il est illustré par la figure 5.11, est constitué des trois éléments suivants:

1. *Agriculteur.*
2. *Expert-pathologies.*
3. *Bases de données pathologies.*

L'agriculteur est un usager du système qui utilise, ce dernier, comme un guide intelligent pour découvrir les diagnostics de ses plantes affectées. Ce système, basé sur un Agent Intelligent d'aide au diagnostic, permet à l'agriculteur de bien suivre l'état de santé de ses plantes afin d'aboutir à une meilleure production. L'expert-pathologies ou le pathologiste est un connaisseur qui détient un savoir sur les pathologies des plantes et qui peut enrichir le système avec de nouvelles connaissances. Les bases de données pathologies (B.D.P.P) sont des sources électroniques des cas de plantes affectées, que le système basé Agent-Apprenant doit toujours scruter afin d'enrichir ses bases d'exemples.

5.3.2.2. Composant Interface

Ce composant fait interface et communique avec:

- Les deux usagers du système: l'agriculteur et l'expert-pathologies.
- Le composant Perception.
- Le composant Application.
- Le composant Evaluation Performance.

5.3.2.3. Composant Perception

Ce composant est d'une grande importance pour un Agent-Apprenant, pour ce système, la perception permet essentiellement:

- D'extraire des exemples d'entraînement des Bases de Données des Pathologies des Plantes (B.D.P.P).
- D'acquérir les connaissances préalables, le domaine étant riche en connaissances et dispose d'une base théorique des pathologies des plantes.
- D'observer les deux usagers et capter leurs manipulations et leurs expériences, à travers le composant Interface pour enrichir les exemples d'entraînement.

5.3.2.4. Composant Bases d'Apprentissage

Le domaine de l'agriculture comporte une théorie et un savoir préalables concernant les pathologies des plantes. Ce savoir structuré et bien établi correspond à l'instanciation du concept «*connaissances préalables*» du modèle général et générique. Ainsi un système apprenant, pour ce domaine, doit être basé sur ces connaissances pour mener son processus d'apprentissage automatique. Pour cette deuxième instanciation, la théorie du domaine est d'une grande importance et elle a un effet sur le reste des éléments de l'apprentissage

automatique de cette instanciation. En effet, le domaine agricole dispose d'un large volume de savoir et de connaissances, accumulés au fil des années et détenus par des experts en pathologies et des agriculteurs. Ce savoir ainsi que plusieurs exemples, de cas de plantes affectées, sont disponibles sur des supports variés [Legros, 2004; Truffaut, 2010], entre autres des bases de données agricoles. Nous présentons ici, des extraits de ses connaissances.

Exemple 1: *Si Feuilles avec:*

- sécrétion d'une substance sucrée;
- un champignon pulvérulent noir;
- dessèchement fort;

Et Rameaux avec formation de méplats;

Alors Cochenilles.

Exemple 2 : *Si Feuilles avec:*

- tâches irrégulières, de forme arrondie et de couleur marron clair;
- pustules noires réparties de façon concentrique;

Et Rameaux avec dessèchement des jeunes pousses;

Alors Die Black.

Afin de tirer profit des connaissances préalables disponibles et des nouvelles expériences, le Composant Bases d'Apprentissage (C.B.A), du système basé Agent-Apprenant pour le diagnostic des pathologies des plantes, est constitué des éléments illustrés par la figure 5.12 suivante.

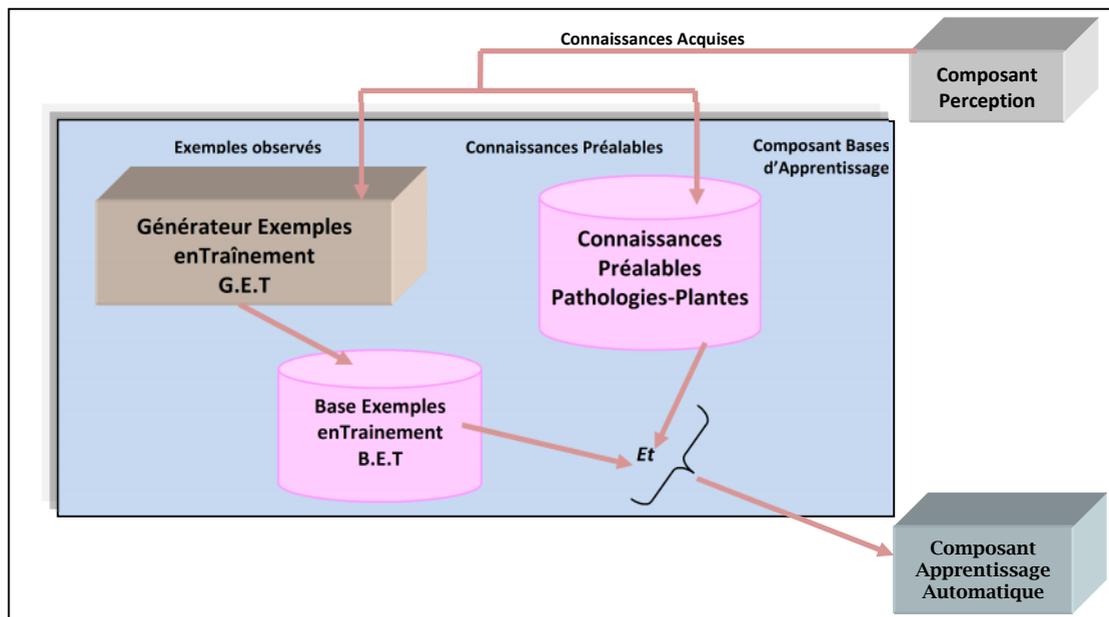


Figure 5.12: C.B.A du système des pathologies-plantes.

Le sous composant Générateur d'Exemples d'enTrainement (G.E.T) est, pour cette instanciation, un simple gestionnaire des exemples d'entraînement. Ces derniers sont

extraits, par le composant de perception, à partir des Bases de Données des Pathologies des Plantes (B.D.P.P) ainsi qu'à partir des manipulations, du système de diagnostic, effectuées par les deux usagers: l'agriculteur et l'expert-pathologies. Ces manipulations seront explicitées à la présentation du composant application. Le G.E.T permet d'épurer les exemples de la redondance et des informations superflues pour générer une Base Exemples d'entraînement (B.E.T). Ainsi, contrairement au G.E.T de la première instanciation, celui du système des pathologies-plantes, n'effectue pas une construction automatique des exemples d'entraînement. Le Composant Bases d'Apprentissage (C.B.A) comporte aussi une Base renfermant la théorie du domaine concernant les pathologies des plantes. L'opérateur (*Et*) indique que la sortie du C.B.A, qui est l'entrée du composant Apprentissage Automatique, est double. Cette entrée est constituée de la B.E.T et de la base des connaissances préalables. Cette entrée double nécessite un apprentissage adéquat.

5.3.1.5. Composant Application

Le composant Application fournit un ensemble de services dont les plus importants, illustrés par l'architecture du système basé Agent-Apprenant pour le diagnostic des pathologies des plantes, sont:

- *Etablissement direct de diagnostic.*
- *Etablissement intelligent de diagnostic.*
- *Preuve d'un diagnostic donné.*
- *Explication d'un diagnostic déjà établi.*
- *Acquisition explicite de connaissances.*

L'instanciation de la « *classes des tâches* » pour le système basé Agent-Apprenant, de diagnostic des pathologies des plantes, est le sous ensemble des services précédents dont la réalisation est continuellement améliorée par apprentissage automatique. Ainsi la « *classe des tâches* » pour ce système est:

- *Etablissement intelligent de diagnostic.*
- *Preuve d'un diagnostic donné.*
- *Explication d'un diagnostic déjà établi.*

Les deux autres services sont effectués par les usagers du système. L'établissement direct d'un diagnostic peut être entrepris par l'agriculteur et l'expert-pathologies. Cependant, l'acquisition explicite de connaissances ne peut être effectuée que par l'expert-pathologies, qui possède un savoir concernant les pathologies des plantes et peut disposer de nouvelles connaissances, établies par la communauté agricole et scientifique.

L'établissement direct de diagnostic représente la manipulation, de la part des deux usagers du système de diagnostic, permettant à ce dernier d'acquérir des exemples d'entraînement. Ce service permet aux usagers d'établir personnellement des diagnostics, s'ils le désirent, pour des cas connus et simples. L'intérêt, pour le système basé Agent-Apprenant, de cette manipulation est l'acquisition des exemples d'entraînement observés. Les écrans suivants

illustrent deux interfaces, du système basé Agent-Apprenant, pour le diagnostic des pathologies des plantes direct et intelligent.

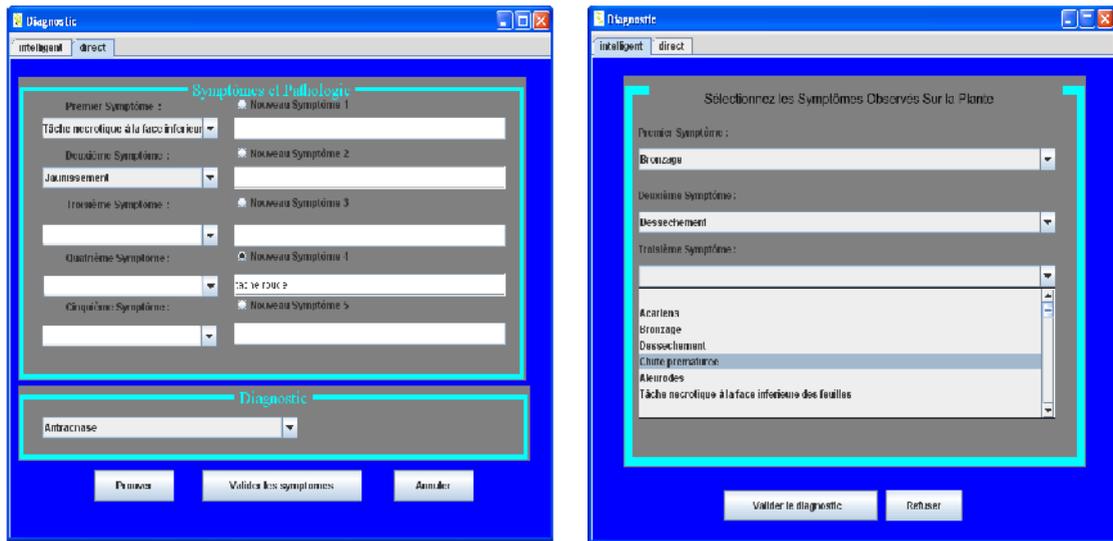


Figure 5.13: Ecrans du système des pathologies-plantes.

Les services : établissement intelligent de diagnostic, preuve d'un diagnostic donné et explication d'un diagnostic déjà établi sont pris en charge par le composant Raisonnement, ainsi ils seront décrits avec ce composant. La capacité de raisonnement, de l'Agent intelligent de diagnostic des pathologies des plantes, est basée sur le composant des connaissances, qui constitue la source principale des prises de décisions. Nous décrivons donc ce composant avant celui du raisonnement.

