

2 mil. 2995,9

Université de Montréal

Pacha : Marché Virtuel pour agents mobiles

Par

Mounir Youssef

Département d'informatique et  
de recherche opérationnelle

Faculté des arts et sciences

Mémoire présenté à la Faculté des études supérieures  
en vue de l'obtention du grade de  
Maître ès sciences (M. Sc.)  
en informatique

août 2002

©Mounir Youssef, 2002



QA

76

U54

2002

v.052

Université de Montréal  
Faculté des études supérieures

Ce mémoire intitulé :

Pacha : Marché Virtuel pour agents mobiles

Présenté Par :

Mounir Youssef

A été évalué par un jury composé des personnes suivantes :

Président rapporteur :	Esma AÏMEUR
Directeur de recherche :	Claude FRASSON
Membre du jury :	Peter KROPF

Mémoire accepté le : ..... 25 octobre 2002 .....

## Résumé

Les agents intelligents reliés au commerce électronique sont en pleine expansion et risquent de bouleverser les pratiques commerciales modernes. Maintes applications logicielles plus ou moins évoluées surgissent pour satisfaire l'engouement actuel pour le commerce sur le Web. La problématique qui en découle est semblable à celle de la recherche d'informations: face à une offre internationale, éparpillée et pléthorique, le consommateur a de plus en plus de mal à choisir un produit. De manière symétrique, face à une demande encore plus atomisée et exigeante, le producteur n'arrive pas toujours à garantir une offre personnalisée et adaptée aux vrais besoins des consommateurs.

Afin de surmonter les difficultés citées ci-dessus, nous avons développé un nouveau système appelé *Pacha*. Ce dernier combine la technologie des agents intelligents mobiles et celle du commerce électronique. C'est en quelque sorte un marché virtuel qui gère les agents électroniques envoyés par les clients afin d'accélérer le processus de recherche. *Pacha* poursuit trois objectifs principaux: orienter le client selon son niveau d'expertise et lui offrir des solutions données par la technologie CBR ("*Case-Based Reasoning*"), permettre le regroupement des PME (Petites-Moyennes Entreprises) en un marché virtuel, les rendant ainsi compétitives avec les grandes entreprises, et enfin permettre aux clients de se débrancher ou de naviguer en attendant le retour de son agent. Par ailleurs, pour appliquer la norme standard des agents médiateurs dans le commerce électronique, *Pacha* respecte les trois premières étapes du modèle CBB ("*Consumer Buying Behaviour*").

Le modèle ainsi proposé se veut un outil essentiel permettant de faire face aux difficultés inhérentes du marché virtuel toujours en croissance.

**Mots-clé :** Agents Intelligents, CBB, CBR, Agents Mobiles, Marché Virtuel, Salles d'enchères.



## Abstract

The intelligent agents on the net are presently fast expanding and this could turn the actual commercial uses up side down. Various software applications, more or less adapted, take place to satisfy the growing fad for electronic commerce. Unfortunately, these create similar problems to the ones related to the search of information on the net: facing an international offer, scattered and overabundant, the consumer has more and more difficulties to choose a product. In a similar way, manufacturers who are dealing with even more specific and hard demanding requests can hardly guaranty a personal and a custom offer to meet his customer's real needs.

In order to overcome the above difficulties, we developed a new system called *Pacha*. This system combines the mobile intelligent agents' technology with the electronic commerce. We could say that it is a Marketplace, which deals with intelligent agents referred by customers designed to speed the searching processes. The *Pacha* system follows three main goals: leads the customer depending on his expertise level and offers him solutions given by the CBR technology ("Case-Based Reasoning"), allows the groupings of small enterprises so that they can compete with larger enterprises and allows customers to disconnect or surf on the net while they wait for the agent's answer. Also, to apply the standard of the agents as mediators in the electronic commerce, Pasha respects the first three stages of model CBB ("Consumer Buying Behavior").

Our system is meant to be an essential tool to face the inherent difficulties related to the on going growth of the Marketplace.

**Key words:** Intelligent Agents, CBB, CBR, Mobile Agents, Marketplace, Auctions sales.

## Table des matières

<b>RESUME.....</b>	<b>I</b>
<b>ABSTRACT .....</b>	<b>II</b>
<b>TABLE DES MATIERES .....</b>	<b>III</b>
<b>LISTE DES FIGURES .....</b>	<b>VI</b>
<b>LISTE DES TABLEAUX.....</b>	<b>VII</b>
<b>REMERCIEMENTS.....</b>	<b>VIII</b>
<b>INTRODUCTION.....</b>	<b>9</b>
1.1    MOTIVATIONS .....	10
1.2    OBJECTIFS GÉNÉRAUX.....	12
1.3    OBJECTIFS SPÉCIFIQUES.....	12
<b>2    LE COMMERCE ELECTRONIQUE ET LES SYSTEMES D’AGENTS</b>	
<b>INTELLIGENTS.....</b>	<b>14</b>
2.1    COMMERCE ELECTRONIQUE .....	14
2.1.1 <i>Types des transactions du commerce électronique</i> .....	14
2.1.2 <i>Evolution des sites Web commerciaux</i> .....	15
2.1.3 <i>Les débuts du commerce en ligne</i> .....	16
2.1.4 <i>Centres commerciaux virtuels</i> .....	17
2.1.5 <i>Les besoins de médiation</i> .....	18
2.1.6 <i>Connaître le client pour établir une interaction privilégiée</i> .....	18
2.1.7 <i>Permettre de personnaliser l'offre</i> .....	18
2.1.8 <i>Adopter une politique de transparence</i> .....	19
2.1.9 <i>Fournir une assistance aux clients</i> .....	19
2.1.10 <i>Offrir des services complémentaires</i> .....	19
2.2    LES SYSTEMES D’AGENTS INTELLIGENTS .....	21
2.2.1 <i>Pourquoi utilise t-on des agents ?</i> .....	21

2.2.2	<i>Les composantes de l'agent intelligent</i> .....	22
2.2.3	<i>Attributs de l'agent intelligent</i> .....	23
2.2.4	<i>Quelques notions d'agent</i> .....	24
2.2.5	<i>Les agents intelligents dans le commerce électronique</i> .....	25
2.2.6	<i>Rôle des agents médiateurs dans le commerce électronique</i> .....	26
2.2.7	<i>Exemples des agents médiateurs dans le commerce électronique</i> .....	31
2.3	CONCLUSION .....	33
<b>3</b>	<b>SYSTEMES ACTUELS</b> .....	<b>34</b>
3.1	<i>BUSINESS-TO-CUSTOMER</i> .....	34
3.1.1	<i>MySimon</i> .....	35
3.1.1.1	Approche de MySimon .....	35
3.1.2	<i>Agent-based Web-commerce</i> .....	36
3.1.2.1	Architecture d' "Agent-based Web-commerce" .....	37
3.2	MARCHE VIRTUEL .....	38
3.2.1	<i>Kasbah</i> .....	39
3.2.1.1	Protocoles de communication .....	39
3.2.1.2	Architecture du marché virtuel (" <i>marketplace</i> ").....	40
3.2.1.3	Architecture de l'agent .....	41
3.2.1.4	Stratégie de négociation .....	41
3.3	SALLES D'ENCHERES.....	42
3.3.1	<i>AuctionBot</i> .....	42
3.3.1.1	Types d'enchère .....	42
3.3.1.2	Protocoles de communication .....	43
3.3.1.3	Architecture de l'agent .....	44
3.4	NOTRE SYSTEME .....	44
3.5	CONCLUSION .....	45
<b>4</b>	<b>TECHNIQUES UTILISEES</b> .....	<b>46</b>
4.1	RAISONNEMENT A BASE DE CAS (CBR).....	46
4.1.1	<i>Description de CBR</i> .....	46
4.1.2	<i>Pourquoi utilise-t-on le CBR ?</i> .....	47

4.1.3	<i>Composantes d'un système à base de cas</i> .....	48
4.1.4	<i>Modèles CBR</i> .....	50
4.1.5	<i>Fonction de similarité</i> .....	52
4.1.6	<i>Distance</i> .....	52
4.1.7	<i>Adaptation</i> .....	55
4.1.8	<i>Algorithme</i> .....	55
4.2	LES AGENTS MOBILES.....	56
4.2.1	<i>Exemple en client/serveur</i> .....	57
4.2.2	<i>Exemples à base d'agents mobiles</i> .....	58
4.2.3	<i>Caractéristiques de l'agent mobile</i> .....	60
4.3	CONCLUSION.....	61
<b>5</b>	<b>PACHA</b> .....	<b>63</b>
5.1	ARCHITECTURE GLOBALE DE PACHA.....	64
5.1.1	<i>Partie-Usager</i> .....	66
5.1.2	<i>Partie-Serveur « Marché Virtuel »</i> .....	68
5.2	PROTOCOLES DE COMMUNICATION.....	72
5.2.1	<i>Processus pour conseiller l'utilisateur</i> .....	73
5.2.2	<i>Processus de recherche d'un produit</i> .....	75
5.2.3	<i>Processus de notification de l'utilisateur</i> .....	76
5.3	DEROULEMENT DE PACHA .....	78
5.3.1	<i>Côté client (choix des produits et transmission des agents)</i> .....	79
5.3.2	<i>Côté serveur (interaction des agents dans le marché virtuel)</i> .....	82
5.3.3	<i>L'interface de l'administrateur</i> .....	84
5.4	L'ENVIRONNEMENT D'EXECUTION .....	86
5.4.1	<i>Implémentation</i> .....	86
5.4.2	<i>RMI</i> .....	87
5.4.3	<i>Sécurité</i> .....	87
5.5	CONCLUSION .....	89
<b>6</b>	<b>CONCLUSION ET PERSPECTIVES</b> .....	<b>90</b>
6.1	CONCLUSION .....	90

6.2 PERSPECTIVES .....	92
<b>7 BIBLIOGRAPHIE .....</b>	<b>93</b>

## Liste des figures

FIGURE 1 : ÉVOLUTION DES SITES WEB COMMERCIAUX.....	15
FIGURE 2 : COMPOSANTES DE L'AGENT INTELLIGENT .....	22
FIGURE 3 : MODELE DE CBB. ....	26
FIGURE 4 : AGENT DE NOTIFICATION.....	27
FIGURE 5 : AGENT DE SELECTION DES PRODUITS.....	28
FIGURE 6 : AGENT DE SELECTION DES VENDEURS .....	29
FIGURE 7 : AGENTS DE NEGOCIATION.....	30
FIGURE 8 : AGENT D'ACHAT NEGOCIE AVEC PLUSIEURS AGENTS DE VENTE.....	32
FIGURE 9 : ARCHITECTURE D'”AGENT-BASED WEB COMMERCE”.....	37
FIGURE 10 : TROIS DIFFERENTES FONCTIONS DE DEGRADATIONS. “FRUGAL”, “ANXIOUS” ET “COOL-HEADED”.....	41
FIGURE 11 : MODELE DE PROCESSUS CBR.....	49
FIGURE 12 : EXEMPLE DE STRUCTURATION D'UN CAS EN CBR STRUCTUREL.....	52
FIGURE 13 : MODELE CLIENT/SERVEUR .....	57
FIGURE 14 : MODELE A AGENT MOBILE.....	59
FIGURE 15 : ARCHITECTURE DU SYSTEME PACHA.....	65
FIGURE 16 : CONSEIL ET CHOIX D'UN PRODUIT .....	73
FIGURE 17 : RECHERCHE D'UN PRODUIT .....	75
FIGURE 18 : AVERTISSEMENT DE L'USAGER .....	77
FIGURE 19 : NOTIFICATION DE L'USAGER.....	78
FIGURE 20 : L'USAGER CHOISIT LA TACHE DE SON AGENT.....	79
FIGURE 21 : L'USAGER DEFINIT SES PREFERENCES.....	80
FIGURE 22 : VERIFICATION DE LA COMMANDE.....	80
FIGURE 23 : ITEM SIMILAIRE .....	81
FIGURE 24 : LISTE DES MARCHANDS.....	81
FIGURE 25 : SCENARIO DE NEGOCIATION AVEC LES AGENTS DU MARCHE VIRTUEL.....	83

FIGURE 26 : RESULTATS OBTENUS PAR L'AGENT. ....	84
FIGURE 27 : INTERFACE PRINCIPALE DE L'ADMINISTRATEUR. ....	84
FIGURE 28 : LISTE DES EVENEMENTS.....	85
FIGURE 29 : EXEMPLE DE CONFIGURATION DU GESTIONNAIRE DE SECURITE.....	88

## Liste des tableaux

TABLEAU 1 : EXEMPLES DE QUELQUES AGENTS COMME MEDIATEURS DANS LE ECOMMERCE. .....	31
--	----

## Remerciements

Je tiens à remercier Claude Frasson, mon directeur de recherche, pour m'avoir proposé ce sujet, pour ses directives, ses conseils et son appui financier.