5.3.1.6. Composant Connaissances

Les connaissances du domaine des pathologies des plantes sont naturellement exprimées en prémisses et conclusions, c'est à dire que: selon les symptômes observés sur une plante, le diagnostic peut être déterminé. Ainsi ces connaissances sont des règles de la forme Si symptômes Alors diagnostic, exemple:

Si Feuilles avec

- sécrétion d'une substance sucrée ;
- un champignon pulvérulent noir;
- dessèchement fort;

Et Rameaux avec formation de méplats;

Alors Cochenilles.

Par conséquent le concept « *représentation des connaissances* », du modèle général et générique, a été instancié par le formalisme des règles de production qui permettent d'exprimer la syntaxe et la sémantique des connaissances du domaine des pathologies des plantes, naturellement exprimées en règles. L'attachement d'une variable exprimant, un facteur donné, à une règle de production est possible. Ce facteur peut correspondre à un taux d'incertitude de la règle, ceci permet de représenter des connaissances incertaines, ou

à un degré de confiance de la règle. A chaque règle ou connaissance, sur les pathologies des plantes, nous attachons un scalaire représentant le degré de confiance (*DC*) de la règle. Ce dernier correspond au degré de satisfaction des usagers dans les prises de décision, concernant les pathologies des plantes, entreprises par le système basé Agent-Apprenant. Par conséquent, la connaissance de l'exemple précédent est formulée ainsi:

Si Feuilles = (sécrétion substance sucrée et champignon pulvérulent noir et dessèchement fort) et Rameaux = formation de méplats Alors Cochenilles avec DC.

Le Composant Connaissances C.C, du système de diagnostic des pathologies des plantes, illustré par la figure 5.14 suivante, est formé de la Base de Connaissances-Pathologies-Plantes (B.C.P.P) et d'un sous-composant de maintien de la cohérence de cette base.

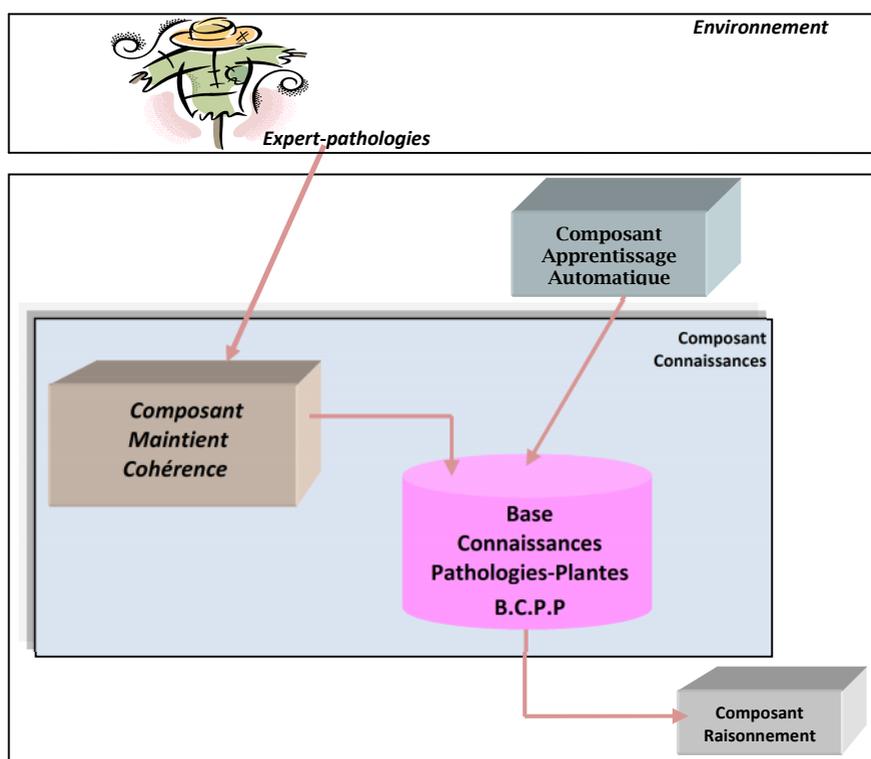


Figure 5.14: Composant connaissances du système pathologies-plantes

La Base de Connaissances-Pathologies-Plantes (B.C.P.P) est l'élément fondamental, sur lequel se base le composant Raisonnement, d'où l'importance de la maintenance de la cohérence des connaissances de cette base. Le sous-composant Maintient Cohérence est utilisé, en exclusivité, par l'expert-pathologies et lui permet d'accéder à la B.C.P.P et de réviser ses connaissances. Tel qu'il a été défini dans le modèle général et générique, présenté au chapitre 4, le sous-composant Maintient Cohérence est de préférence semi-automatique. Ainsi un cognicien, ici l'expert-pathologies, est donné un contrôle, sur la B.C.P.P, pour veiller à l'intégrité de ses règles plutôt que de confier, complètement au système, cette tâche critique qui peut avoir des effets sur la fiabilité des décisions prises par le composant Raisonnement.

5.3.1.7. Composant Raisonement

Le composant Raisonement, au niveau du système basé Agent-Apprenant de diagnostic des pathologies des plantes assure:

- *L'Établissement intelligent de diagnostic.*
- *La Preuve d'un diagnostic donné.*
- *L'Explication d'un diagnostic déjà établi.*

Ce composant utilise un moteur d'inférence qui fonctionne en chaînage mixte, donc cette instanciation est basée sur des mécanismes du chaînage avant et du chaînage arrière. Le chaînage avant est utilisé pour l'établissement intelligent de diagnostic alors que le recours au chaînage arrière sert à prouver un diagnostic donné. Pour l'établissement intelligent de diagnostic, le composant Raisonement, illustré par la figure 5.15, commence par construire une Base de Faits (B.F), à partir des symptômes observés sur une plante, et invoque son moteur d'inférence en chaînage avant. Les inférences sont établies à l'aide de la Base de Connaissances-Pathologies-Plantes (B.C.P.P) et la B.F. Dans le cas où l'un des deux usagers, du système, établit un diagnostic, puis demande une preuve de ce dernier; le composant de raisonnement invoque son moteur d'inférence en chaînage arrière.

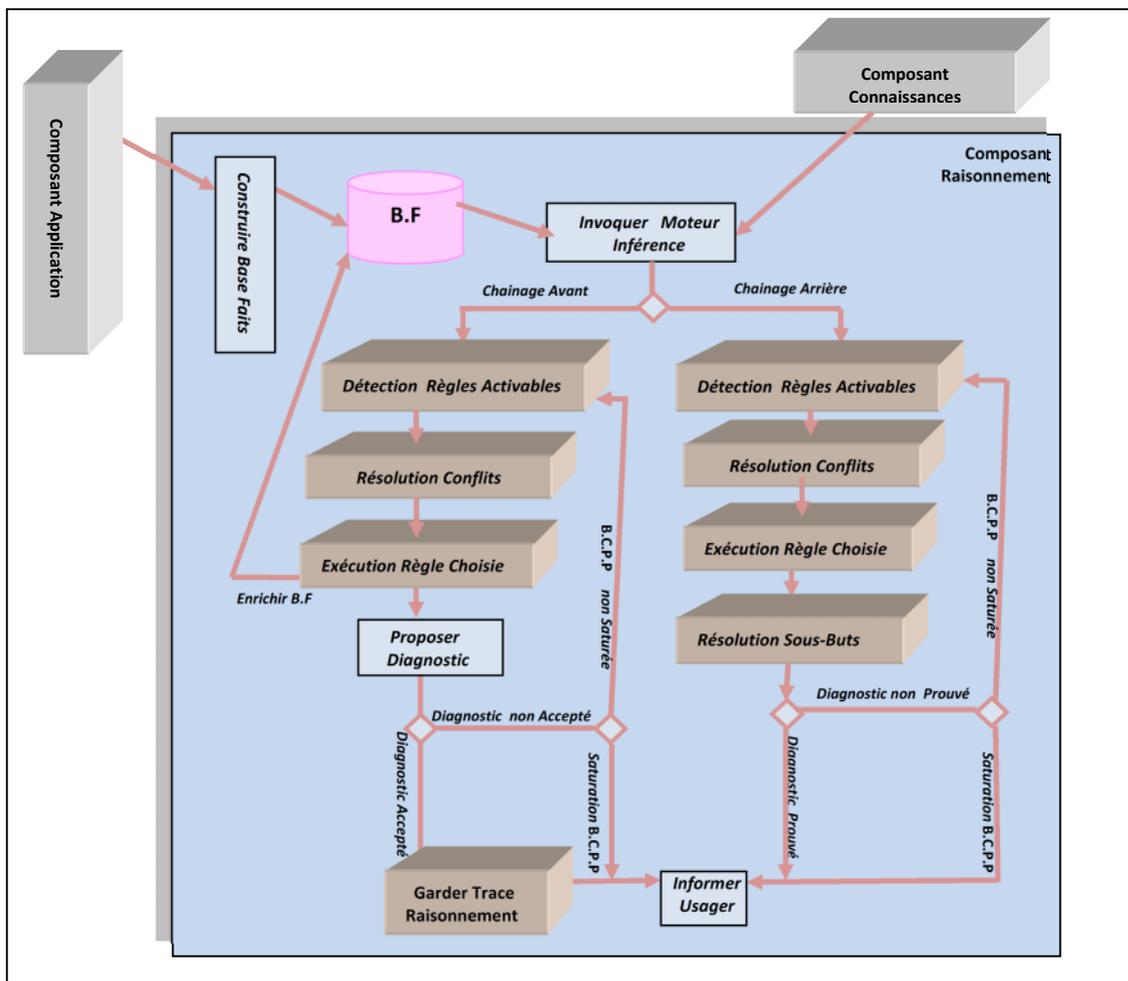


Figure 5.15: Composant raisonnement du système pathologies-plantes.

Le fonctionnement du moteur d'inférence, selon les deux chaînages, tel qu'il est montré par la figure 5.15, suit un cycle répétitif qui comporte trois phases:

Première phase: détection de l'ensemble des règles activables.

Le moteur d'inférence détecte dans la Base de Connaissances-Pathologies-Plantes (B.C.P.P) toutes les règles dans le ou les faits situé(s) dans la partie:

1. Prémisse correspond (ent) aux symptômes de la B.F, ainsi, le moteur d'inférence fonctionne en chaînage avant pour établir un diagnostic intelligent.
2. Conclusion correspondant au diagnostic de la B.F, ainsi, le moteur d'inférence fonctionne en chaînage arrière pour prouver un diagnostic établi par l'agriculteur ou l'expert-pathologies.

Deuxième phase: choix de la règle à activer par résolution de conflit.

Le moteur d'inférence choisit la règle qui va être activée, selon une stratégie de résolution de conflit. La stratégie utilisée est le choix de la règle possédant le plus haut degré de confiance parmi l'ensemble des règles activables.

Troisième phase: exécution de la règle choisie.

Dans le chaînage avant, lorsqu'une règle est déclenchée, les faits conclusions, de la règle, sont ajoutés à la B.F; ainsi l'arbre des solutions des diagnostics possibles est exploré dans sa profondeur. Afin d'assumer l'explication d'un diagnostic inféré, sur demande de l'utilisateur, le composant Raisonnement garde trace des inférences menées. Pour la preuve d'un diagnostic donné avec un chaînage arrière, la B.F contient tout à la fois des faits établis dits symptômes et des faits à établir dits diagnostics. Ainsi, le déclencheur de règles de la B.C.P.P se réfère uniquement aux diagnostics. Cela signifie donc qu'une règle n'est déclenchée que si les faits de sa partie prémisse sont des symptômes à établir dans la B.F, ceci correspond à la résolution des sous buts. Après retour arrière et saturation de la B.C.P.P, c'est l'échec et le diagnostic n'est pas prouvé.