J'aimerais remercier mes chers amis, Wissam, Mohammed et Julie pour leurs encouragements et leur support.

J'aimerais enfin remercier mes collègues du laboratoire HERON pour leur participation et leur dialogue constructif.

# Chapitre 1

## Introduction

Le commerce électronique est un secteur producteur dont la croissance dépend étroitement de celle de l'Internet. Pendant que l'échange d'information sur l'Internet devient de plus en plus dynamique et hétérogène, les agents logiciels « software agents »<sup>1</sup> sont classifiés comme des nouveaux modules intégrés avec les services Web.

La progression rapide du taux d'informations disponibles et l'augmentation du nombre d'utilisateurs sur l'Internet (327 milliards en 2003 par Forrester research Inc [Forrester, 02]) complexifient la recherche parmi un large volume d'informations et accroît le temps de réponse aux requêtes en provenance des utilisateurs. Pour pallier la situation, les agents logiciels sont utilisés pour augmenter l'efficacité des processus et réduire la surcharge du travail associé à la recherche d'information sur le Web. Des travaux ont été effectués sur ce sujet pour abaisser l'intervention humaine dans les processus de négociation commerciale en ajoutant de l'intelligence aux systèmes implantés. Cependant, nous constatons que dans les travaux qui pourraient s'avérer intéressants le niveau d'intelligence associé à leurs systèmes est modéré. Dans ce mémoire nous présentons notre contribution dans le domaine des agents mobiles intelligents en élaborant un système *multi-agents* qui tire profit de la technologie de mobilité des agents pour assurer un haut degré d'autonomie, ce qui différencie notre système des autres systèmes classiques.

---

<sup>1</sup> La terminologie anglaise est utilisée dans le texte pour éviter l'introduction d'imprécisions syntaxiques et sémantiques.



## 1.1 Motivations

Le profil du client contient des informations personnelles, intérêts et historique d'achats. Le marchand utilise ce profil pour analyser la demande et identifier son client afin de personnaliser l'offre et les conseils de chaque client. Souvent, il y a peu ou pas de garantie d'intimité pour les informations des clients. Les marchands qui détiennent ces données, les vendent, les revendent ou se les échangent entre eux. L'individu concerné par ces données a peu de contrôle sur ce processus.

Des données statistiques tirées des études par Forrester Research, montrent une méfiance des clients en ligne vis à vis de leurs profils. 90% de clients demandent le droit de contrôler comment ces informations personnelles sont utilisées après avoir été collectées. La conservation en ligne d'informations personnelles des clients est le facteur principal conduisant à les décourager de faire des emplettes en ligne. D'autre part, selon des études faites par Industry Privacy Failures [Industry, 02], sur un échantillon de 10 000 utilisateurs d'Internet examinés, 84% d'utilisateurs ont le contenu de leurs profils vendu ou revendu à d'autres compagnies.

La majorité des systèmes actuels traitent le sujet de conseil d'une manière statique. Les conseils de ces systèmes reposent sur une base de données qui répond aux choix des clients et des marchands d'une façon unique et par défaut. Ces systèmes ne prennent pas en considération le niveau d'expertise et l'expérience du client ni les produits alternatifs qui existent chez les marchands. Évidemment, ces systèmes ont un niveau d'intelligence limité, parce qu'ils gardent une vue fixe durant la vie des transactions actuelles et futures de leurs clients. Ce point est expliqué dans l'exemple concret suivant :

Considérons un système de vente d'ordinateurs sur le Web destiné à chercher un ordinateur avec un prix qui convient au budget d'un client. Suite à un choix quelconque des composants d'un produit, les systèmes actuels réagissent avec le client d'une façon prédéfinie, c'est-à-dire, peu importe le choix de l'utilisateur ou son niveau d'expertise.

Les conseils sont toujours les mêmes. Ainsi, les conseils tirés ne se développent pas ni avec le temps ni avec l'expérience de l'utilisateur qui est supposé être en changement progressif.

Dans de tels systèmes, une autre problématique pourrait s'ajouter, car la plupart des systèmes sont adaptés aux grandes entreprises, or les « PME » ne sont pas en mesure d'avoir de grands systèmes informatiques pour présenter leurs produits. Par conséquent, ces PME sont loin de pouvoir entrer en compétition avec ces grandes entreprises.

Par ailleurs, plusieurs applications commerciales utilisent les mécanismes de base du modèle client-serveur classique. Les communications entre le client et le serveur du marchand se traduisent par un échange de requêtes et de réponses. Cet échange de messages rend nécessaire un maintien continu de la communication entre les deux entités, et c'est le client qui supervise le déroulement des opérations. De plus il doit éventuellement puiser dans ses ressources pour maintenir une communication durable entre les deux entités, ce qui entraîne une augmentation de trafic sur le réseau donc de temps qui joue un rôle très important et peut être que le client abandonne finalement l'achat.

C'est donc ces problématiques que nous tenterons d'aborder et de résoudre dans ce mémoire. L'approche que nous avons choisie est de combiner le commerce électronique avec plusieurs technologies comme l'intelligence artificielle et la mobilité des agents.

## 1.2 Objectifs généraux

La croissance exponentielle du taux d'échange d'informations hétérogène sur le Web nécessite l'intervention de médiateurs tels que les agents logiciels pour faciliter les transactions entre les vendeurs et les acheteurs. Un tel facteur a donné naissance à notre système Pacha dont les objectifs généraux sont les suivants :

- aider l'utilisateur à choisir ses produits sur l'Internet.
- établir un marché virtuel qui serve de discussion pour les agents de chaque usager.
- développer un système intelligent qui améliore au fur et à mesure les recommandations faites auprès de l'utilisateur.

En partant de ces points, nous pouvons élaborer une description plus précise de nos objectifs.

## 1.3 Objectifs spécifiques

Ce que nous visons à faire précisément avec Pacha, est récapitulé en trois points prépondérants :

- notre système doit répondre aux besoins en offrant des conseils spécifiques aux usagers. Ce mécanisme tire profit du niveau d'expertise de l'utilisateur pour lui fournir de l'aide ultérieurement.
- notre système sert à créer des agents personnalisés pour chaque usager qui tient compte de leurs particularités et de leurs préférences. Ces agents seront toutefois mobiles c'est-à-dire envoyés par les usagers pour s'exécuter à distance à une destination spécifique. Cette caractéristique donne à l'agent la possibilité d'être indépendant de son usager initial et plus apte à réaliser des tâches spécifiques demandées par son usager. De plus, cette mobilité permet à l'utilisateur de se débarrasser de l'autorité de la connexion après avoir envoyé son agent, il pourrait donc faire d'autres tâches en attendant la réponse.

- il est vital de créer un marché virtuel qui accueille des agents mobiles capables d'échanger diverses informations en provenance des usagers. Ce marché peut aider les agents délégués par les usagers à trouver les produits recherchés et aussi leur offrir des recommandations et des solutions alternatives. A noter que ces recommandations s'adaptent progressivement aux évolutions des choix de l'utilisateur.

Notre méthode pour contribuer à la solution des problèmes mentionnés repose sur le recours à des agents intelligents capables de conseiller l'utilisateur selon son niveau d'expertise et lui offrir des produits alternatifs. Ces agents offrent également une assistance en fournissant des explications sur les causes potentielles d'erreurs. De plus, notre agent est mobile et ainsi l'utilisateur ne sera pas à la merci de la connexion.

Dans ce mémoire, le Chapitre 2 fera d'abord le point sur le commerce électronique et sur le rôle que jouent les agents intelligents dans ce type de commerce. Ensuite, une étude sera présentée au Chapitre 3 sur l'état de la recherche sur les systèmes des agents intelligents dans le domaine du commerce électronique en présentant certains d'entre eux. Quelques-uns de ces systèmes traitent d'une problématique semblable à la nôtre. Il s'agira d'une étude qui fera ressortir les caractéristiques et les lacunes de ces systèmes et qui nous permettra ensuite de faire des choix éclairés quant à l'architecture et aux méthodes que nous implanterons. Le Chapitre 4 proposera l'avantage des techniques utilisées dans notre système. L'architecture et l'implémentation de notre système seront présentées dans le Chapitre 5. Le Chapitre 6 vise à déterminer si nos objectifs ont été atteints ou non. Ce dernier chapitre ouvrira finalement la porte sur les possibilités futures de notre architecture et sur les améliorations possibles qui pourraient lui être apportées.

## Chapitre 2

### 2 Le commerce électronique et les systèmes d'agents intelligents

#### 2.1 Commerce électronique

L'OCDE (organisation de coopération et de développement économique) [OCDE, 99] observe qu'il y a deux définitions possibles pour le commerce électronique. La première désigne toutes les transactions financières ou commerciales qui ont lieu sur un support électronique, celle-ci reflète une image plutôt ancienne du commerce électronique. La seconde définition décrit une vision plus récente qui ne tient compte que des transactions qui ont lieu des détaillants vers les clients sur les réseaux ouverts comme Internet. Les ateliers de l'OCDE ont récemment commencé à explorer ces phénomènes afin de les définir et d'en mesurer l'amplitude.

##### 2.1.1 Types des transactions du commerce électronique

Actuellement, plusieurs types de commerce électronique existent sur le marché. Les plus connus sont *business-to-business (B2B)*, *business-to-consumer (B2C)*, *consumer-to-business (C2B)* et *consumer-to-consumer (C2C)* [Turban,02] :

- *business-to-business (B2B)* : les participants de ce type de transactions sont soit des entreprises soit des organisations. Présentement, ce genre de commerce prédomine le marché électronique.(B2B).
- *business-to-consumer (B2C)* : ce modèle de transactions consiste en la vente au détail directement aux acteurs individuels. Un exemple de ce type est le site Amazon.com.

- *consumer-to-business (C2B)* : tout individu qui utilise l'Internet pour vendre ses produits ou services aux entreprises et/ou organisations utilise ce type de transactions. On y retrouve aussi les individus qui cherchent des vendeurs, communiquent avec eux et font des transactions en ligne au besoin.
- *consumer-to-consumer (C2C)* : cette catégorie est caractérisée par toutes les transactions inter-consommateurs pour la vente de voitures ou de propriétés résidentielles pour ne citer que quelques exemples. Le site classified2000.com utilise ce genre de transactions.

### 2.1.2 Evolution des sites Web commerciaux

L'évolution du commerce électronique (Figure 1) se fait progressivement. Tout d'abord, il commence par la mise en ligne de descriptions de produits dans un but essentiellement publicitaire. Petit à petit, la possibilité de commander en ligne est apparue. Ceci entraîne la transformation des sites Web statiques en des sites Web dynamiques. Dès lors, nous voyons émerger deux approches différentes : d'une part, certains sites se sont spécialisés dans une offre réduite de produits que l'on peut personnaliser et d'autre part, les sites catalogues sont plus étoffés par la multiplication des articles disponibles. Actuellement, on assiste à une fusion des deux courants dans laquelle l'offre ainsi que la possibilité de personnalisation sont très importantes.

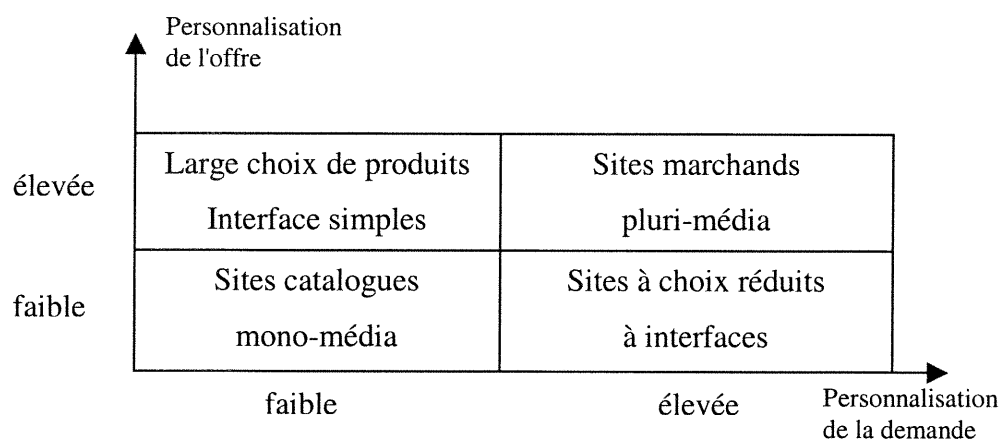


Figure 1 : Évolution des sites Web commerciaux

### 2.1.3 Les débuts du commerce en ligne

Au cours des dernières années, le nombre de sites conçus pour vendre des marchandises sur Internet n'a pas cessé de croître. Au début, il s'agissait de sites web simples qui montraient quels sont produits disponibles et qui fournissaient des informations à leur sujet, sans toutefois des transactions en ligne. Une des ressources cruciales du commerce électronique est l'adresse E-mail parce qu'elle est directement reliée au client. C'est pourquoi une méthode de publicité massive qui consiste à envoyer automatiquement des E-mails à des listes d'adresses était et reste encore très utilisée. Le problème principal avec ces E-mails publicitaires est qu'ils ne sont pas ciblés, ainsi la plupart du temps, les destinataires ne se sont pas intéressés. Pire, ils se sentent dérangés par ces méthodes brutales et protestent, à juste titre. Avec l'apparition de nouvelles techniques tel les ASP ou les JSP, ces sites Web statiques basés sur le HTML sont progressivement remplacés par les sites dynamiques, ainsi s'ouvrent la porte d'une communication active mondiale. Comme exemple de site Web commercial bien connu, citons Amazon.com qui, avec son approche "one click", permet aux clients d'acheter des produits très facilement.

Pour un bref aperçu de ces techniques dynamiques, on peut commencer par les CGI ("*Common Gateway Interface*") qui sont les précurseurs du marché sur le web dynamique. Les CGI et, plus récemment les servlets Java sont des programmes appelés par l'intermédiaire du protocole HTTP sur un serveur web. Ils envoient les résultats à l'utilisateur sous forme des pages HTML. Les ASP ("*Active Server Pages*"), les JSP ("*Java Server Pages*") et les PHP sont des techniques plus accessibles et plus flexibles : il s'agit de pages HTML classiques dans lesquelles des scripts ont été ajoutés pour être interprétés au niveau serveur. D'autres techniques de scripts et d'applets sont largement répandues, mais comme elles fonctionnent sur le poste du client, elles sont moins adaptées pour des questions de rapidité ou de sécurité.

### **2.1.4 Centres commerciaux virtuels**

Les portails commerciaux apparaissent ensuite sur le Web pour remplacer les catalogues en ligne. Il s'agit en quelque sorte de grands centres commerciaux ayant des galeries virtuelles qui hébergent des boutiques multiples et variées. Un portail peut être vu comme un serveur Web qui fournit un certain nombre de services aux magasins hébergés:

- la gestion du caddie virtuel dans lequel le client met toutes les marchandises qu'il veut acheter ;
- la gestion des commandes avec des formulaires et des outils de transactions sécurisées ;
- des outils de gestion pour la mise à jour, la livraison et le suivi ;
- le référencement dans les moteurs de recherche et la publicité du magasin ;
- d'autres services comme le support technique.

Aujourd'hui la majeure partie des transactions porte sur les voyages, les locations de voitures, les vêtements, les finances, les logiciels, la culture et les services Internet. Il existe maintenant d'autres genres de portails qui regroupent et synthétisent l'information : portails thématiques, revues de presse, boîte à outils, etc.



### 2.1.5 Les besoins de médiation

Les objectifs du commerce électronique sont de fournir les meilleurs produits pour satisfaire les clients et de les fidéliser afin de maximiser les gains de l'entreprise. Les magasins virtuels offrent plusieurs avantages tels l'ouverture 24 heures sur 24 et l'accès à une clientèle mondiale. Ils permettent également de réduire les coûts et limitent les problèmes liés au stockage. Cependant, la notion de proximité disparaît, cela signifie qu'il faut faire face à une concurrence mondiale. Un élément de solution est donné par la règle des "7 I" du marketing « one-to-one » énumérés par J.-Y. Granger<sup>2</sup> : "interactivité, immédiateté, information, intégration, individualisation, interconnexion et itération". Nous allons maintenant préciser les points qui nous paraissent essentiels pour offrir un bon service sur les centres commerciaux virtuels.

### 2.1.6 Connaître le client pour établir une interaction privilégiée

F. Lorentz<sup>2</sup>, responsable de la mission *Commerce Électronique* auprès du gouvernement français, explique qu'il faut "créer un climat de confiance" en entretenant une "relation privilégiée" avec les clients. En effet, comme ces derniers sont directement en ligne et que l'on souhaite les fidéliser, il est important d'établir un contact interactif avec eux. Pour cela, il suffit d'apprendre à les connaître en observant leur comportement afin d'adapter les interactions au fur et à mesure.

### 2.1.7 Permettre de personnaliser l'offre

Le magasin virtuel doit développer la fidélité des clients en leur offrant des services et des produits de grande qualité qu'il est possible de personnaliser. Comme le souligne

---

<sup>2</sup> Table ronde *Commerce et Distribution* du Congrès Net 2000.

P.Guillanton, PDG de Yahoo France, la personnalisation des produits apporte une valeur ajoutée et favorise une relation de qualité avec le client. Notons qu'un nombre croissant de sociétés permettent de concevoir des produits sur mesure. À titre d'exemple, citons : Dell, Général Motors, Levis, Mattel, Renault, etc. En dehors des produits mêmes, il est important de personnaliser les bannières publicitaires qui utilisent des arguments spécifiques selon les désirs du client.

### **2.1.8 Adopter une politique de transparence**

Le magasin virtuel doit également faire preuve de transparence afin de gagner la fidélité particulièrement lorsque les clients sont hésitants face aux nouvelles technologies. Naturellement, de grandes améliorations permettent maintenant de garantir la confidentialité et la sécurité lors des transactions, mais il reste encore beaucoup à faire. La politique commerciale de la société doit également être clairement affichée de sorte que le client lui accorde facilement sa confiance.

### **2.1.9 Fournir une assistance aux clients**

Avec la diversification de l'offre et la multiplication des produits disponibles, le client se retrouve souvent envahi en masse d'informations. Il est rarement en mesure de fournir exactement une description exacte de ses besoins. C'est pourquoi, bénéficier d'une assistance omniprésente, dès l'arrivée sur le site, est fort appréciable. Cette assistance peut prendre, par exemple, la forme d'un guide virtuel qui accueille et conseille le client.

### **2.1.10 Offrir des services complémentaires**

Une fois qu'il a choisi un produit, le client est plus sensible à la proposition de services ou d'accessoires complémentaires. Il s'agit donc d'automatiser un élargissement par une

recherche dans le voisinage du produit. Cet aspect de voisinage est utilisé par exemple dans les magasins Fnac avec la mention : "Si vous avez aimé *A*, vous aimerez *B*".

Tous ces besoins mettent de l'avant la nécessité d'une médiation entre les clients et les boutiques virtuelles. Qu'il s'agisse de guider le client, de l'aider à formuler une requête ou de lui proposer des offres complémentaires, il ressort que la modélisation des utilisateurs est un point clé pour pouvoir fournir adéquatement tous ces services adaptatifs.

M. Perkowitz et O. Etzioni ont lancé un défi [Perkowitz, 97] sur les sites Web adaptatifs à la communauté de l'Intelligence Artificielle. Ce défi consiste à personnaliser dynamiquement les sites Web et à les optimiser à la volée. Concernés par ce défi, nous proposons de réaliser cette médiation "*Business-to-Customer*" au moyen d'agents intelligents.

## 2.2 Les Systèmes d'agents intelligents

### 2.2.1 Pourquoi utilise t-on des agents ?

Les solutions aux questions vues dans la section précédente relèvent de plusieurs domaines, donc il est plus réaliste de répartir les fonctions sur plusieurs agents. Par exemple, afin de vendre un produit, on doit tenir compte de la connaissance du domaine auquel il appartient, de la connaissance et des stratégies de la société commerciale qui le vend, et de bien d'autres facteurs. Nous supposons ici qu'un agent est une entité logicielle autonome qui perçoit partiellement son environnement au travers de capteurs, raisonne à son sujet et produit des changements dessus au moyen d'actions. L'environnement de l'agent contiendra donc les clients eux-mêmes, mais aussi un grand nombre d'objets allant d'Internet aux bases de données.

En utilisant les critères de Wooldridge [Weiss, 99], l'environnement auquel nous avons affaire, est caractérisé par les propriétés suivantes :

- il est virtuel parce que les agents le perçoivent avec des capteurs logiciels et qu'ils agissent en utilisant des programmes et autres modules ;
- il est impossible que les agents en aient une vue complète;
- il n'est pas déterministe : les agents agissent mais il n'y a, a priori, aucune garantie de succès;
- il ne peut pas être considéré comme épisodique : le passé doit être pris en ligne de compte car il a une grande influence sur le présent, particulièrement dans des interactions commerciales ;
- il est fortement dynamique : les marchés, les bases de données et les sites Web ne cessent d'évoluer en reflétant l'activité mondiale humaine ;
- il doit être considéré comme continu car l'ensemble d'actions potentielles est infini.

## 2.2.2 Les composantes de l'agent intelligent

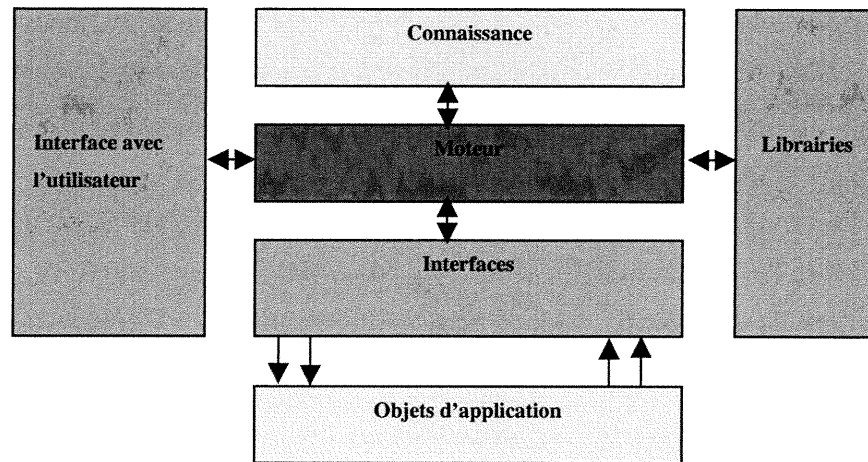


Figure 2 : Composantes de l'agent Intelligent

Un agent intelligent contient un ou plusieurs des éléments suivants (Figure 2):

- Le **moteur** : c'est le cerveau de l'agent qui permet de tenir des raisonnements plus ou moins complexes;
- La **connaissance** : ce que l'agent sait, croit et pense. La connaissance peut être basée sur des applications existantes, par exemple une base de données ou un agent de courrier. La connaissance est stockée dans les librairies;
- L'**interface avec les autres applications** : par exemple avec le courrier électronique ou avec le Web. C'est l'interface qui détermine le champ d'application d'un agent;
- L'**interface avec l'utilisateur** : permet la communication entre l'agent et l'utilisateur. L'utilisateur donne ses instructions à l'agent, ce dernier renvoie les résultats ou donne des informations. Cette interface peut être graphique usuelle, mais l'avenir privilégiera le langage naturel parlé.