5.3.1.8. Composant Apprentissage Automatique

Le système basé Agent-Apprenant, de diagnostic des pathologies des plantes, fait recours à l'apprentissage pour une construction automatique de connaissances servant à la prise en charge des tâches à supporter par ce système. Tel que, nous l'avons expliqué, au niveau du composant Bases d'Apprentissage, les connaissances concernant les pathologies des plantes sont naturellement exprimées sous forme de règles, et le concept «représentation des connaissances» a été instancié par le formalisme des règles de production. Pour le composant Apprentissage Automatique, nous commençons par présenter l'instanciation du concept «*approche/méthodes d'apprentissage*».

Le domaine d'agriculture comporte une théorie et un savoir préalables concernant les pathologies des plantes. Ce savoir correspond à l'instanciation du concept «*connaissances préalables*» du modèle général et générique. Ainsi un système apprenant, pour ce domaine,

doit être basé sur ces connaissances pour mener son processus d'apprentissage automatique. Ce domaine, étant riche en connaissances et dispose d'une base théorique sur les pathologies des plantes, dite aussi *connaissances préalables*, qui est d'une grande importance pour le choix de l'approche d'apprentissage au niveau de cette instanciation. En effet, selon la mise en correspondance, présentée au chapitre 3, le choix dépend de la théorie du domaine si elle est parfaite ou non. Pour le domaine agricole, cette théorie n'est pas complète, en d'autres termes les connaissances préalables ne couvrent pas tous les cas d'affections possibles des plantes. De nouvelles pathologies, non recensées auparavant, peuvent toujours apparaître. Les causes sont multiples, entre autres: le changement climatique, les élévations des températures, la nature des engrais et des produits chimiques utilisés en agriculture ainsi que certains nouveaux croisements génétiques des plantes.

Ainsi les connaissances disponibles, pour ce domaine, sont approximatives et par conséquent, selon la mise en correspondance, établie au chapitre 3 entre les systèmes basés Agent-Apprenant et les approches d'apprentissage automatique, l'approche d'apprentissage hybride est la plus appropriée pour ce domaine. Pour cette instanciation, le composant Apprentissage Automatique se base sur les connaissances préalables et les exemples d'entraînement, correspondant aux manipulations des usagers, et ceux extraits à partir des bases de données des cas d'affections des plantes. Par conséquent, pour instancier le concept «*méthodes d'apprentissage*», nous avons besoin de techniques qui combinent entre les connaissances préalables et les exemples d'entraînement, qui pour cette instanciation sont nombreux. Partant de ce besoin et du formalisme de représentation des connaissances utilisé pour le domaine des pathologies des plantes, nous avons opté pour la méthode dite Programmation Logique Inductive (PLI) et un de ses algorithmes. Avec la PLI, il est possible de se baser sur les connaissances préalables, dites aussi le Background, pour généraliser à partir des exemples d'entraînement. L'algorithme choisi, dit FOIL [Quinlan, 1990; Quinlan et Cameron-Jones, 1993] pour First Order Inductive Logic, se base sur les principes de la PLI pour l'apprentissage de règles.

La programmation logique inductive, tel qu'elle a été présentée au chapitre 2, définit un cadre formel général pour l'apprentissage de règles en logique du premier ordre. La PLI permet d'apprendre des règles qui sont correctes et complètes par rapport aux exemples et aux connaissances préalables. Une règle est correcte si elle ne couvre pas de fait sur le concept cible qui soit faux dans l'interprétation, définie par les exemples d'entraînement et les connaissances préalables. Elle est complète si elle couvre tous les exemples qui sont vrais dans l'interprétation [Lavrac et Dzeroski, 1994; Muggleton, 1991]. Les règles apprises par la PLI sont des clauses de Horn avec possibilité d'avoir des variables dans les prés et post conditions; ainsi la PLI, et par conséquent FOIL aussi, utilise un formalisme de représentation de connaissances expressif. La figure 5.16 illustre le raffinement du composant Apprentissage Automatique, du système des pathologies des plantes.

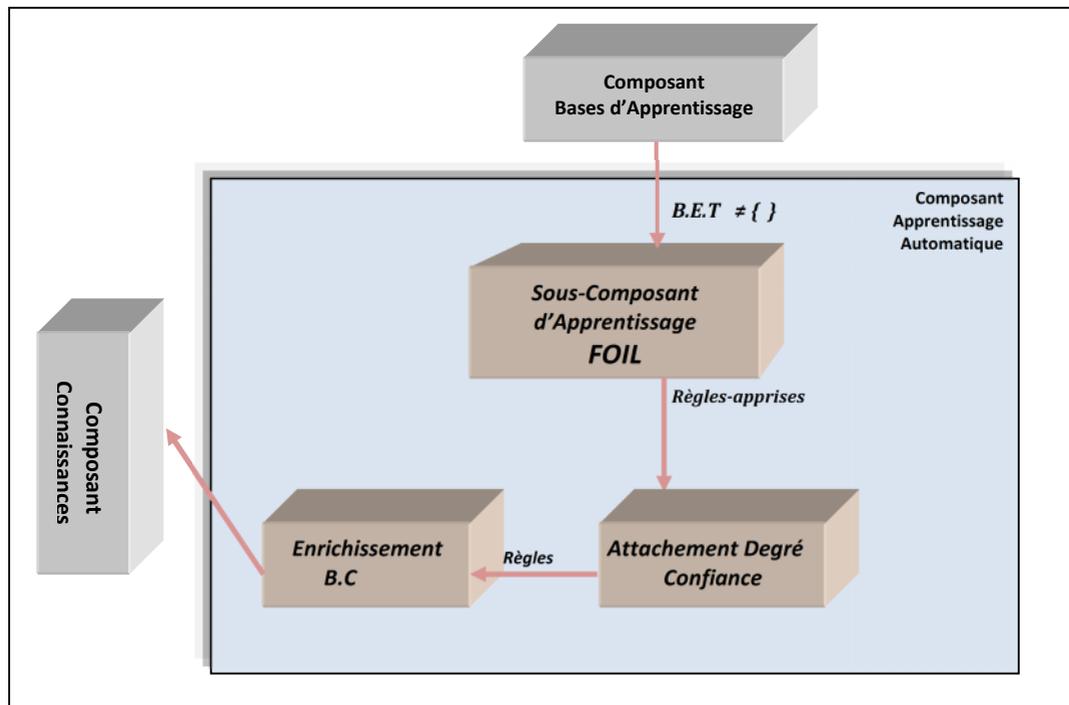


Figure 5.16: Composant d'apprentissage du système pathologies-plantes.

En partant de la sortie du composant Bases d'Apprentissage, le composant Apprentissage Automatique, du système des pathologies-plantes, invoque son sous composant d'apprentissage FOIL. Ce dernier se charge d'apprendre un ensemble de règles, selon un principe de fonctionnement qui sera explicité dans la section suivante. Les règles générées, par FOIL, sont augmentées par un degré de confiance variable, puis elles enrichissent la base de connaissances des pathologies des plantes (B.C.P.P).

Après avoir instancié tous les concepts liés à l'apprentissage automatique pour cette application, nous explicitons le principe de fonctionnement du sous composant FOIL.

5.3.1.8.1. Principe de Fonctionnement du Sous Composant FOIL

Le sous composant FOIL permet d'apprendre un ensemble de règles, qui couvrent la plupart des exemples positifs et rejettent la plupart des exemples négatifs en se basant sur les connaissances préalables du domaine des pathologies des plantes. Les exemples positifs et négatifs sont issus de la Base des Exemples d'entraînement, la *B.E.T* du composant Bases d'Apprentissage. Le principe de fonctionnement de ce sous composant est illustré par le pseudo code suivant.

FOIL (B.E.T) : Règles-apprises

Début

Pos \leftarrow L'ensemble des exemples pour lesquels le diagnostic cible est Vrai

Neg \leftarrow L'ensemble des exemples pour lesquels le diagnostic cible est Faux

Règles-apprises \leftarrow {} (* ensemble de règles-apprises initialisé à vide*)

Tant que Pos \neq {} faire

(* Apprendre une règle*)

Nouv-règle \leftarrow une règle contenant le diagnostic cible en tête et de corps vide

(* c'est la règle la plus générale*)

Neg-Nouv-règle \leftarrow Neg

Tant que Neg-Nouv-règle \neq {} faire

(* Spécialiser Nouv-règle par ajout d'un littéral *)

Cand-Litt \leftarrow l'ensemble des littéraux candidats pour la règle Nouv-règle

Sélectionner Litt \in Cand-Litt tel que Litt maximise une mesure de gain d'informations

Ajouter Litt au corps de Nouv-règle

Neg-Nouv-règle \leftarrow Neg-Nouv-règle - { exemples rejetés par Nouv-règle }

Fin tant que

Règles-apprises \leftarrow Règles-apprises + Nouv-règle

Pos \leftarrow Pos - { exemples couverts par Nouv-règle }

Fin tant que

Return Règles-apprises

Fin

La Nouv-règle apprise est initialisée avec une règle contenant le diagnostic cible en tête et de corps vide, ainsi une règle dans FOIL, est générée en partant du général au spécifique par ajout de littéraux à la partie prémisse de la règle. La sélection du meilleur littéral, parmi l'ensemble des littéraux candidats, est effectuée au moyen d'une mesure de gain. Cette mesure évalue les proportions d'exemples positifs et négatifs couverts par la règle, avant et après l'ajout d'un littéral. La spécialisation, par FOIL, de la Nouv-règle, se fait par son raffinement qui permet d'ajouter des prémisses à la règle maximisant la mesure de gain d'informations pondéré [Quinlan, 1990; Quinlan et Cameron-Jones, 1993], calculé par la formule suivante:

$$FOIL-Gain(R, L) = t (\log_2 (p_1/p_1 + n_1) - \log_2 (p_0/p_0 + n_0)) \quad \text{Eq. (3)}$$

Le gain d'informations pondéré est basé sur la différence des contenus d'informations des règles (rapport du nombre des exemples positifs sur le nombre de tous les exemples couverts), et sur une pondération de cette différence par le nombre d'exemples positifs. Cette formule estime l'utilité de l'ajout d'un littéral en se basant sur le nombre de substitutions positives et négatives avant et après l'ajout du littéral. Soit R une règle et L un littéral candidat qui va être ajouté au corps de R; soit R1 la règle obtenue par ajout du littéral L à la règle R. Alors la valeur du gain d'informations FOIL-Gain(R, L) d'ajout de L à R est donnée par Eq. (3), Avec:

- p_0 nombre de substitutions positives de R.
- n_0 nombre de substitutions négatives de R.
- p_1 nombre de substitutions positives de R1.
- n_1 nombre de substitutions négatives de R1.
- t nombre de substitutions positives de R qui sont aussi des substitutions positives de R1.