### 2.2.3 Attributs de l'agent intelligent

Les quatre attributs principaux d'un agent sont l'intelligence, l'autonomie, les interactions (capacité de communication et coopération) et la mobilité.

- **L'intelligence**

L'intelligence est la faculté de raisonnement et d'apprentissage. C'est la capacité de l'agent à accepter les demandes de l'utilisateur et de mener à bien la tâche qui lui est confiée. Au minimum, l'utilisateur doit pouvoir spécifier ses préférences, par exemple sous forme de règles, avec un mécanisme de raisonnement qui agit par rapport à ces préférences. À un degré d'intelligence plus élevé, l'agent devrait comprendre ce que l'utilisateur veut, et planifier les moyens à mettre en œuvre pour atteindre ce but.

- **L'autonomie**

L'agent doit pouvoir prendre des initiatives et agir sans l'intervention de l'utilisateur final.

- **Interactivité**

L'agent peut interagir avec quatre composantes de l'environnement à savoir : le matériel (détection d'un nouveau composant ou d'une panne matérielle), les humains, les autres agents (coopération et échange d'information) et enfin les logiciels qui ne sont pas des agents.

Voici aussi une liste d'attributs supplémentaires que peut posséder un agent :

- **Réactivité**

L'agent doit pouvoir faire face aux modifications de l'environnement, que ce soit la modification des objectifs de l'utilisateur ou des ressources disponibles.

- **Délégation**

L'agent est mandaté par l'utilisateur pour effectuer une tâche. Dans le cas du commerce électronique, un degré de délégation faible impliquerait que l'agent aille chercher les opportunités intéressantes et les propose à l'utilisateur. À un degré plus élevé, l'agent effectue lui-même l'achat. Mais dans ce cas, il y a un risque que l'agent effectue une mauvaise opération de là, la nécessité d'une relation de confiance solide entre l'utilisateur et l'agent.

- **Personnalisation**

L'utilisateur détermine la façon dont l'agent interagit. Dans la majorité des cas, l'agent s'adapte à l'utilisateur.

- **Prévisibilité**

Il est nécessaire que l'utilisateur sache à quel résultat s'attendre de la part de l'agent.

## 2.2.4 Quelques notions d'agent

On trouve ces notions d'agent dans divers domaines de l'informatique :

- Les *systèmes multi-agents*, provenant des domaines de la robotique, système répartis et de l'intelligence artificielle qui proposent de structurer une application en différentes fonctions représentées chacune par un agent. Les différents agents coopèrent pour la résolution des problèmes de l'application [Shoham, 90] et [Sierra, 99].

- Les *agents d'interface*, provenant des domaines de l'interface homme-machine et de l'intelligence artificielle, essaient de simplifier la vie de l'utilisateur en automatisant des tâches réalisées couramment par l'observation des comportements répétitifs ou en surveillant des ressources comme le courrier électronique [Coen, 94].
- Les *agents mobiles*, provenant du domaine des réseaux dont l'objectif est de répartir de tâches sur un ensemble de machines interconnectées pour le compte d'un utilisateur ou d'une application.

La notion d'agent mobile est assez récente dans le domaine des réseaux et des systèmes répartis et engendre une activité de recherche très importante. Le système *Telescript* [J. E., 94] de *General Magic*, qui est à l'origine du premier système d'agents mobiles, a déclenché l'intérêt des grands opérateurs de télécommunication pour cette nouvelle forme de structuration des communications. Beaucoup de projets de recherche sont en cours pour approfondir ce nouveau modèle de communication et aussi étudier l'impact de cette technologie sur l'efficacité des systèmes repartis.

### **2.2.5 Les agents intelligents dans le commerce électronique**

Dans le domaine d'échange commercial, plusieurs tâches peuvent être associées aux agents intelligents telles que la négociation (traitement des enchères) et aussi la notification des clients sur l'accusation des nouveaux produits. De plus, les agents intelligents peuvent jouer un rôle de faciliter d'une part la sélection des produits selon le goût des consommateurs et d'autre part de permettre aux vendeurs de mieux connaître et comprendre la commande. Ces tâches seront traitées en détail dans la section suivante.

Il existe deux types d'agents en commerce électronique:



➤ **Les agents contrôlés par le consommateur**

On les appelle « shopping agents » ou « shopbots ». Ces agents aident à :

- Rechercher un produit précis;
- Comparer les prix ou les services;
- Acheter ou recommander des produits.

➤ **Les agents d'analyse de la demande des consommateurs**

Ces agents aident à :

- Analyser la demande;
- Connaître les consommateurs;
- Gérer des profils clients;
- Personnaliser l'offre pour chaque client.

## 2.2.6 Rôle des agents médiateurs dans le commerce électronique

Le modèle CBB (Consumer Buying Behavior) [Guttman, 98b], est un outil puissant qui aide à comprendre le rôle d'un agent comme médiateur dans le commerce électronique.

Ce modèle propose six étapes fondamentales du commerce électronique (Figure 3) :

Étapes de CBB (Agent Médiateur)	
• Identification du besoin	• Négociation
• Sélection du produit	• Achat et Livraison
• Sélection du vendeur	• Service après vente et évaluation

Figure 3 : Modèle de CBB.

- **Identification du besoin**

Cette étape consiste à aviser le client des produits imprévisibles susceptibles de l'intéresser. Ce consommateur sera informé à propos des informations sur les nouveaux produits. Les agents de notification (Figure 4) jouent ce rôle. Ces agents avertissent les clients de la venue de nouveaux produits en comparant les catalogues de produits du marchand avec le profil et les préférences de chaque client. Ses avertissements sont généralement envoyés par courrier électronique.

Parmi ces agents on cite : **Amazon.com** qui annonce les nouveautés sur les livres disponibles.

- **Fastparts.com** qui annonce les rabais de produits spécifiques.

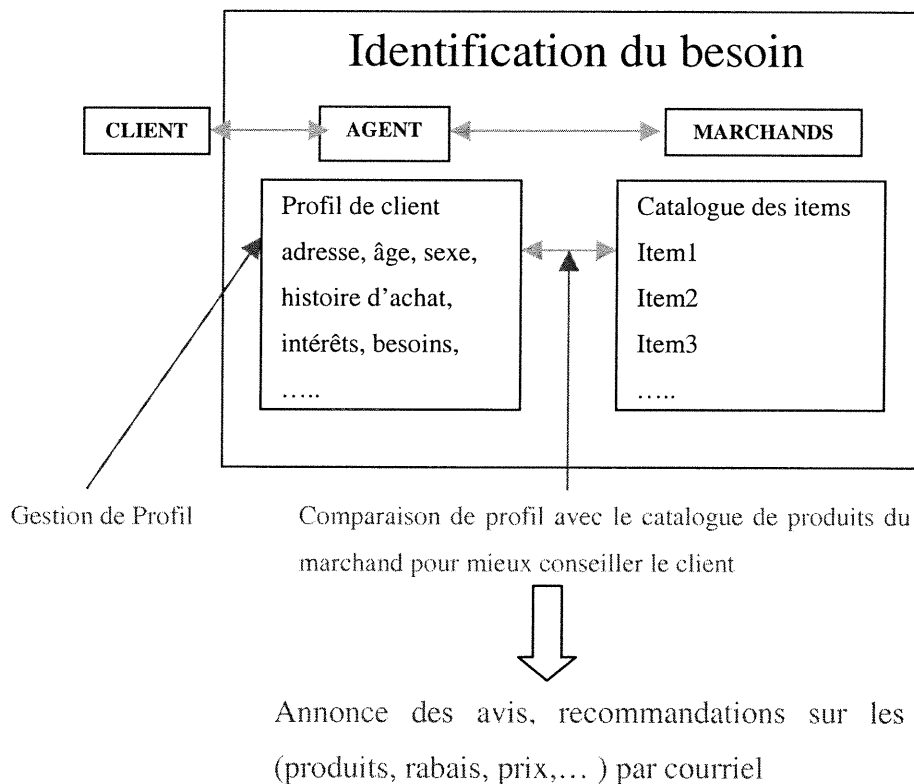


Figure 4 : Agent de notification

- **Sélection du produit**

Cette étape consiste en la recherche d'informations pour aider à déterminer les achats. En plus le consommateur peut évaluer des produits alternatifs offerts par les marchands. Cette évaluation est en général, basée sur des critères et des contraintes fournis par le consommateur. Il existe des agents de recommandation ou de conseils (Figure 5) comme Amazon [Amazon, 02] et eBay [eBay, 02], dont le rôle est d'offrir des produits aux consommateurs en se basant sur le profil du client et de quelques technologies intelligentes comme le “*data mining*” [Rakesh, 93]. Ces technologies aident les agents à indiquer et prévoir les produits à présenter aux clients.

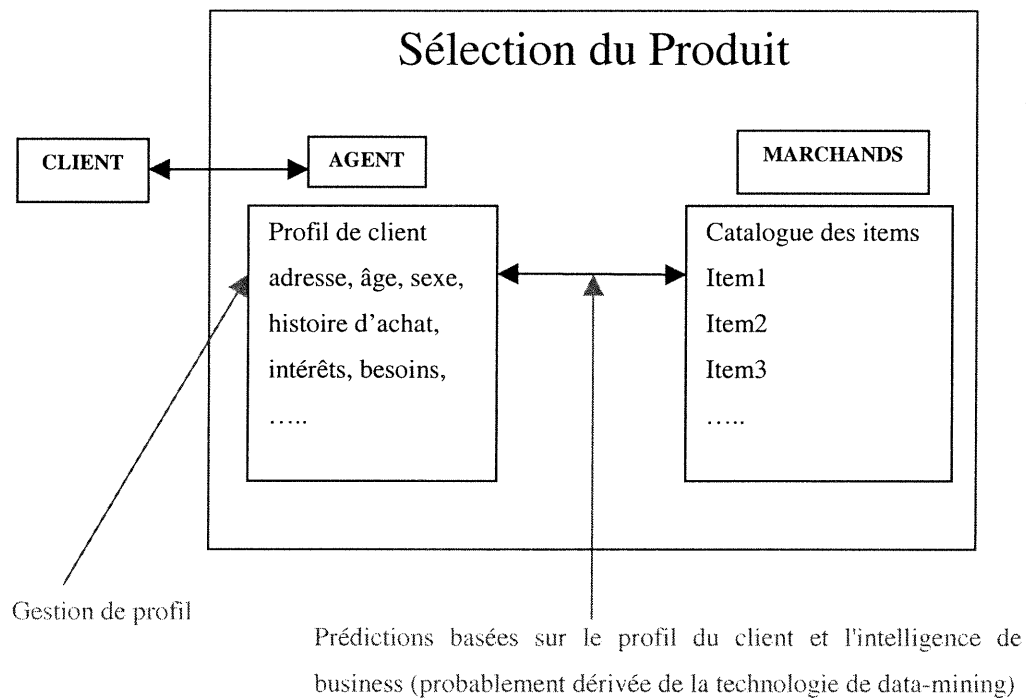
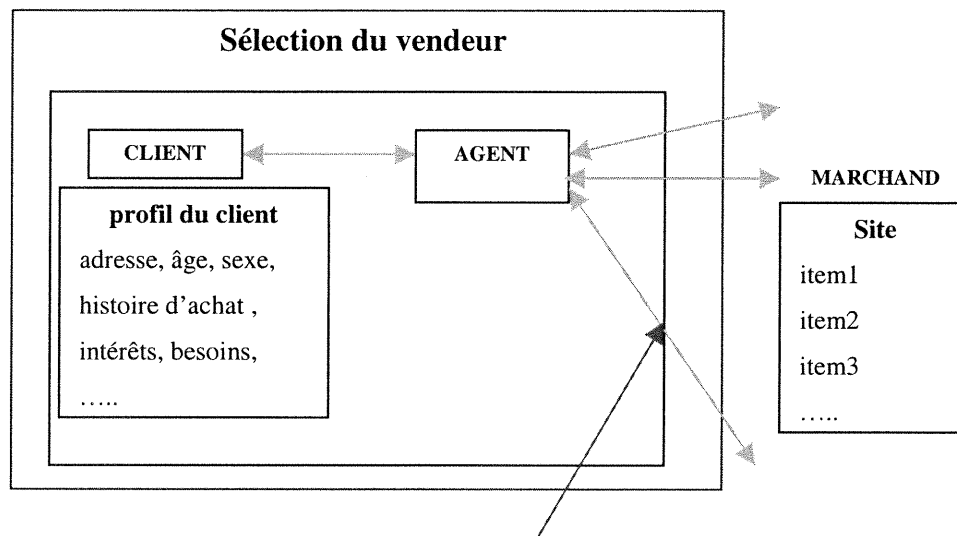


Figure 5 : Agent de sélection des produits

- **Sélection du vendeur**

Cette étape combine les résultats de l'étape précédente (recommandation de produits) et ajoute les services offerts de chaque marchand pour aider les clients à choisir la source des achats qui lui convient. Ceci inclut l'évaluation des marchands alternatifs basés sur des critères fournis par le client (prix, garantie, temps de livraison,...) (Figure 6). Cette étape est dédiée à l'agent de recommandation de marchands comme *Priceline* [Priceline, 02].



Extrait d'informations sur les produits demandés y compris les services offerts de chaque marchand

Figure 6 : Agent de sélection des vendeurs

- **Négociation**

Cette étape est utilisée pour arranger un achat et pour déterminer les termes d'une transaction comme la durée, le prix de produit. Ces termes changent selon le marché. Par exemple, le prix d'un produit est fixé dans les marchés traditionnels, en revanche, dans des marchés virtuels, le prix est à la merci de courtage (solde et compétitions). Cette étape est dédiée à l'agent de négociations comme Tete-a-Tete [Guttman, 98a] et CASBA [Michael, 99] (Figure 7).

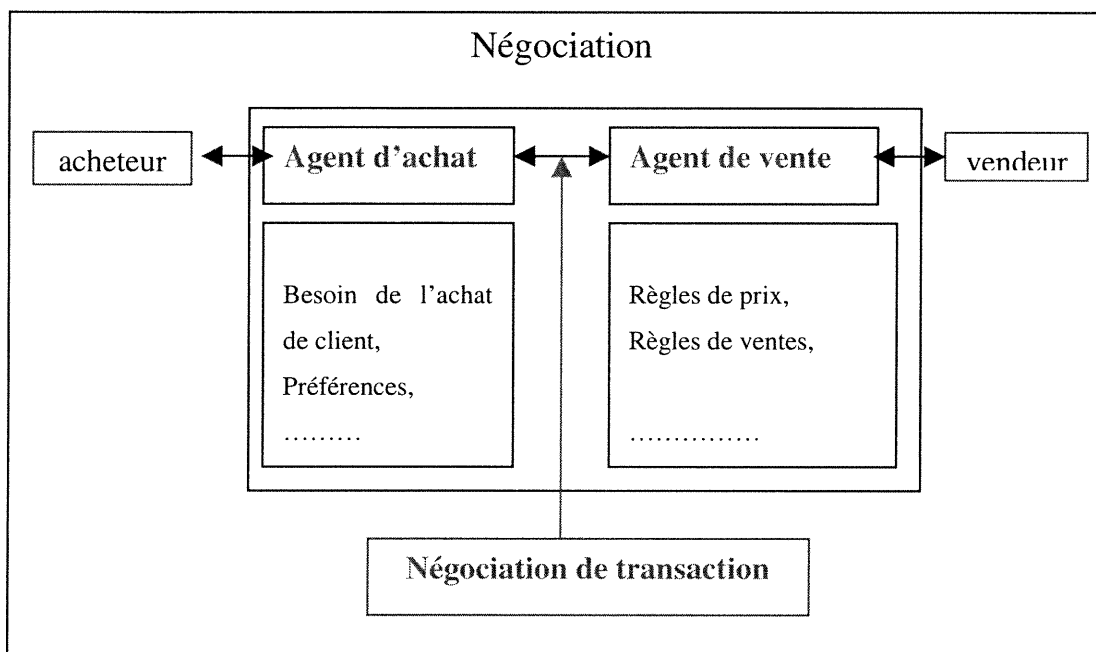


Figure 7 : Agents de Négociation.

- **Achat et Livraison**

Cette étape signale la fin de la négociation, ou bien elle se produit plus tard. En général, le mode de paiement et/ou l'option de livraison influencent la recommandation des produits et des marchands.

- **Service après vente et évaluation**

Le service de l'évaluation du produit se fait après l'étape de l'achat. Celle-ci implique le service de produit, le service de clientèle et l'évaluation de la satisfaction des consommateurs à l'égard de ses achats.

## 2.2.7 Exemples des agents médiateurs dans le commerce électronique

Les agents intelligents sont déjà utilisés par les acheteurs et les vendeurs pour réaliser certaines étapes du processus d'achat sur internet. Il existe actuellement plusieurs agents qui peuvent assister le consommateur dans au moins trois étapes (Tableau 1).

	PersonaLogic	Jango	Tete-a-Tete
1. Identification du besoin	Un peu des agents qui utilisent des outils (event-alerting ) pour avertir et informer ses clients à propos des produits spécifiques		
2. Recommandation de Produits	x	X	X
3. Recommandation de marchandises		X	X
4. Négociation			X
5. Achat et Livraison			
6. Service et évaluation			

Tableau 1 : exemples de quelques agents comme médiateurs dans le Ecommerce.

- **PersonaLogic**

C'est un outil qui permet au client de sélectionner les produits selon ses besoins, en leur proposant de multiples caractéristiques sur les produits. Lorsque les spécifications du

produit sont indiquées par le client, l'agent PersonaLogic présente une liste de produits qui satisfait approximativement les caractéristiques indiquées par le client qui peut choisir alors le produit qui lui convient le mieux [PersonaLogic, 02].

- **Jango**

Cet agent fonctionne d'une manière semblable à PersonaLogic, mais il présente en plus les services de chaque marchand comme la date de livraison, la garantie et d'autres informations [Jango, 02]. Ces services permettent au client de choisir le produit et le marchand pour réaliser l'achat.

- **Tete-a-Tete**

L'agent Tete-a-Tete [Guttman, 98a], respecte les trois étapes essentielles du modèle CBB (Consumer Buying Behavior) : recommandation de produit, recommandation de marchands et négociation. À la différence d'autres Agents, Tete-a-Tete sert les consommateurs et les marchands. Il aide le client à chercher et trouver le produit et en même temps à bien évaluer et connaître les services offerts par les marchands (Figure 8).

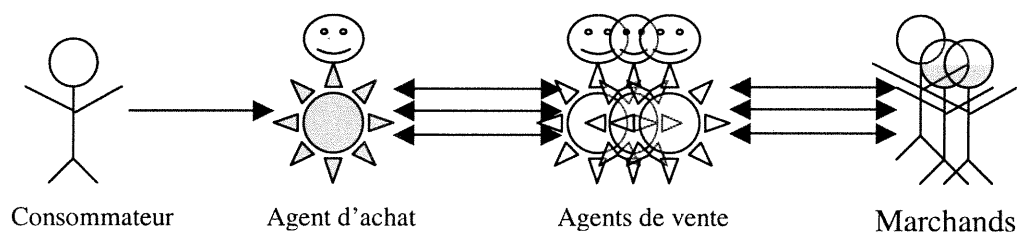


Figure 8 : Agent d'achat négocie avec plusieurs agents de vente.

L'agent Tete-a-Tete engage les agents d'achat et ceux de vente dans une négociation concernant les offres et les services des marchands ainsi que les besoins du client. Dans une négociation, un marchand ne s'évalue pas seulement par ses prix mais aussi par les services offerts aux clients comme la date de livraison et la garantie d'un produit. Par

exemple, un client peut considérer le temps de livraison, le service de maintenance et la garantie comme conditions d'achat d'un item. Les agents de vente proposent des offres à l'agent d'achat. Chaque offre contient les caractéristiques du produit, le prix, et autres services du marchand. L'agent d'achat évalue et sélectionne l'offre qui lui convient en se basant sur les préférences du client. Si l'agent d'achat n'est pas satisfait des offres qui lui sont proposés, alors il peut les critiquer. Dans ce cas, l'agent d'achat envoie ses nouvelles préférences aux agents de vente pour qu'ils puissent lui proposer des offres plus adéquates..

## 2.3 Conclusion

Le commerce électronique génère une nouvelle activité appelée l'intermédiation ou la médiation électronique. Cette activité vise à faciliter les échanges électroniques entre le vendeur et l'acheteur. La croissance exponentielle d'une masse d'information déjà considérable nécessite des intermédiaires électroniques incorporant des agents intelligents capables d'aider le vendeur ou l'acheteur à faire la recherche, la localisation, la commande et la livraison d'un produit ou d'une information. Dans ce chapitre nous avons présenté le rôle principal des agents intelligents comme des intermédiaires dans le commerce. Le prochain chapitre parlera de l'état de la recherche des systèmes des agents intelligents dans le domaine du commerce électronique en présentant plusieurs de ceux-ci, dont quelques-uns traitent d'une problématique semblable à la notre.



## *Chapitre 3*

### **3 Systèmes actuels**

Le chapitre présente quelques systèmes d'agents du commerce électronique : *Mysimon*, *Agent-based Web-commerce*, *Kasbah* et *AuctionBot*. Ces derniers ont grandement inspiré la conception du système que nous proposons. Cette discussion nous amènera ensuite vers un survol des caractéristiques que possédera notre système.

Concernant le commerce électronique actuel sur Internet, les agents à l'étude se subdivisent en trois catégories : "*Business-to-Customer*", les marchés virtuels et les salles d'enchères.

#### **3.1 *Business-to-Customer***

Le B2C pour "*Business-to-Consumers*" s'occupe de la vente au grand public de biens et de services qui peuvent être livrés en ligne ou non. Amazon ou Dell constituent les archétypes de ce genre de service sur le *Web*. Les services B2C sont rendus le plus souvent par des intermédiaires commerciaux, qu'ils soient de nouveaux entrants comme MySimon [MySimon, 02] ou des intermédiaires traditionnels, comme Amazon.

Dans cette catégorie, nous trouvons des agents tels MySimon, Jango [Jango, 02] et PersonnaLogic qui sont spécialisés dans la recherche de produits pour les clients. Typiquement, on leur donne une description des articles recherchés (titres, auteurs et autres caractéristiques...) puis ils fonctionnent sur l'Internet comme des moteurs de recherche.

### 3.1.1 MySimon

MySimon [MySimon, 02], est un moteur de comparaison de prix indépendant lancé le 1<sup>er</sup> décembre 1999, entre autre il trouve et compare des produits (livres, disques, logiciels, jeux, hi-fi, télévision, jeux vidéo, matériel informatique...) et des prix.

Pour comparer les prix, les délais et les coûts de livraison afin d'optimiser son achat sur Internet, MySimon parcourt des dizaines de sites marchands en même temps à la recherche de ce que l'internaute souhaite acheter. Ensuite, il rassemble, organise et présente l'information pour offre la meilleure décision d'achat. Il peut ainsi, en temps réel, rechercher et choisir le marchand qui répond le mieux à son critère de décision : le moins cher ou, par exemple, celui qui peut livrer le plus rapidement.