L'utilisation de la mesure du gain d'informations, pour la sélection du meilleur littéral au cours de la spécialisation d'une règle, permet à FOIL d'apprendre des règles qui classifient bien les exemples d'entraînement. Cette mesure permet de maximiser la pureté des exemples couverts par les règles apprises. L'algorithme FOIL opère sa recherche de règles avec une préférence pour les règles possédant plusieurs substitutions positives et peu de substitutions négatives. Cette préférence permet d'accélérer le processus d'apprentissage de règles de l'algorithme FOIL et par conséquent, elle fait augmenter son efficacité.

Il faut noter que l'exploitation des connaissances préalables, sur les pathologies des plantes permet aussi de réduire la complexité de l'apprentissage. En effet les règles générées par le composant Apprentissage doivent être consistantes avec les connaissances préalables et les exemples d'entraînement. Ainsi la taille de l'espace de règles est réduite à celles consistantes avec l'interprétation. Cette dernière est renforcée par les connaissances préalables, qui rejettent toute règle non satisfaite et non consistante avec ses connaissances. L'algorithme FOIL avec son formalisme de représentation des connaissances expressif et facilement interprétable permet à cette instanciation de construire automatiquement des connaissances et les mettre à la disposition du système pour qu'il puisse prendre en charge les tâches de ce domaine. La facilité d'interprétation des règles apprises permet aussi aux usagers de vérifier leur fiabilité et de comprendre les traces des raisonnements lors des explications des diagnostics établis.

5.3.1.9. Composant Evaluation Performance

Le but final de l'apprentissage automatique étant l'amélioration de la performance du système apprenant, donc l'amélioration de la prise en charge des tâches à supporter. Pour le système basé Agent-Apprenant, de diagnostic des pathologies des plantes, le concept « *une mesure de la performance* » a été instancié par la mesure de satisfaction des usagers, ici l'agriculteur et l'expert-pathologies, concernant les prises de décisions du système. Cette mesure, P , est donc identique à celle de la première instanciation, définie par le quotient du nombre de diagnostics acceptés sur le nombre d'inférences totales.

$$P = \text{Nbr Diag-Acc} / \text{Nbr Inf-Tot}$$

Avec :

Nbr Diag-Acc = Nombre de Diagnostics Acceptés.

Nbr Inf-Tot = Nombre d'inférences Totales.

Une série d'évaluation de la performance, P , du système basé Agent-Apprenant, de diagnostic des pathologies des plantes est illustrée par les graphes suivants.

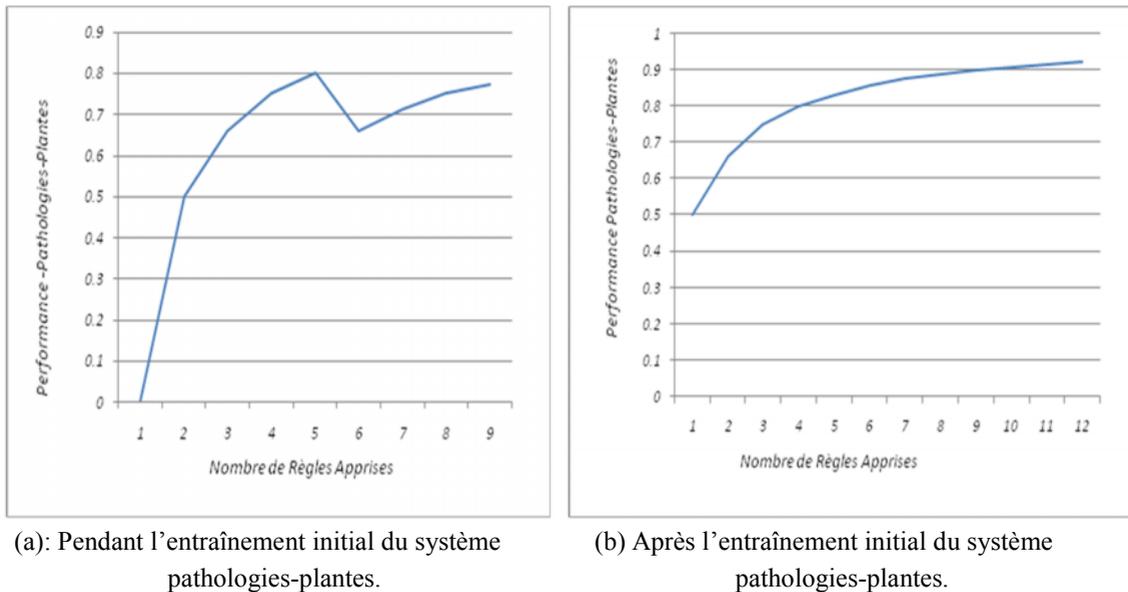


Figure 5.17: Graphes des tests de la performance du système pathologies-plantes.

La première courbe illustre les variations de la performance P , durant les phases initiales d'entraînement du système; alors que la seconde courbe présente les variations de la performance P , après l'entraînement initial, donc à la phase d'exploitation du système basé Agent-Apprenant. Pour cette deuxième instanciation, la courbe (a) comporte peu d'inclinaisons, et les prises de décisions ont rapidement atteint un haut degré de satisfaction des usagers à la phase d'entraînement initial. Aussi la courbe (b), pour le système des pathologies-plantes, est en évolution continue et elle est plus stable que celle correspondant à la première instanciation. Cette amélioration de la performance pour la prise en charge du diagnostic des pathologies des plantes, est renforcée par la disponibilité des connaissances préalables. En effet, ces dernières, comme nous l'avons précisé ci-dessus, permettent d'accélérer le processus d'apprentissage et aident le système à apprendre des règles consistantes avec l'interprétation, définie par les connaissances préalables et les exemples d'entraînement, donc des règles fiables. Ainsi les prises de décisions, basées sur ces règles, sont généralement satisfaites; par conséquent la performance, concernant la prise en charge des tâches de ce domaine, est en amélioration continue.

5.4. Conclusion

Ce chapitre a présenté deux instanciations du modèle général et générique, établi pour les systèmes basés Agent-Apprenant. L'instanciation consiste à réutiliser le modèle proposé, avec sa base conceptuelle et ses composants génériques, et l'étendre par les spécificités de l'application. La première instanciation concerne le courrier électronique, qui est devenu un moyen de communication populaire et une carte d'identité de n'importe quel internaute. Cependant, l'accès, aux E-mails, et leur manipulation deviennent des

tâches répétitives et consommatrices de temps. D'où l'idée d'instancier le modèle générique par ce domaine pour aboutir à un système basé Agent-Apprenant pour la notification et la prise en charge intelligente des E-mails. L'architecture de ce système, nommé l'I.M.N.H, a été engendrée à partir de la spécialisation du modèle général par le domaine des E-mails. Les composants noyaux du modèle font partie de l'architecture, en plus des extensions et des raffinements relatifs au domaine de l'application. L'instanciation de la base conceptuelle, du modèle, pour le domaine des E-mails, a été faite progressivement et parallèlement avec celles des composants génériques. L'I.M.N.H est un système basé Agent-Apprenant Personnel qui a pour objectif l'adaptation à son usager. La personnalisation et par conséquent l'amélioration de la performance, de ce système, sont assurées par un apprentissage automatique des pratiques, habitudes et préférences de son usager concernant la prise en charge des E-mails.

La deuxième instanciation est relative à une application de diagnostic des pathologies des plantes; qui a été menée, tout comme la première instanciation, par une suite d'extensions et de raffinements des composants génériques, du modèle général, et des concepts qui leur sont associés. Le système basé Agent-Apprenant de diagnostic des pathologies des plantes a été implémenté sous la plateforme JADE. Cette dernière a été utilisée pour la possibilité de définition de comportements variés au sein d'un Agent et surtout pour faciliter l'extension future de cette instanciation vers un système multi-agents de diagnostic des pathologies des plantes. Les connaissances préalables disponibles, pour le domaine d'agriculture, avec les exemples d'entraînement aident à l'apprentissage de connaissances générales fiables. Ainsi, les prises de décisions, du système basé Agent-Apprenant de diagnostic des pathologies des plantes, sont en amélioration continue.

Ces deux instanciations nous ont permis d'exploiter et de réutiliser le modèle général et générique proposé, pour faciliter la construction de systèmes basés Agent-Apprenant. Aussi elles nous ont servi à mettre en œuvre les différents éléments de la base conceptuelle du modèle. La mise en correspondance entre les systèmes basés Agent-Apprenant et les approches/méthodes d'apprentissage automatique, présentée au chapitre 3, a guidé les choix, établis pour les deux systèmes basés Agent-Apprenant présentés dans ce chapitre.

Conclusion Générale

1. Introduction

Nous nous sommes intéressés dans ce travail de thèse à la technologie Agent et l'apprentissage automatique, des disciplines fortement liées à l'intelligence artificielle. Cette dernière tente, depuis longtemps, non seulement de comprendre le fonctionnement des entités intelligentes, mais aussi d'en construire.

D'une part, par son potentiel et son répertoire puissants la technologie Agent, émergeant principalement de l'intelligence artificielle distribuée, met en œuvre un ensemble de concepts et de techniques permettant d'aboutir à des systèmes intelligents. D'autre part, l'apprentissage automatique par ses capacités de construction automatique de connaissances et d'apprentissage de stratégies de contrôle, permet la résolution de problèmes complexes. Avec les avancées technologiques, ces problèmes posent de plus en plus des goulots d'étranglement au développement de systèmes informatiques.

La technologie Agent et l'apprentissage automatique, un domaine de recherche prometteur, permettent la résolution de problèmes nécessitant des capacités d'intelligence, d'autonomie, de flexibilité, d'apprentissage et d'adaptation. Nos contributions, dans ce domaine attrayant, concernent un travail de base, sur l'apprentissage automatique dans un Agent. Cette thèse, est le fruit de longues investigations et réflexions qui ont abouti à un travail réalisé pouvant être réutilisé, exploité et ouvrant des voies et des perspectives de recherche.

2. Contributions

Le travail réalisé dans cette thèse est d'un apport capital pour le thème de l'apprentissage automatique dans un Agent; en effet il apporte un fondement et un support de base, comme il constitue une assise conceptuelle et un modèle général et générique avec des expérimentations réelles à ce thème de recherche. Les contributions, de notre travail, permettent de réaliser et de concrétiser les objectifs visés par cette thèse; elles se résument principalement aux points suivants:

1. Mise en Correspondance entre les types d'Agents-Apprenants et les Approches/ Méthodes d'apprentissage automatique.

La mise en correspondance établie a permis de faire un appariement entre les types d'Agents-Apprenants, déterminés au préalable, et les approches/méthodes d'apprentissage automatique les plus appropriées. Donc celles renforçant les capacités d'intelligence, d'autonomie, de flexibilité, d'adaptation et de personnalisation d'un Agent. Cette mise en correspondance a nécessité un travail préliminaire concernant:

- La Catégorisation des méthodes d'apprentissage.
- La Détermination des différents types d'Agents-Apprenants.

La Catégorisation des méthodes d'apprentissage.

Confrontés à une grande panoplie et une pléthore de méthodes d'apprentissage, nous avons analysé ces méthodes et les principes de leur classification pour les catégoriser. Ceci nous a permis de dégager des approches d'apprentissage puis d'étudier les méthodes les plus connues et les plus représentatives pour chacune des approches dégagées.

La Détermination des différents types d'Agents-Apprenants.

Basés sur un ensemble de critères de classification d'Agents, que nous avons décrits et explicités au préalable, nous avons dégagé les différents types d'Agents Apprenants. Ensuite nous avons défini et caractérisé les systèmes basés Agent-Apprenant.

Ainsi, l'appariement établi a mis les systèmes réactifs adaptatifs en correspondance avec l'apprentissage par renforcement. Ce dernier est le plus apte à apprendre des stratégies de contrôle, pour une adaptation du comportement. Pour pallier des difficultés au niveau de ce type d'apprentissage, une bonne conception de la fonction de renforcement ainsi qu'une balance entre l'exploration et l'exploitation, de l'espace de situation-action, sont nécessaires. Pour une construction automatique de connaissances, le facteur clé pour mettre en correspondance des systèmes basés Agent Apprenant avec l'une des trois approches d'apprentissage: déductive, inductive ou hybride, est la disponibilité d'une théorie de domaine, d'exemples

d'entraînement ou les deux ensemble. Des critères, liés à la nature des connaissances à apprendre et les formalismes de représentation de ces connaissances, peuvent guider le choix des méthodes d'apprentissage, surtout inductives.

2. Etablissement d'un modèle général et générique

Afin de disposer d'un modèle pour l'apprentissage automatique dans un Agent, cette thèse a contribué par l'établissement d'un modèle général et générique pour les systèmes basés Agent-Apprenant. Pour fonder notre modèle et renforcer ses propriétés de généralité et de généricité, nous avons défini une assise conceptuelle. Cette dernière est formée par un ensemble de concepts liés à l'apprentissage automatique, qui nous ont servi de base pour la définition des différents composants, du modèle, ainsi que leurs relations d'interdépendances.

Les composants du modèle, ont été définis avec un raffinement de ceux liés directement à l'apprentissage automatique. Le Composant Bases d'Apprentissage a été décomposé en sous composants, où le Générateur d'Exemples d'enTraînement peut assurer la construction d'une base d'exemples et son enrichissement par les données perçues. Les entrées nécessaires à chaque approche d'apprentissage sont ainsi prévues, par le modèle; et ses différents composants collaborent pour permettre, aux systèmes basés Agent-Apprenant, d'améliorer leurs performances par apprentissage automatique.

Le modèle proposé est indépendant de tout domaine et de toute approche ou méthode d'apprentissage. Sa propriété de généralité lui permet d'être instancié et réutilisé par différentes applications; sa généricité lui permet de couvrir les différentes approches et méthodes d'apprentissage automatique.

3. Instanciation du modèle général et générique

Dans un but de réutilisation, par instanciation, le modèle que nous avons proposé a été construit avec une réflexion centrée sur l'apprentissage automatique générique et sur l'indépendance de tout domaine d'application. L'instanciation consiste à réutiliser le modèle proposé avec ses composants génériques, constituant la base des systèmes apprenants, et l'étendre par les spécificités du domaine. Après l'établissement du modèle générique, nous l'avons projeté et instancié par des domaines d'applications des systèmes basés Agent-Apprenant. Cette thèse a présenté deux instanciations destinées respectivement à la prise en charge intelligente du courrier électronique et au diagnostic des pathologies des plantes.

Pour la prise en charge intelligente des E-mails, le système se base sur un apprentissage inductif à partir des pratiques de son usager. Le composant de perception est en observation continue de l'usager afin de getter et capter toute action de ce dernier. Basé sur les exemples d'entraînement, construits par un générateur

automatique, l'algorithme ID3 est invoqué pour un apprentissage de connaissances. Ces dernières enrichissent la base des connaissances, qui est la source des inférences intelligentes assumées par le composant de raisonnement. La performance de la prise en charge de la classe des tâches de ce domaine, par ce système basé Agent-Apprenant Personnel, s'améliore avec son expérience et son apprentissage des profils de l'utilisateur.

La deuxième instanciation a été implémentée sous la plateforme JADE. Cette dernière a été utilisée pour faciliter l'extension future de cette instanciation vers un système multi-agents. Les connaissances préalables et les exemples d'entraînement disponibles, pour le domaine d'agriculture, sont utilisés par l'algorithme FOIL pour un apprentissage de connaissances plus générales. Ces dernières servent de base pour les prises de décisions, du système basé Agent-Apprenant Intelligent de diagnostic des pathologies des plantes. La disponibilité des connaissances préalables renforce le processus d'apprentissage et par conséquent la performance de la prise en charge du diagnostic des pathologies des plantes, par ce système, est en amélioration continue.

3. Perspectives

Ayant contribué au thème de l'apprentissage automatique dans un Agent, le travail réalisé dans cette thèse ouvre des axes et voies de recherches. Plusieurs perspectives paraissent intéressantes, entre autres:

1. La représentation du modèle par un formalisme ou un langage de description

La représentation du modèle proposé par un formalisme ou un langage de description permet de l'exprimer avec des notations bien déterminées et pourra aider dans son exploitation plus ou moins automatique. Il serait intéressant d'utiliser un Langage de Description d'Architectures; un ADL permet de bien décrire les composants, du modèle de systèmes basés Agent-Apprenant, ainsi que leur interactions. Nous proposons d'opter pour le langage AgentUML pour représenter les différents aspects de notre modèle; les motivations principales de cette orientation sont:

- AgentUML est une extension d'UML qui est devenu un standard de modélisation largement utilisé. Ainsi une représentation avec AgentUML permettra une large exploitation de notre modèle.
- AgentUML prend en charge le concept Agent et offre des éléments permettant de représenter la propriété de généricité de notre modèle.

2. Outils d'aide au développement

Il serait aussi intéressant d'étendre des outils de développement existants, basés sur des langages de description, par des éléments de

conception orientés Agent. Et disposer ainsi d'outils d'aide au développement de systèmes basés Agent Apprenant, surtout en phases de conception et de production de code. Ceci permettra de tirer le maximum de profit du modèle proposé en aidant le développeur d'un système basé Agent Apprenant spécifique dans:

- Le raffinement des représentations conceptuelles génériques.
- L'extension de ces représentations par les spécificités de son domaine.
- La génération automatique d'une partie du code du système basé Agent Apprenant, à développer, par un enrichissement du code abstrait et générique associé au modèle.

3. *Instanciations du modèle par des systèmes multi Agents*

Il serait intéressant d'instancier ce modèle par des applications de systèmes multi Agents (SMA). Ce type d'instanciations nécessite des extensions aux entrées et sorties du composant d'apprentissage automatique du modèle. Ces extensions concernent les perceptions et les entrées issues des différents Agents du SMA, ainsi que leur partage de la base des connaissances. La configuration vers laquelle converge cette perspective concerne un SMA formé par un ensemble d'Agents dont l'un est Apprenant; les autres Agents utilisent les connaissances apprises par l'Agent-Apprenant. Donc, des interactions indirectes par partage de la base des connaissances sont à prévoir.

Cette configuration permet l'extension de la deuxième instanciation, présentée au chapitre 5, vers un SMA, basé sur un Agent Apprenant, pour le diagnostic des pathologies des plantes à utiliser par plusieurs usagers au sein d'un centre ou d'un laboratoire d'agriculture. Une autre instanciation intéressante, avec cette configuration, sera une application réseau en médecine. Avec cette application, un ensemble d'Agents intelligents de diagnostic, utilisés par différents médecins d'une spécialité donnée, partagent les connaissances apprises par un Agent-Apprenant du SMA. Les entrées d'apprentissage seront toutes les perceptions et collectes des différents Agents intelligents de diagnostic. Pour ce type d'instanciations, le sous composant maintient de la cohérence, de la base des connaissances, devra être robuste afin d'éviter les incohérences possibles, dues aux entrées multiples issues des divers usagers du système.

Références et bibliographie

- [**Aha et al., 1991**]: D. Aha, D. Kibler, et M. Albert. Instance-based learning algorithms. *Machine Learning*, 6, pp: 37-66. 1991.
- [**Agha, 1986**]: G. Agha. *ACTORS: A Model of Concurrent Computation in Distributed Systems*. The MIT Press, Cambridge, MA. 1986.
- [**Agha et Hewitt, 1988**]: G. Agha et C. Hewitt. Concurrent programming using actors. In Y. Yonezawa et M. Tokoro, editors, *Object-oriented Concurrent programming*, MIT Press. 1988.
- [**Allen et al., 1990**]: J. Allen, J. Hendler et A. Tate. *Reading in planning*. Morgan Kaufmann Publishers: San Mateo, CA. 1990.
- [**Allen et al., 2007**]: J. Allen, N. Chambers et G. Ferguson. *Plow: A Collaborative task learning Agent*. Association for the Advancement of Artificial intelligence. (www.aaai.org). 2007.
- [**Alliot et Schiex, 1994**]: J. M. Alliot et T. Schiex. *Intelligence Artificielle & Informatique Théorique*. Editions Cépadués, Toulouse. 1994
- [**Amblard et Phan, 2006**]: F. Amblard et D. Phan. *Amblard and Phan. Modélisation et simulation multiagents: applications pour les Sciences de l'homme et de la société*. Eds, Lavoisier, Paris, France. 2006.
- [**Armstrong et al., 1995**]: R. Armstrong, D. Freitag, T. Joachims et T. Mitchell. Webwatcher: A learning apprentice for World Wide Web. *Proceedings of AAAI Spring Symposium on information Gathering*, pp:6-12.1995.
- [**Bellifemine et al., 1999**]: F. Bellifemine et A. Poggy, G. Rimassa, P. Turci. *An Object-Oriented Framework to Realize Agent Systems*. 1999.
- [**Bellman, 1978**]: R.E. Bellman. *An Introduction to Artificial Intelligence: Can computers think?* Boyd et Fraser Publishing Company, San Francisco. 1978.
- [**Berthet et al., 1992**]: S. Berthet, Y. Damazeau et O. Boissier. Knowing each other better. 11th workshop of DAI, pp: 23-42. 1992.
- [**Billsus et Pazzani, 1999**]: D. Billsus et M. Pazzani. A Personal News Agent that Talks, learns and explains. *Proceeding of the third international conference on Autonomous Agents*, Seattle Wa, pp: 268-275. 1999.
- [**Booch, 1994**]: G. Booch. *Object Oriented Analysis and design*. Addison Wesley, Reading, Ma. 1994.
- [**Bourret et al., 1991**]: P. Bourret, J. Reggia et M. Samuelides. *Réseaux Neuronaux : une Approche Connexionniste de l'Intelligence Artificielle*. France: Editions Teknea. 1991.