Concrètement, si nous cherchons un Palm Pilot, MySimon va proposer rapidement le Palm de notre choix dans les différents magasins sur le Web. MySimon peut éventuellement préciser la disponibilité et le prix du Palm Pilot recherché. Il fournit également le lien pour se connecter au service marchand pour commander le Palm en ligne ou les coordonnées du magasin ou du catalogue qui le propose à un prix intéressant.

#### 3.1.1.1 Approche de MySimon

Le moteur de comparaison de prix de MySimon utilise la technologie du “*Virtual Learning Agent*” (VLA) développé par MySimon (leader mondial de la comparaison de prix). Cette technologie lui permet de comparer les prix et la disponibilité du produit en

temps réel. Il recherche les prix de plusieurs dizaines de millions de produits chez plus de deux mille boutiques en ligne.

Les ingénieurs de MySimon ont commencé à développer la technologie “*Virtual Learning Agent*” (VLA) en 1998. Celle-ci peut créer des robots virtuels (ou des agents) qui agissent comme de véritables internautes surfant sur le Web. Les robots d'achats sont alors programmés pour pouvoir collecter, en temps réel, des informations de prix, de disponibilité et de délais de livraison chez n'importe quel site marchand à la manière d'un enquêteur relevant les prix manuellement dans un grand magasin ou un hypermarché.

Ainsi, grâce à la technologie du “*Virtual Learning Agent*” (VLA), dès qu'une recherche est lancée sur MySimon.com, un "robot" par marchand s'active pour le produit demandé. C'est ainsi que plusieurs "robots" sont sur le qui-vive, pour satisfaire l'internaute immédiatement et lui offrir une fiabilité totale sur les informations recueillies.

### 3.1.2 Agent-based Web-commerce

Le projet “Agent-based Web-commerce”, à commencé en avril 1997 à l'école des Sciences Appliquées, Université Technologique de Nanyang, Singapour [Gunghao, 98]. Son objectif principal est de concevoir et de mettre en application une infrastructure de logiciel pour un grand système réparti qui s'appelle “Agent-based Web-commerce”. Ce projet vise à construire les prototypes génériques pour trois types d'agents à savoir: agents d'achat (“*buyer agent*”), agents de vente (“*seller agent*”) et agents de répertoire (“*Directory agent*”) :

- du côté vendeur, ce projet fournit une gamme de logiciels qui intègrent le Web, la gestion de base de données et la technologie d'agent. Ces logiciels permettent à l'entreprise de créer et de maintenir ses pages Web et ses « *agents de ventes* ». Ces derniers fournissent des services aux humains et en plus ils peuvent servir d'autres agents.

- ce projet fournit également un outil pour créer des « *agents d'achat* » pour des individus ou des organismes.
- “*Directory Agent*” : *cet agent* garde des informations sur les autres agents (Figure 9). Il fonctionne comme les moteurs de recherche conventionnels en fournissant des liens aux sites Web basés sur des mots-clés. Quand un agent veut localiser d'autres agents possédant certaines capacités, il cherche d'abord “*Directory Agent*”. Étant donné que les agents de répertoire ont des URLs uniques qui sont bien connus.

Le projet fournit aussi des protocoles d'interaction entre les agents d'achat et de vente de sorte que tous les agents créés puissent communiquer entre eux.

### 3.1.2.1 Architecture d' “Agent-based Web-commerce”

La Figure 9 ci-dessous montre les composantes principales du système “*Agent-based Web-commerce*”. Une transaction commence par un agent d'achat. L'utilisateur que l'agent représente indique le ou les item(s) qu'il veut acheter. L' « *agent d'achat* » communique avec un « *agent de répertoire* » pour l'adresse URL des « *agents de vente* » qui offrent l'item(s).

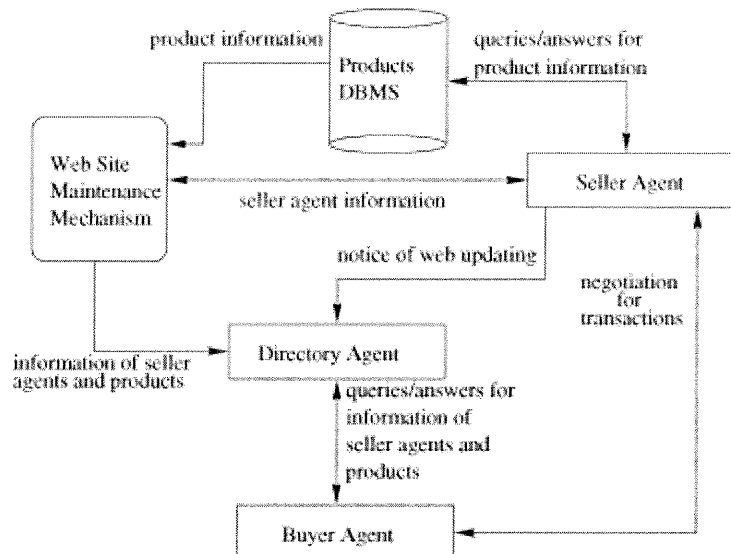


Figure 9 : Architecture d' “Agent-based Web commerce”.

(extrait de [Gunghao, 98]).

L' « *agent d'achat* » demande également les attributs (et certaines de leurs valeurs), que les « *agents de vente* » utilisent pour décrire le produit à acheter. Ce dernier emploie cette information pour aider un client à raffiner sa requête d'achats. Cette information sera par la suite stockée pour de futurs usages, de sorte que quand le même item est demandé encore, des attributs puissent être immédiatement présentés à l'utilisateur. Comme les différents vendeurs ont un vocabulaire différent pour les attributs d'un même item, un « *agent de répertoire* » les intègre dans un seul vocabulaire global et commun.

L' « *agent de répertoire* » cherche des informations sur des agents de vente en utilisant des robots de Web, tout comme les moteurs de recherche conventionnels. Il passe par les pages de Web de divers organismes pour chercher des informations sur des agents de vente. Dans la Figure 9, le “*Web Site Maintenance Mechanism*” met à jour les informations des « *agents de vente* » et de produits. Dès qu'une information sur un site Web est changée, un agent de vente envoie l'information mise à jour à l'agent de répertoire.

Quand des agents de vente pour un produit sont trouvés, l'agent d'achat entre directement en contact avec eux. Ensuite il compare et négocie à propos les prix, et fait l'achat si le prix lui convient.

### **3.2 Marché virtuel**

Parmi les autres applications du commerce électronique, les marchés virtuels se développent de façon importante. Il s'agit de points de réunion entre agents négociateurs. Kasbah [Anthony, 96] est un bon exemple de marché virtuel flexible. Les vendeurs proposent des produits aux acheteurs qui vérifient si la description correspond à leurs buts. S'il y a appariement, ils commencent à négocier les prix en utilisant les règles données par leurs propriétaires respectifs. Des systèmes d'intermédiaires [Collins, 98] et des agents de médiation [Kuokka, 98] ont également été développés pour amorcer ou

arbitrer le processus de négociation. Ces intermédiaires reçoivent les requêtes d'un agent demandeur au moyen de descriptions de produits et des offres, de la part des fournisseurs. Après avoir trouvé la meilleure correspondance, l'agent médiateur peut se contenter de lancer le processus de communication, rester en tant qu'arbitre ou encore servir comme intermédiaire de confiance entre les protagonistes.

### 3.2.1 Kasbah

Pour avoir une idée sur le fonctionnement de Kasbah, nous allons parler en bref des protocoles de communication, d'architecture du marché virtuel, d'architecture d'agent et des stratégies de négociation qui sont fournis par Kasbah.

#### 3.2.1.1 Protocoles de communication

Kasbah est composé de deux modules principaux : agents et "*marketplace*". Ces deux modules fournissent des méthodes pour les appeler. Pour la communication "agent-agent" et "marketplace-agent", les méthodes de l'agent seront invoquées. Et pour la communication "agent-marketplace", les méthodes du "marketplace" seront appelées. A noter que l'utilisateur peut aussi appeler des méthodes du "marketplace".

Les agents communiquent entre eux en utilisant les méthodes suivantes :

- "*Accept-offer?*", demande à l'agent s'il accepte une offre fournie par un autre agent. Elle retourne vrai si l'agent accepte l'offre, et faux si non.
- "*What-is-price?*", demande le prix offert d'un autre agent. Cette méthode retourne le prix.
- "*What-is-item?*", demande à l'autre agent qu'est ce qu'il essaye de vendre/acheter. Cette méthode retourne une description de l'item.

Le “*marketplace*” peut communiquer avec les agents via les méthodes suivantes :

- “*add-potential-customer*” alerte l’agent de vente, lorsqu’il y a un agent d’achat qui veut acheter l’item offert.
- “*add-potential-seller*” alerte l’agent d’achat, lorsqu’il y a un agent de vente qui veut vendre l’item demandé.
- “*remove-potential-seller*” alerte l’agent d’achat, quand l’agent de vente n’existe plus.
- “*remove-potential-buyer*”, qui alerte l’agent de vente, quand l’agent d’achat n’existe plus.

Le “*marketplace*” fournit des méthodes qui sont utilisées par les agents pour communiquer avec le “*marketplace*” :

- “*agent-terminated*” informe le “*marketplace*” que l’agent n’existe plus.
- “*deal-made*”, informe le “*marketplace*” que les agents ont accepté la transaction d’un article pour un prix donné.

L’utilisateur communique avec le “*marketplace*” en utilisant ces méthodes :

- “*add-sell-agent*”, pour ajouter un agent de vente au “*marketplace*”.
- “*add-buy-agent*”, pour ajouter un agent d’achat au “*marketplace*”.

### 3.2.1.2 Architecture du marché virtuel (“*marketplace*”)

Le “*marketplace*” communique avec les agents comme cela est expliqué dans la section précédente. L’utilisateur communique avec une interface graphique. Il peut spécifier quel genre d’agent il veut utiliser. Alors le marketplace ajoute cet agent et envoie un message aux autres agents qui veulent vendre ou acheter le même article. Autrement le “*marketplace*” travaille comme un médiateur pour ces agents jusqu’à ce qu’ils finissent ou aboutissent à une affaire.

### 3.2.1.3 Architecture de l'agent

L'agent peut négocier avec les autres agents. Il peut aussi prendre une décision à propos des offres reçues ou bien celles qu'il doit présenter. Pour prendre ces décisions, Kasbah a implémenté trois différentes stratégies de négociation, qui sont basés sur trois fonctions de prix. Ces derniers sont expliqués dans la prochaine section.

Quand le tour d'un agent vient, il peut communiquer avec un autre agent. Ensuite, une fois terminé, il passe le tour à l'autre agent. Prenons l'exemple d'un agent de vente. Cet agent suit cette procédure :

- détermine le prix courant demandé. L'agent décrémente le prix demandé en respectant la fonction du prix déjà spécifiée ;
- décide avec quel agent veut communiquer pour qu'il accepte son offre ;
- communique avec d'autre agent, en lui proposant son offre.

### 3.2.1.4 Stratégie de négociation

Kasbah offre une stratégie de négociation simple et facile. Chaque utilisateur choisit l'intervalle de temps pour vendre/acheter ainsi que la fonction de dégradation de prix.

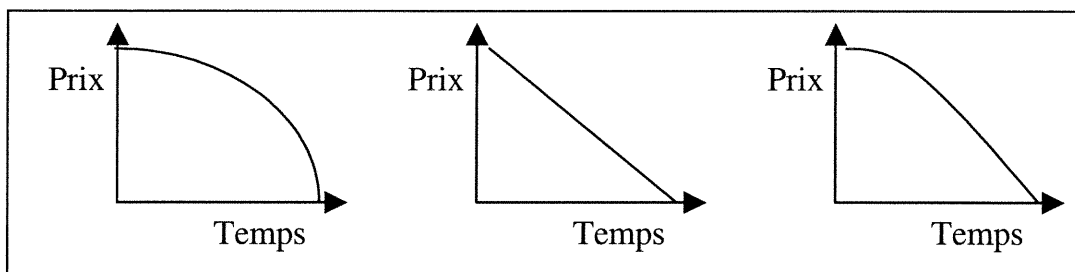


Figure 10 : Trois différentes fonctions de dégradations. "Frugal", "Anxious" et "cool-headed".