- [**Brooks, 1986**]: R.A. Brooks. A Robust layered control system for a mobile robot. IEEE journal of Robotics and Automation, 2,1, pp: 14-23. 1986.
- [**Brooks, 1991a**]: R.A. Brooks. Intelligence without reason. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, Australia, pp: 569-595. 1991a.
- [**Brooks, 1991b**]: R.A. Brooks. Intelligence without representation. Journal of Artificial Intelligence, 47, pp: 139-159. 1991b.
- [**Buchanan, 1972**]: B.G. Buchanan, E.A. Feigenbaum et N.S. Sridharan . Heuristic theory formation: Data interpretation and rule Formation. Machine intelligence, Halsted press, Wiley 7, pp:967-290. 1972.
- [**Bratman, 1987**]: M.E. Bratman. Intentions, Plans and Practical reason. Havard University Press, Cambridge, MA. 1987.
- [**Bratman, 1988**]: M.E. Bratman. Planning and the stability of intentions. Minds and Machines, 2, pp:1-16. 1988.
- [**Bratman et al., 1992**]: M.E. Bratman, D.J. Israel et M.E. Pollack. Plans and ressource bounded practical reasoning. Journal of Computational Intelligence, 4, pp: 349-355. 1992.
- [**Brustoloni, 1991**]: J.C. Brustoloni. Autonomous Agents: Characterization and Requirements. Carnegie Mellon Technical Report CMU-CS-91-204, Pittsburgh: Carnegie Mellon University. 1991.
- [**Carzaniga et al., 1997**]: A. Carzaniga, G.P. Picco et G. Vigna. Designing distributed applications with mobile code paradigms, 19 ième international conference on Software Engineering. ACM press, 1997.
- [**Castelfranchi, 1995**]: C. Castelfranchi. Commitments : From individual intentions to groups and organizations, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS95), San Francisco, CA, pp: 41-48. 1995.
- [**Chapman, 1987**]: D. Chapman. Planning for conjunctive goals. Journal of Artificial Intelligence, 32, pp: 333-378. 1987.
- [**Charniak et McDermott, 1985**]: E. Charniak et D. McDermott. Introduction to Artificial Intelligence. Addison Wesley, Reading. Massachusetts. 1985.
- [**Chen et Sycara, 1998**]: L. Chen et K. Sycara. Webmate: A personal Agent for browsing and searching. In proceeding of the second international conference on Autonomus Agents, Menniapolis, 1998.
- [**Chung et Wu, 1997**]: K. T. Chung et C. H. Wu. Dynamic scheduling with Intelligent Agent. Metra Application, Palo Alto, 1997.
- [**Cohen et Levesque, 1990**]: P.R. Cohen et H.J. Levesque. Intention In Choice with Commitment. Artificial Intelligence, 42, pp: 213-261. 1990.
- [**Cornuéjols et Miclet, 2002**]: A. Cornuéjols et L. Miclet. Apprentissage artificiels : concepts et algorithmes, Eyrolles. 2002.
- [**Clément, 1989**]: F.M. Clément. Les réseaux de neurones. Editions Eyrolles, 1989.
- [**Crawford et Veloso, 2007**]: E. Crawford et M. Veloso. An experts approach to strategy selection in multiagent meeting scheduling. Auton Agent Multi-Agent Systems, 15, pp:5-28. 2007.

- [**David, 2006**]: Mc. David. The ID3 Decision tree Algorithm. CSE520, Data mining, pp: 1-8. 2006.
- [**Demazeau et Müller, 1990**]: Y. Demazeau et J.P. Müller. Decentralized Artificial Intelligence. Yves Demazeau and Jean-Pierre Müller (Eds), Elsevier Science Publisher, pp: 3-16. 1990.
- [**Dejong et Mooney 1986**]: G. Dejong et R. Mooney. Explanation Based Learning: An alternative view. Machine Learning, 1, 2, pp: 145-176. 1986.
- [**Dietterich, 2002**]: T.G. Dietterich. Machine learning for sequential data: A review. In Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Springer-Verlag, London, UK, pp: 15-30. 2002.
- [**Dietterich et Flann, 1995**]: T.G. Dietterich et N.S. Flann. Explanation Based Learning and Reinforcement Learning: a unified view. In Proceedings of 12th International Conference on Machine Learning, pp:176-184. 1995.
- [**Dougherty et al., 1995**]: J. Dougherty, R., Kohavi et M. Sahami. Supervised and unsupervised discretization of continuous attributes. In Proceedings of 12th International Conference on Machine Learning, pp: 194-2002. 1995.
- [**Doorenbos et al., 1997**]: R. Doorenbos, O. Etzioni et D. Weld. Ascaleable comparision-shopping Agent for the world wide web. In proceeding of the first international conference on Autonomus Agents, Marina del Rey, 1997.
- [**Duda et al., 2000**]: R. O. Duda, P. E. Hart et D. G. Stork. Pattern Classification. New York : Wiley Interscience. 2000.
- [**Durfee et al., 1989**]: E.H. Durfee, V.R. Lesser, D.C. Corkill. Trends in cooperative distributed problem solving, IEEE Transactions on Knowledge and Data Engineering, 1, 1, pp: 63-83. 1989.
- [**Edwards et Davies, 1993**]: P. Edwards et W. Davies. A Heterogeneous Multi-Agent Learning System. In Proceedings of the Special Interest Group on Co-operating Knowledge Based Systems, S. M. Deen (Ed.), pp:163-184. 1993.
- [**Etzioni, 1996**]: O. Etzioni. Moving up the information food chain: deploying softbot on the world wide web. In proceedings of thirteenth national conference on Artificial Intelligence, Portland, 1996.
- [**Etzioni et Weld 1994**]: O. Etzioni et D. Weld. A Softbot-Based Interface to the Internet. Communications of the ACM, July 37,7, pp: 72-79. 1994.
- [**Ferber, 1995**]: J. Ferber. Les Systemes Multi-Agents. Vers une Intelligence Collective. Interedition, Paris. 1995.
- [**Ferber, 1997**]: J. Ferber. Les Systemes Multi-Agents.Un apercu général. Techniques et sciences informatiques, 16, 8. 1997.
- [**Ferguson et al., 2009**]: G. Ferguson, J. Allen, L. Galescu, J. Quinn et M. Swift. CARDIAC: An Intelligent Conversational Assistant for Chronic Heart Failure Patient Heath Monitoring. Proceedings of the AAAI Symposium on Virtual Healthcare Interaction, Arlington, VA, November. 2009.
- [**Finin et al., 1992**]: T. Finin, R. Fritzson, et D. McKay. An overview of KQML: A Knowledge Query and Manipulation Language. Technical report, University of Maryland Baltimore Country, 1992.

- [**FIPA, 2006**]: FIPA. Foundation for Intelligent Physical Agents. FIPA Agent Management Specification, Foundation for Intelligent Physical Agents, 2006.
- [**Fikes et Nillson,1971**]: R.E. Fikes et N. Nillson. A new approach to the application of theorem proving to problem solving. *Journal of Artificial Intelligence*, 5, 2, pp: 189-208. 1971.
- [**Fischer et al., 2004**]: F. Fischer, M. Rovatsos et G. Weiss. Hierarchical reinforcement learning in communication-mediated multiagent coordination. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp: 1334-1335. 2004.
- [**Gold, 1965**]: E.M. Gold. Limiting recursion. In *Journal of Symbolic Logic*, 30, 1, pp: 28-46. 1965.
- [**Goldberg, 1994**]: D. Goldberg. *Algorithmes génétiques*, Addison-Wesley, France. 1994.
- [**Griffeth et Velthuijsen, 1994**]: N.D. Griffeth et H. Velthuijsen. The negotiating agents approach to run time feature interaction resolution. In L.J. Bouma and H. Velthuijsen, editors, *Feature interaction in Telecommunications Systems*, IOS Press, pp: 217-235, 1994.
- [**Guarino et Giaretta, 1995**]: N. Guarino et P. Giaretta. Ontologies and Knowledge bases: towards a terminological clarification, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. N. Mars, IOS Press, pp: 25-32. 1995.
- [**Guessoum et Briot, 1999**]: Z. Guessoum et J-P. Briot. From Active Objects to Autonomous Agents. *IEEE Concurrency*, 7,3, pp :68-76. 1999.
- [**Hakansson et al., 2010**]: A. Hakansson, N. Thanh Nguyen, R. Hartung, R.J. Howlett et L.C. Jain. Conference report of the third KES Symposium on Agent and Multi-Agent Systems: Technologies and Applications. *International Journal of Knowledge-based and Intelligent Engineering Systems*, IOS Press, 14, pp: 45-47. 2010.
- [**Hammoud et Sahnoun, 2005**]: D. Hammoud et Z. Sahnoun. La Technologie Agent et l'Apprentissage Automatique. *Proceeding of the International Conference on Productique, CIP, Tlemcen, Algeria., December. 3-4*, pp: 1-12. 2005.
- [**Hammoud et al., 2008**]: D. Hammoud, Z. Sahnoun, R. Kebache et B. Benelmadani. A Model for Personal Learning Agents: with an Inductive Learning Agent Based System. *Proceedings of the International Conference, Intelligent Systems and Agents ISA, Amsterdam, the Netherlands., ISBN: 978-972-8924-60-7 c 2008 IADIS, july. 22-24*, pp: 166-171. 2008.
- [**Hammoud et Sahnoun, 2008**]: D. Hammoud et Z. Sahnoun. L'apprentissage automatique dans un Agent. *Journée scientifique sur le Genie Logiciel & les Architectures des systèmes, Constantine, Algeria., December. 3-3*, pp: 1-27. 2008.
- [**Hammoud et Sahnoun, 2009a**]: D. Hammoud et Z. Sahnoun. Suitable Machine Learning Methods for Agent-based Systems. *Proceedings of the International Conference on Electronic, Technologies of Information and Telecommunication SETIT, Hammamet., Tunisia, March.22-26*, pp: 1-8. 2009.
- [**Hammoud et Sahnoun, 2009b**]: D. Hammoud et Z. Sahnoun. Suitable Machine Learning Methods for Agent-based systems, selected by SETIT and published in *ICGST International Journal of Artificial Intelligence and Machine Learning AIML 9, 2*, pp: 45-52. 2009.