L'utilisateur peut choisir une de trois fonctions de dégradations (Figure 10). La première est *frugal*. Ce type de stratégie décrémente, au début, un peu le prix de vente, en espérant que ce prix convienne un agent. Ce prix sera décrémente brusquement avant la fin du temps de vente. La fonction *Anxious*, quant à elle, décrémente le prix sous forme linéaire. Dans ce cas, l'agent veut vendre rapidement. La fonction *cool-headed* se situe entre les deux.

### 3.3 Salles d'enchères

Les salles d'enchères sont des marchés évolués où le rôle des agents est plus complexe et plus structuré. Les agents sont gérés comme dans une vraie salle des ventes et l'utilisation de protocoles spécifiques permet de garantir l'équité entre les acheteurs d'un même produit. Nous pouvons citer AuctionBot [Wurman, 98] et CASBA [Michael, 00].

#### 3.3.1 AuctionBot

Cet agent a été développé à l'Université du Michigan [Wurman, 98]. *AuctionBot* offre un service centralisé d'enchères sur Internet, accessible à la fois aux humains et aux autres agents.

##### 3.3.1.1 Types d'enchère

Deux catégories d'enchères sont supportées par "*AuctionBot*" : simple et double. Dans la première catégorie, les "*bidders*" sont soit des vendeurs soit des acheteurs, alors que la deuxième peut admettre des vendeurs et des acheteurs simultanément.

La catégorie simple dispose de ces types d'enchères :

- “*English auction*” : dans ce type, la procédure de négociation commence par un prix minimal proposé par le vendeur. Il procède à la sollicitation de prix de plus en plus haut. Le gagnant est celui qui offre le plus haut prix.
- “*Dutch auction*” : contrairement au type précédent, l’enchère s’ouvre par un prix maximal proposé par le vendeur. Ce prix diminue jusqu’à ce qu’un acheteur l’accepte. Ce dernier devient alors le gagnant et paie alors son prix.
- “*FPSB*” (*First Price sealed Bid*) : dans ce cas, le gagnant de l’enchère est celui qui a offert le plus haut prix.
- “*Vickrey Auction*” : même mécanisme que celui de “*FPSB*” sauf qu’ici le gagnant est celui qui a le deuxième plus haut prix offert.

La deuxième catégorie est celle de l’enchère *Double* qui dispose plusieurs types d’enchère dont le plus connus sont :

- “*Continuous Double Auctions*” : dans ce type, plusieurs transactions individuelles de type vendeurs et acheteurs sont traités simultanément. Le processus de l’enchère ne s’arrête pas si le prix est conclu. Un exemple typique de ce genre d’enchère est la bourse financière.
- “*Chronological Match Auctions*” : c’est le même principe que *Continuous Double Auctions*, sauf que toutes les offres qui dépassent l’intervalle du temps de l’enchère seront rejetées.

### 3.3.1.2 Protocoles de communication

Plusieurs messages sont fournis par les agents et l’enchère pour supporter les activités de l’enchère. Ainsi, nous pouvons trouver entre autres, le message :

- “*Bid*” : envoyé par l’agent pour proposer une offre;
- “*Bid withdrawal*” : envoyé de l’enchère pour retirer une offre de l’enchère;
- “*Bid admittance*” : envoyé de l’enchère à l’agent pour lui annoncer que l’offre satisfait aux conditions de l’enchère;

- “*Bid rejection*” : envoyé de l’enchère à l’agent pour lui annoncer que l’offre ne satisfait pas aux conditions de l’enchère;
- “*Price quote*” : envoyé de l’enchère à l’agent. Cette méthode spécifie le prix nécessaire pour gagner l’enchère à un temps précis ;
- “*Transaction notification*” : Envoyé de l’enchère aux agents. Cette méthode spécifie les termes de l’échange, y compris le prix, la quantité, etc.

### 3.3.1.3 Architecture de l’agent

“*AuctionBot*” ne fournit aucun agent pour aider les utilisateurs dans la négociation. Si une personne veut proposer une offre, il implémente lui-même et délègue son agent pour la négociation.

## 3.4 Notre système

Cette discussion permet maintenant de faire ressortir les caractéristiques que possédera notre système. Nous nommerons ce dernier « *Pacha* ».

D’après notre étude préliminaire sur de tels systèmes, nous constatons que le client ne profite pas des informations qui existent dans son profil et que seulement le marchand peut profiter de ces informations pertinentes. De plus, la majorité des systèmes ne prennent pas en considération le niveau d’expertise et l’expérience du client qui s’évalue avec le temps. Enfin, ces systèmes ont un niveau d’intelligence limité, parce qu’ils gardent une vue fixe pendant toute la vie des transactions récentes et futures des clients.

Le système *Pacha* offre un mécanisme de conseils pour aider ses clients s’ils sont indécis ou perdus. Il les aide à mieux choisir les caractéristiques du produit qui leur convient. De plus, *Pacha* propose des solutions d’aide à la vente basées sur le raisonnement à partir de cas (CBR) [Bergmann, 99]. Le principe du CBR consiste à identifier et proposer à l’usager des produits similaires à ceux qu’il recherche.

Pacha est orienté vers les PME (Petites-Moyennes Entreprises) incapables d'avoir des grands systèmes informatiques pour présenter leurs produits. Ces PME sont loin de pouvoir entrer en compétition avec des grandes entreprises telles que Amazon [Amazon, 02] et eBay [eBay, 02]. Nous proposons comme solution, de regrouper ces PME dans un marché virtuel pour qu'elles aient la chance de se mesurer aux grandes entreprises.

Pacha comporte aussi les trois premières étapes du modèle CBB (Consumer Buying Behavior), à savoir *l'identification de besoins*, *la sélection du produit* et enfin *la sélection du vendeur*.

Une autre caractéristique de notre système est qu'il utilise la technologie des agents mobiles pour limiter l'utilisation du réseau au strict minimum. Dans le chapitre qui suit, nous expliquerons en détails l'avantage de l'utilisation de cette technologie récente.

Les aspects relatifs à la sécurité dans les agents mobiles restent un problème ouvert, comme on peut le constater à travers les différentes approches mises en place [Harrison, 95]. Pacha dispose d'un serveur (marché virtuel) capable d'accueillir des agents visiteurs et assure un certain niveau de sécurité pour l'utilisateur et au niveau du serveur.

### **3.5 Conclusion**

Ce chapitre a présenté plusieurs systèmes d'agents du commerce électronique. En analysant ces systèmes, nous avons pu déterminer plusieurs caractéristiques que possédera notre logiciel. Il faut maintenant se pencher sur les techniques utilisées pour réaliser la recherche d'un produit. Ce sera l'objet du prochain chapitre qui présentera les techniques de mobilité des programmes et de raisonnement à partir de cas (CBR).

# *Chapitre 4*

## **4 Techniques utilisées**

Notre système utilise la technologie récente des agents intelligents mobiles. Ceux-ci captent les préférences de l'utilisateur et voyagent à travers les serveurs afin de chercher les produits demandés. Cette mobilité rend nos agents plus autonomes et indépendants de l'utilisateur. L'utilisation de Raisonement à base de cas (CBR) est aussi un excellent moyen pour conseiller l'utilisateur et de lui offrir des produits similaires à ceux qu'il préfère. Cette technique (CBR) différencie nos agents et leur permet de devenir de plus en plus sophistiqués et intelligents.

Ce chapitre fait le point sur les avantages des ces deux techniques adoptées par notre système. Les implémentations et les algorithmes seront présentés dans le prochain chapitre.

### **4.1 Raisonement à base de cas (CBR)**

Tout d'abord, cette section vise à décrire l'approche de calcul de recommandation, basée sur les techniques de raisonnement à partir de cas et à illustrer l'application de cette approche pour aider à trouver un produit similaire via le système Pacha.

#### **4.1.1 Description de CBR**

De manière générale, la réutilisation de comportements utilisateur s'appuie sur le raisonnement à partir de cas qui est une approche de résolution de problèmes basée sur la

réutilisation par analogie d'expériences passées appelées « cas » au cours d'un cycle de raisonnement [Leake, 96] et [Aïmeur, 00]. Un cas est composé de deux parties : la description du problème et la solution qui lui est appliquée. Plus précisément, un cas est une expérience passée permettant au système de résoudre des problèmes plus efficacement ou d'éviter les échecs passés : il doit donc être utile pour le raisonnement. Dans ce but, un ensemble de caractéristiques est extrait ou sélectionné dans la représentation de chaque cas. Ces caractéristiques, appelées indices, déterminent dans quelle situation un cas est applicable et utile. Les indices doivent tout d'abord être prédictifs, c'est-à-dire qu'ils doivent être responsables de l'élaboration de la solution mise en œuvre dans le cas et permettre ainsi de déterminer dans quelle mesure une solution est applicable pour un autre problème. Les indices doivent également être suffisamment abstraits pour permettre la réutilisation des enseignements du cas pour différents problèmes, tout en restant suffisamment concrets pour être facilement identifiables dans un nouveau problème.

Lorsqu'un nouveau problème est à résoudre, un cas cible est construit avec la partie solution inconnue, celle-ci doit être apportée par le raisonnement. Les cas sources représentent des expériences passées qui sont stockées dans une mémoire. Ces deux types de cas partagent le même formalisme de représentation. L'objectif du raisonnement est de transférer les enseignements pertinents des cas sources pour élaborer la solution du cas cible.

#### **4.1.2 Pourquoi utilise-t-on le CBR ?**

L'approche CBR offre de nombreux avantages. Elle permet d'éviter les problèmes d'acquisition de connaissance ("*knowledge bottleneck*") qui rendent difficile la construction de bases de connaissances de taille importante. Pour certaines applications, l'approche CBR est plus simple à mettre en œuvre que les approches basées sur un modèle du domaine (e.g. base de règles). De plus, le CBR est particulièrement bien adapté pour les applications ayant les caractéristiques suivantes:

- la tâche est accomplie par des humains expérimentés dans leur domaine et ces expériences sont disponibles dans une base de données, dans des documents ou chez un expert humain;
- une analyse détaillée du domaine n'est pas nécessaire pour obtenir des solutions satisfaisantes et la tâche n'exige pas une solution optimale ("*satisficing solution*");
- un modèle du domaine ne peut être élaboré parce que le domaine est mal formalisé (peu de documentation, expert non disponible) ou parce qu'il n'existe pas de principes généraux qui sont éprouvés (exemple : comment investir à la bourse);
- les situations sont répétitives et les solutions sont réutilisables. Ces situations, dites monotones, sont telles que de petites différences dans le problème entraînent de petites différences dans la solution. De plus, une solution valide à un moment, le demeure à un autre moment.

Pour notre système, la méthode de CBR sert d'abord à conseiller l'utilisateur en se basant sur son profil technique qui contient l'historique de ses activités. Alors, tant que l'utilisateur utilise le système, le niveau d'expertise s'améliore et en même temps les conseils de notre système deviennent de plus en plus sophistiqués et toujours compréhensifs pour cet utilisateur. D'autre part, en utilisant la méthode de CBR, notre système propose des offres similaires à celles que l'utilisateur préfère. Ces dernières ont déjà été utilisées par d'autres utilisateurs et qui ont obtenu des bons résultats. Ces offres sont stockées dans une base de données.

### **4.1.3 Composantes d'un système à base de cas**

Ce raisonnement se décompose habituellement en quatre phases principales qui peuvent être totalement ou partiellement automatisées (Figure 11).