- [**Hammoud et al., 2011**]: D. Hammoud, R. Maamri et Z. Sahnoun. Machine Learning in an Agent: A Generic Model and an Intelligent Agent Based on Inductive Decision Learning. *Journal of Artificial Intelligence*, 4, 1, pp: 29-44. 2011.
- [**Hayes-Roth, 1995**]: B. Hayes-Roth. An Architecture for Adaptive Intelligent Systems, *Artificial Intelligence: Special Issue on Agents and Interactivity*, 72, pp: 329-365. 1995.
- [**Haton, 1990**]: P. Haton. Les systèmes à base de connaissances. In *Artificial Intelligence IV: Methodology, Systems and applications*, Proceedings of AIMSA. 1990.
- [**Hewitt et Inman, 1991**]: C. Hewitt et J. Inman. DAI betwist and between: from intelligent Agents to open systems science. *IEEE transaction*, 21, 6, pp: 1409-1418. 1991.
- [**Holland, 1975**]: J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. 1975.
- [**Hopfield, 1982**]: J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79, pp: 2554-2558. 1982.
- [**Hsinchen et al., 1996**]: C. Hsinchen, H. Andrea et N. Jay. Toward intelligent meeting agents. *IEEE computer*, pp: 62-69. 1996.
- [**Huhns et Weiss, 1998**]: M. Huhns et G. Weiss (Eds.). Special Issue on Multi-agent Learning, *Machine Learning Journal*. 33, pp:2-3. 1998.
- [**Igor, 1987**]: A. Igor. *Designing intelligent System: an introduction*. Great Britain, Worcester. 1987.
- [**Ingrand et al., 1992**]: F.F. Ingrand, M. P. Georgeff et A. S. Rao. An architecture for real time reasoning and system control. *IEEE Expert*, 7,6, 1992.
- [**Iredi et al., 2001**]: S. Iredi, D.Merkle et M. Middendorf. Bi-Creterion optimization with multi coloy ant algorithms. *First International Conference on multi optimization*, Springer verlag, pp: 359-372. 2001.
- [**Jennings et Wooldridge, 1998**]: N.R.Jennings et M.Wooldridge. *Applications of intelligent Agents*. Queen Mary ET Westfield College, University of London. 1998.
- [**Jennings et al., 1998**]: N.R. Jennings, K. Sycara et M. Wooldridge. A Roadmap of Agent Research and development. *Journal of Autonomus Agents and Multi-Agents Systems*, 1, pp: 275-306. 1998.
- [**Jennings, 2001**]: N.R. Jennings. An Agent-based Approach for Building Complex Software Systems. *Communication of the ACM*, 44(4), pp:35-41. 2001.
- [**Kaelbling, 1993**]: L.P. Kaelbling. Learning to achieve goals. In *IJCAI*, pp: 1094-1099. 1993.
- [**Kayser, 1997**]: D. Kayser. *La Représentation des connaissances*. Hermes, Paris. 1997.
- [**Kazar, 2005**]: O. Kazar. Un modèle d'Agent à raisonnement adaptable. Thèse de Doctorat en sciences, laboratoire LIRE, université de Constantine, Algérie. 2005.
- [**Kedar-Cabelli et McCarty, 1987**]: S. Kedar-Cabelli et T. McCarty. Explanation Based Generalization as resolution theorem proving. *Proceedings of the fourth International Workshop on Machine learning*, pp: 383-389. 1987.

- [**Kernchen et al., 2007**]: S. Kernchen, F. Zbrog et R. Dumke: ABEL-GUI: An Agent-Based Graphical User Interface for ELearning, Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST 2007), Barcelona, Spain, pp. 491-494. 2007.
- [**Kolodner, 1993**]: J. Kolodner. Case-Based Reasoning. Morgan Kaufmann Publishers. 1993.
- [**Kozierok et Maes, 1993**]: R. Kozierok et P. Maes. A learning interface agent for scheduling meetings, Proceedings of the 1st international conference on Intelligent user interfaces, United States, pp: 81-88. 1993.
- [**Koyama et al., 2001**]: A. Koyama, L. Barolli, A. Tsuda et Z. Cheng. An Agent-Based Personalized Learning System. Proceedings of the 15th International Conference on Information Networking (ICOIN'01), Japan, pp: 895-899. 2001.
- [**Kurzweil, 1990**]: R. Kurzweil. The Age of Intelligent Machines. Cambridge, MIT press. 1990.
- [**Lang, 1995**]: K. Lang. NewsWeeder: Learning to Filter NetNews, In ICML95 Conference Proceedings, pp: 331-339. 1995.
- [**Lavrac et Dzeroski, 1994**]: N. Lavrac et S. Dzeroski. Inductive logic programming. Techniques and applications. Edition Ellis Horwood, New York, pp:3-38. 1994.
- [**Legros, 2004**]: J. Legros. Les Secrets de Monsieur TRUFFAUT. Ed Tallandier. France.
- [**Lenat, 1977**]: D.B. Lenat. Automated theory formation in mathematics. Proceedings de la 5ieme conference en intelligence artificielle, pp : 833-842. 1977.
- [**Lieberman, 1995**]: H. Lieberman. Letizia : An Agent that assists web browsing. In proceedings of fourteenth international conference on Artificial Intelligence, Montreal, pp: 924-929. 1995.
- [**Lieberman et al., 2001**]: Lieberman, H., E. Rosenzweig, et P.A. Singh. An Agent for annotating and retrieving images. IEEE computing. 134,7, pp: 57-61. 2001.
- [**Ljunberg et Lucas, 1992**]: M. Ljunberg et A. Lucas. The OASIS air traffic management system. In proceedings of the second pacific international conference on Artificial Intelligence, Korea, 1992.
- [**Maes, 1990**]: P. Maes. Designing Autonomous Agents. Cambridge, MIT press. 1990.
- [**Maes, 1992**]: P. Maes. Learning behavior networks from experience. Proceedings of the First Conference on Artificial Life, F. J. Varela and P. Bourguin (in eds.) MIT Press/Bradford Books, 1992.
- [**Maes, 1994**]: P. Maes. An Agent that reduce information overload. Communications of the ACM, July, 37,7, pp: 31-40. 1994.
- [**Maes et Schneiderman, 1994**]: P. Maes et B. Schneiderman. Direct Manipulation vs. Interface Agents: a Debate. In Interactions, Vol. IV, 6, ACM Press, 1997
- [**Masini et al., 1989**]: G. Masini, A. Nappoli, D., Colnet et K. Tombere. Les langages à objets : langages de classes, langages de frames, langages d'acteurs. Inter Editions. 1989.
- [**McCarthy et al., 1955**]: J. McCarthy, M. Minsky, N. Rochester et C. Shannon. Proposal for the Darmouth summer research project on Artificial Intelligence. Technical report, Darmouth College. 1955.

- [**McCulloch et Pitts, 1943**]: W.S. McCulloch et W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, pp: 115-133. 1943.
- [**Mencke et Dumke, 2007**]: S. Mencke et R. Dumke. A Framework for Agent-Supported E-learning. *Conference ICL2007, Austria*, pp1-20. 2007.
- [**Michalski et Chilausky, 1980**]: R.S. Michalski et R. Chilausky. Learning by being told and Learning from examples. *Policy Analysis and Informations systems*, 4, 2, R.S. Michalski, J.G. Carbonell, T. Mitchell Eds, Morgan Kaufman, pp: 163-190. 1980.
- [**Michalski et al., 1993**]: R.S. Michalski, J.G. Carbonell, T. Mitchell et Y. Kodratoff. *Apprentissage symbolique: une approche de l'intelligence artificielle*. Tome 1 et 2, Editions Cépadues, Toulouse. 1993.
- [**Michalski et al., 1998**]: R.S. Michalski, I. Bratko et M. Kubat. *Machine Learning and Data Mining : Methods and Applications.*, John Wiley & Sons, Chichester, England, 1998.
- [**Minsky, 1975**]: M. Minsky. A framework for representing knowledge. In *Psychology of computer vision*, P.H. Winston Eds, McCraw-hill, New York, pp: 211-277. 1975.
- [**Mitchell, 1977**]: T.M. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *proceedings of fifth international joint conference on Artificial Intelligence*, pp: 305-310. 1977.
- [**Mitchell et al., 1983**]: T.M. Mitchell, P. Utgoff et R. Banerji. Learning by experimentation, Acquiring and refining problem solving heuristics. *Dans Machine learning: an artificial intelligence approach*.
- [**Mitchell et al., 1986**]: T.M. Mitchell, R. Keller et K. Cabeli. Explanation based generalisation: a unifying view. *Machine learning*, vol 1, pp: 47-80. 1986.
- [**Mitchell et Thrun, 1993**]: T. Mitchell et S. Thrun. *Explanation-Based Learning: A Comparison of Symbolic and Neural Network Approaches*. *Proceedings of the Tenth International Conference on Machine Learning, CA - U.S.A.*: P. Utgoff (Ed.), Morgan Kaufmann, San Mateo. 1993.
- [**Mitchell et Garuana, 1994**]: T. Mitchell et R. Garuana. Experience with a learning personal assistant. *Communication of the ACM*, 37, 7, pp: 81-91. 1994.
- [**Mitchell, 1997**]: T. Mitchell, 1997. *Machine learning*. Edition MIT press, USA.
- [**Mitchell, 1999**]: T. Mitchell. *Machine Learning and Data Mining*. *Communications of the ACM*, 42, 11, pp: 31-36. 1999.
- [**Mitchell, 2006**]: T. Mitchell. *The Discipline of Machine Learning*. CMU-ML-06-108, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 pp: 1-7. 2006.
- [**Mjolsness et Decoste, 2001**]: E. Mjolsness et D. Decoste. *Machine learning for science: state of the art and future prospects*. *Science*, 293, pp: 2051-2055. 2001.
- [**Muggleton, 1991**]: S. Muggleton. *Inductive Logic Programming*. *New Generation Computing Edition*, pp: 295-318. 1991.
- [**Müller et al., 1995**]: J. P. Müller, M. Pischel, et M. Thiel. *Modeling Reactive Behaviour in Vertically Layered Agent Architectures*, in Wooldridge and Jennings Eds., *Intelligent Agents*, Berlin: Springer-Verlag, pp : 261-276. 1995.

- [**Müller, 1996**]: J. P. Müller. The Design of Intelligent Agents – A layered Approach. Lecture Notes in artificial Intelligence, Springer. 1996.
- [**Myers et Rose, 1996**]: J. Myers et M. Rose. Post Office Protocol-Version 3.pp:1-23.1996.
- [**Naderi et al., 2007**]: H. Naderi, B. Rumber, et J. Pinon. An efficient Information Retrieval System by incorporating the user Profile. Adaptative Multimedia Retrieval, LNCS 4398 (), pp:247-257. 2007.
- [**Nauck et Kruse, 1996**]: D. Nauck et R. Kruse. Neuro-Fuzzy Classification with NEFLASS. In: Operational Research Proceedings 1995. Springer, Berlin, pp:294-299. 1996.
- [**Ndumu et al., 1998**]: D. T. Ndumu, J. C. Collis et H. Nwana. Towards Desktop Personal Travel Agents. BT Technologie Journal,16, 3,pp: 69-78. 1998.
- [**Negroponte, 1995**]: N. Negroponte. Being Digital. Hodder and Stoughton. 1995.
- [**Nilsson, 1996**]: N. J. Nilsson. Decision Trees. Chapter 6 of Introduction to Machine Learning. 1996.
- [**Nilsson, 1997**]: N. J. Nilsson. Introduction to Machine Learning (Draft of a proposed new textbook). U.S.A: Robotics Lab., Dept. of Computer Science, Stanford University. 1997.
- [**Nilsson, 1980**]: N. J. Nilsson. Principles of Artificial Intelligence. California -U.S.A: Tioga Press, Palo Alto. 1980.
- [**Nwana et Wooldridge, 1996**]: H.S. Nwana et M. Wooldridge. Software Agent: An overview. Knowledge Engineering Review, 11, 3, pp: 205-244. 1996.
- [**Okamoto et al., 2009**]: S. Okamoto, K. Sycara, et P. Scerri. Personal Assistants for Human Organizations. in Multi-Agent Systems - Semantics and Dynamics of Organizational Models, Ed. V. Dignum, IGI-Global, Hershey, Pennsylvania, U.S.A. 2009.
- [**Parunak, 1987**] : H.V. D. Parunak. Manufacturing experience with the contract net. In M. Huhns, editor, Distributed Artificial Intelligence, Morgan Kaufman, pp: 285-310, 1987.
- [**Poole et al., 1998**]: D. Poole, A.K. Mackworth et R. Goebel. Computational intelligence: A logical approach. Oxford University Press, UK. 1998.
- [**Postel, 1982**]: J.B. Postel. Simple Mail Transfer Protocol. pp: 1-67. 1982.
- [**Quinlan, 1979**]: J. R. Quinlan. Discovering Rules by Induction from Large Collections of Examples. In: D. Michie (Ed.), Expert Systems in the Micro Electronic Age. Edimburgh, UK: dimburgh University Press. 1979.
- [**Quinlan, 1986**]: J. R. Quinlan. Induction of decision trees. Machine learning, 1,1, pp: 81-106. 1986.
- [**Quinlan, 1990**]: J. R. Quinlan. Learning logical definitions from relations. Machine Learning, 5, pp: 239-266. 1990.
- [**Quinlan et Cameron-Jones, 1993**]: J. R. Quinlan et R. Mike Cameron-Jones. FOIL: A midterm report Proceedings of the 6th European conference on Machine Learning. Lecture Notes in Computer science, Springer-Verlag, pp: 3-20. 1993.

- [Rao et Georgeff, 1991]:** A. S. Rao et M. P. Georgeff. Modelling Agents within a BDI – Architecture. In R. Fikes and E. Sandewall, editors, Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91), pp: 473-484. 1991.
- [Rao et Georgeff, 1995]:** A. S. Rao et M. P. Georgeff. BDI – agents: from theory to practice. Proceedings of the First International Conference on Multi – Agent Systems, San Francisco. 1995.
- [Rey et al., 2007]:** A. Rey, J.R. Debove et Collaborateurs. Le Grand Robert de La Langue Francaise. Eds Le Robert, 2007.
- [Rovatsos et al.,2003]:** M. Rovatsos, G. Weiss et M. Wolf. Multiagent learning for open systems: a study in opponent classification. In E. Alonso, D. Kudenko and D. Kazakov (Eds.), Adaptive agents, Lecture Notes in Computer Science, Volume 2636. Springer-Verlag, pp: 66-8. 2003.
- [Russel et Norvig, 2006]:** S. Russel et P. Norvig. Intelligence Artificielle. 2nd Edn, Pearson Education, France. 2006.
- [Samuel, 1959]:** A.L. Samuel. Some studies in Machine learning using the game of checkers. IBM journal of Research and development, 3, pp: 211-229. 1959.
- [Schwab et Koychev, 2000]:** I. P. Schwab et W. Koychev. Learning to Recommend from Positive Evidence, Proceedings of Intelligent User Interfaces, ACM Press, pp: 241-247. 2000.
- [Selker, 1994]:** T. Selker. Coach: A teaching Agent that learn. Communication of the ACM, 37, 7, pp: 92-104. 1994.
- [Sen et Weiss, 1999]:** S. Sen et G. Weiss. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, chapter Learning in Multi-agent systems, MIT Press, Cambridge, MA, pp: 259-298. 1999.
- [Sheirin et Lieberman, 2001]:** S. Sheirin et H. Lieberman. Intelligent profiling by examples. MIT medi laboratory, Combridge, pp: 1-7. 2001.
- [Sheth, 1994]:** B. Sheth. A learning Approach to Personalized information Filtring. Master's thesis, Department of Electrical engineering and computer science, MIT. 1994.
- [Shirky, 2005]:** C. Shirky. Ontology is overrated : Categories, links, and tags. O'Reilly ETech conference. 2005.
- [Shoham, 1993]:** Y. Shoham. Agent Oriented Programming. Artificial Intelligence , 60,1, pp: 51-92, 1993.
- [Shoham et Cousins, 1994]:** Y. Shoham et S. B. Cousins. Logics of Mental Attitudes in AI: a very preliminary survey. In G. Lakemeyer and B. Nebel (eds.) Foundations of Knowledge Representation and Reasoning, Springer Verlag, pp: 296-309. 1994.
- [Sigaut et Buffet, 2008]:** O. Sigaut et O. Buffet. Processus Décisionnels de Markov en Intelligence Artificielle. Lavoisier, Paris, France. 2008.
- [Simon, 1983]:** H. A. Simon. Machine Learning : An Artificial Intelligence Approach, chapter Why should machine learn ? CA : Morgan Kaufmann, Los Altos. 1983.
- [Smith et al., 1994]:** D. C. Smith, A. Cypher et J. Spohrer. Programming Agents Without a Programming Language, Communications of the ACM, 37, 7, pp: 55-67. 1994.

- [**Sian, 1991**]: S. Sian. Extending Learning to Multiple Agents. In Proceedings of the European Working Session on Learning (EWSL91), Y. Kodratoff (Ed.), SpringerVerlag, pp: 458-472. 1991.
- [**Stone et Veloso, 2000**]: P. Stone et M. Veloso. Multiagent Systems: A survey from the machine learning perspective. *Autonomous Robots*, 8,3, pp:345-383. 2000.
- [**Sun, 2001**]: R. Sun. Introduction to sequence learning. In *Sequence Learning - Paradigms, Algorithms, and Applications*, London, UK, Springer- Verlag, pp: 1-10. 2001.
- [**Sun, 1992**]: R. Sun. Connectionist Models of Rule-based Reasoning. In: *AISB Quarterly, Special Issue on Hybrid Systems*. 79, pp. 21-24. 1992.
- [**Sutton et Barto, 1998**]: R. Sutton et A.G. Barto. Reinforcement Learning: an introduction. MIT Press, USA. 1998.
- [**Sutton et McCallum, 2006**]: C. Sutton et A. McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor et Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press. 2006.
- [**Sybil et Lieberman, 2001**]: S. Sybil et H. Lieberman. Intelligent profiling by example, Proceedings of the 6th international conference on Intelligent user interfaces, January 14-17, Santa Fe, New Mexico, United States, pp:145-151. 2001.
- [**Tama et Wooldrige, 2004**]: V. Tama et M. Wooldrige. Serser: Searching for semantic Web content. Department of computer science university of Liverpool L69 united kingdom, pp 1-5. 2004.
- [**Tecuci et al., 2002**]: G. Tecuci, M. Boicu, et D. Marcu. Training and using Disciple Agents: a case study in the gravity domain military center. *Artificial Intelligence Magazine*, 23, 4, pp: 51-66. 2002.
- [**Thomas et al., 2007**]: V. Thomas, C. Bourjot, V. Chevrier. Construction de systèmes multi-agents par apprentissage collectif à base d'interactions. *Revue d' Intelligence Artificielle*, pp: 1-19. 2007.
- [**Towel et Shavlik, 1989**]: G. Towel et J. Shavlik. An approach to combining explanation based and neural learning algorithms, *connection Science*, 1, pp: 233-255. 1989.
- [**Treuil et al., 2008**]: J.P. Treuil, A. Drogoul et J.D. Zucker. Modélisation et simulation à base d'Agents. Eds Dunod, Paris. 2008.
- [**Truffaut, 2010**]: http://www.truffaut.com/jardinage/maladies-ravageurs-plantes/parasites-maladies-plantes_/recherche/parasite.html#produits.
- [**Turing, 1950**]: A. Turing. Computer machinery and intelligence. *Mind*, 59, pp: 433-460. 1950.
- [**Utgoff, 1989**]: P.E. Utgoff. Induction incrémentale d'arbres de décision. Kluwer academic publishers. 1989.
- [**Vanderveken, 1990**]: D. Vanderveken. On the Unification of Speech Act Theory and Formal Semantics. In P.R. Cohen, J. Morgan, and M.E. Pollack, editors, *Intentions in Communication*, MIT Press, Cambridge, MA, pp : 195-220. 1990.
- [**Vivacqua, 1999**]: A.S. Vivacqua. Agents for expertise location. Proceedings of AAI spring symposium, Stanford, in cyberspace CA, USA. 1999.

- [Wang et al., 2005]: Y.Wang, W. Wang et C. Lin. An Intelligent Semantic Agent for e-Learning Message Communication, *Journal of Information Science and Engineering*, 21, 5, pp: 1031-1051. 2005.
- [Wang et Wang, 1997]: H. Wang et C. Wang. Intelligent Agent in the nuclear industry. *IEEE computer* , 30,11, pp: 28-34. 1997.
- [Wang et Silva, 2008]: Y. Wang et C.W. de Silva. A machine-learning approach to multi-robot coordination. *Engineering Applications of Artificial Intelligence*, 21,3, pp: 470-484, 2008.
- [Watkins et Dayan, 1992]: C.Watkins et P. Dayan. Q-learning. *Machine learning*, 8, pp: 279-292. 1992.
- [Wenyin et al., 2009]: L. Wenyin, H. Tianyong, C. Wei et F. Min. A Web-Based Platform for User-Interactive Question-Answering. *World Wide Web*, 12, pp:107-124. 2009.
- [Weiss, 1996]: G. Weiss. Adaptation and learning in Multi-Agents Systems. *The Lecture Notes in Artificial Intelligence Review*, Springer-Verlag, 1042, pp: 1-21. 1996.
- [Weiss, 1997]: G. Weiss. Distributed artificial intelligence meets machine learning. *Lecture Notes in Artificial Intelligence*, Volume 1221, Springer-Verlag, ISBN 3-540-62934-3, 1997.
- [Weiss, 1999]: G. Weiss. Timeweaver : a genetic algorithm for identifying predictive patterns in sequences of events. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, Morgan Kaufmann, pp. 718-725. 1999.
- [Wilkins, 1988]: D. Wilkins. *Practical Planning: extending the classical AI Planning Paradigm*. Morgan Kaufmann Publishers: san Mateo, CA. 1988.
- [Winston, 1975]: P.H. Winston. *Learning structural description from examples. Psychology of computer vision*, P.H. Winston Eds, McCraw-hill, New York, 1975
- [Witten et Eib, 2005]: I. Witten et F. Eib. *Data mining: Practical Machine Learning Tools and Techniques*”, Second edition, Morgan Kaufmann. 2005.
- [Wooldridge et Jennings, 1995]: M. Wooldridge et N.R. Jennings. Intelligent Agents: theory and practice. *The knowledge Engineering Review* , vol 10, 2, pp:115-152.1995.
- [Wooldridge, 2000]: M. Wooldridge. Intelligent Agents. In: Weiss, G. *Multi-Agent systems: a modern approach to distributed artificial intelligence*. The MIT Press, Second printing. 2000.
- [Yang et al., 2009]: X. Yang, P. Xiang et Y. Shi. Finding User’s Interest Blocks using Significant Implicit Evidence for Web Browsing on Small Screen Devices. *World Wide Web*, 12, pp:213-234. 2009.
- [Zanker et Jessenitschnig, 2009]: M. Zanker et M. Jessenitschnig. Case-studies on exploiting explicit customer requirements in recommender systems. *User Model User-Adaptive Interfaces*, 19, pp:133-166. 2009.
- [Zeigler, 1999]: B. Zeigler. *Theory of modeling and Simulation*. New York, John Wiley, 1973. Second edition par Zeigler. B., T.G. Kim, H. Praehofer and N.Y John Wiley. 1999.
- [Zeng et Sycara, 1997]: D. Zeng et K. Sycara. Benefits of learning in negotiation. *Proceedings of AAI*, Providence, Rhode Island, USA. 1997.