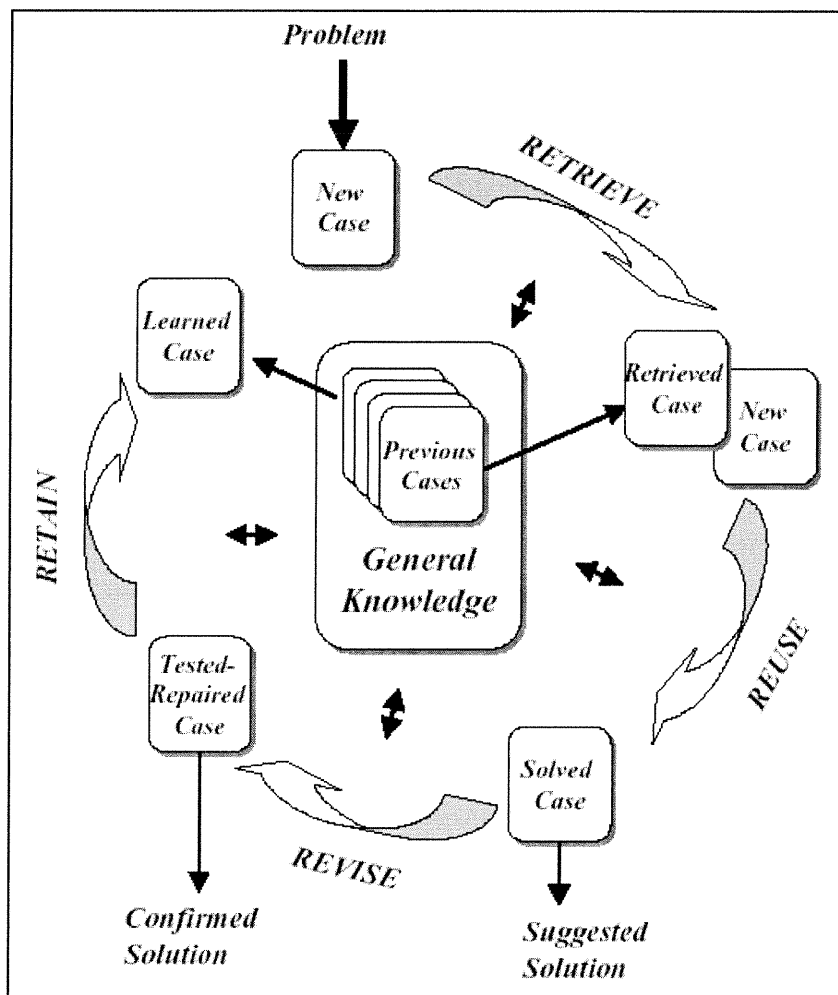


Figure 11 : Modèle de processus CBR.

(extrait de [Leake, 96]).

Le calcul des recommandations est effectué au cours d'un raisonnement en quatre phases, selon la commande courante (ordinateur par exemple) d'un utilisateur :

- **Recherche** ("Retieve") cette phase permet de déterminer les cas de base qui sont les plus similaires à la commande traitée. La procédure de recherche de similarité est implantée par une approche basée sur les plus proches voisins ("*k-nearest-neighbors*"). Cette approche utilise des métriques de similarité pour mesurer la



correspondance entre chaque cas et le problème à résoudre. Les meilleurs cas sont alors retournés.

- **Réutilisation** (“*reuse*”). Chaque cas trouvé représente une offre et la phase de réutilisation est de proposer la liste des produits de ces offres à l'utilisateur courant.
- **Révision** (“*revise*”) La phase de révision est effectuée par l'utilisateur lorsqu'il continue sa recherche et qu'il évalue les offres proposées s'il le désire. La révision se poursuit jusqu'à la fermeture de la tâche courante et met en attente tous les raisonnements effectués pour une recherche (offres recommandées et confirmées).
- **Apprentissage et maintenance de la mémoire** (“*retain*”) : L'apprentissage prend en considération la description de la commande courante et l'ensemble des raisonnements associés pour en faire une synthèse. La mémoire est alors mise à jour : par la création de nouveaux cas, par la mise à jour des cas et par l'ajout des descriptions de la commande courante. Parallèlement au processus de raisonnement, les cas et les commandes devenus obsolètes sont régulièrement effacés de la mémoire.

#### 4.1.4 Modèles CBR

Il existe plusieurs modèles pour le raisonnement à base de cas. Ceux-ci sont regroupés en trois grandes familles: conversationnelle, textuelle et structurelle. Nous nous sommes particulièrement intéressés dans notre système au modèle structurel.

- **Modèle conversationnel**

Comme son nom l'indique, le modèle CBR conversationnel mise sur l'interaction entre l'utilisateur et le système (d'où la notion de “conversation”) pour définir progressivement le problème à résoudre et pour sélectionner les solutions les plus appropriées [Aha, 01].

Pour obtenir des solutions satisfaisantes, l'utilisateur doit avoir a priori une bonne idée de tous les facteurs pouvant influencer la résolution de son problème. Toutefois, il est difficile de déterminer à l'avance les aspects de la situation qui sont importants. Par ailleurs, cette exigence présuppose une expertise du domaine d'application, ce qui n'est pas le cas chez tous les usagers de systèmes CBR. Des usagers novices éprouvent parfois des difficultés à bien caractériser une situation à l'aide de valeurs numériques ou symboliques. Par exemple, les préposés des centres d'appels ne connaissent pas tous les aspects techniques des produits vendus par la compagnie, et leurs clients encore moins. Le modèle conversationnel a donc été proposé pour surmonter ces difficultés.

- **Modèle textuel**

Les travaux sur le raisonnement à base de cas textuel portent sur la résolution de problème à partir d'expériences dont la description est contenue dans des documents textuels. Dans cette approche, les cas textuels sont soit non-structurés ou semi-structurés. Ils sont non-structurés si leur description est complètement en "free-text". Ils sont semi-structurés lorsque le texte est découpé en plusieurs portions étiquetées par des descripteurs tels que "problème", "solution", etc. Un cas textuel non-structuré est un cas qui a un seul attribut dont la valeur est textuelle tandis qu'un cas textuel semi-structuré est un cas dont un sous-ensemble des attributs est textuel. Pour ce modèle, la représentation textuelle des cas joue habituellement un rôle important dans la résolution du problème. Elle peut être une finalité en soit: par exemple, obtenir le texte d'un jugement légal servant de jurisprudence à une nouvelle cause. Elle peut aussi décrire une situation et une solution qui ne peuvent être facilement codifiées selon un schéma de représentation de connaissance.

- **Modèle structurel**

Le modèle structurel a émergé des premières vagues de systèmes CBR. Dans ce modèle, toutes les caractéristiques importantes pour décrire un cas sont déterminées à l'avance par

le concepteur du système. Ainsi le concepteur doit être capable d'élaborer un modèle de données du domaine applicatif. Tel qu'illustré à la Figure 12, les cas sont complètement structurés et sont représentés par paires <attribut, valeur> (comme un objet). D'un point de vue applicatif, un attribut représente une caractéristique importante du domaine d'application. Les échelles de valeurs les plus fréquemment utilisées pour structurer les attributs sont les entiers/réels, les booléens et les symboles.

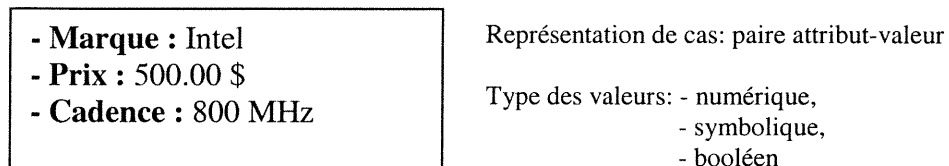


Figure 12 : Exemple de structuration d'un cas en CBR structurel

#### 4.1.5 Fonction de similarité

La fonction de similarité est un moyen utilisé pour trouver la correspondance entre plusieurs « cas ». En effet, cette fonction représente la *distance* entre les valeurs de même attribut de deux cas. Celle-ci peut être définie de plusieurs façons; les plus connues sont celles des mesures Euclidienne [Christoph, 95].

#### 4.1.6 Distance

La distance est un moyen de mesurer la similarité entre deux cas. Chaque cas est défini par la pair  $(X, \text{classe}(X))$  où  $X$  désigne la description du cas et où  $\text{classe}(X)$  désigne la classification de ce cas. La distance entre deux cas  $(X, \text{classe}(X))$  et  $(Y, \text{classe}(Y))$  est estimée par une somme pondérée des distances de ses attributs. Algébriquement, la distance est donnée par la formule suivante :

$$\text{Distance}(X,Y) = \sum d_{(x_i-y_i)}$$

Où  $X, Y$  : sont les deux cas et où  $x_i, y_i$  sont les attributs respectifs des cas  $X, Y$ .

La distance  $d_{(X_i-Y_i)}$  entre deux attributs de deux cas est prédéfinie par le concepteur. Comme les attributs d'un cas n'ont pas tous la même importance et que celle-ci varie d'une situation à l'autre, un poids est accordé à chaque attribut de chaque cas. Ces poids permettent de pondérer la similarité globale entre deux cas en accordant un "vote" plus important aux attributs les plus méritants.

Pour illustrer le calcul des distances entre les cas, prenons l'exemple d'un cas X où l'item est le CPU d'un ordinateur avec les attributs suivants :

<b>X</b>	<b>Attributs</b>
	<ul style="list-style-type: none"> <li>- <b>Marque_x</b> : Intel</li> <li>- <b>Prix_x</b> : 500.00 \$</li> <li>- <b>Cadence_x</b> : 800 MHz</li> </ul>

Une fois le « cas » identifié, le concepteur peut définir les critères de comparaison suivant les priorités des ses attributs.

Dans notre exemple, nous pouvons observer que l'attribut le plus prioritaire est la « Marque\_x » du CPU. Ainsi, nous pourrions définir la distance entre deux attributs « Marque\_x » et « Marque\_y » d'un autre cas Y existant dans la base de données CB de cas (Case Based) de la façon suivante :

**Si** ( $Marque\_y = Marque\_x$ ) **alors**  
 $d(Marque\_x, Marque\_y) = 0$  ;// distance est très proche  
**Sinon** alors  $d(Marque\_x, Marque\_y) = seuil$ ; // distance est très grande.

*A noter que seuil est une valeur très grande qui sert à identifier la différence de distance entre deux attributs.*

Après avoir calculé la distance entre les attributs les plus prioritaires, on procède à calculer graduellement les autres distances. Pour celui de l'attribut « Cadence », l'utilisateur choisit la valeur de cadence et aussi les deux valeurs (maximal et minimal) qui représentent les bornes de son choix. Ci-dessous, nous montrons qu'il est alors possible de définir la distance :

```

Si (Cadence_y ∈ [Cadence_min, Cadence_max]) Alors
Debut
    d(Cadence_x, Cadence_y) = 0 ;
Sinon
    Si (Cadence_y < Cadence__min) Alors
        Debut
            Si (Cadence_y >= Cadence__min-100) Alors
                d(Cadence_x, Cadence_y) = Cadence__min - Cadence_y;
            Sinon d(Marque_x, Marque_y) = seuil ;
        Fin
    Sinon
        Si (Cadence_y > Cadence__max) Alors
            Debut
                Si (Cadence_y <= Cadence__max + 100) Alors
                    d(Cadence_x, Cadence_y) = Cadence__max - Cadence_y ;
                Sinon d(Marque_x, Marque_y) = seuil ;
            Fin
Fin.

```

Pour l'attribut « Prix », on procède de la même façon que pour la « Cadence ». Étant donné que le prix maximal est entré par l'utilisateur, la distance entre deux prix est définie par :

```

Si Prix_y <=Prix_max Alors
Debut
    d(Cadence_x, Cadence_y) = 0 ;
Sinon
    Si (Prix_y <= 2 * Prix_max) Alors
        d(Cadence_x, Cadence_y) = (100 * (Prix_y - Prix_max)) /Prix_max ;
    Sinon
        d(Cadence_x, Cadence_y) =seuil ;
Fin.

```

Ayant calculé toutes les distances entre les attributs, la distance globale entre deux cas (X,Y) est la somme des distances de leurs attributs. La valeur de cette distance définit la similarité estimée entre deux cas. Cependant, si la valeur est grande, c'est-à-dire, qu'elle dépasse le seuil, ceci montre que les deux cas ne sont pas similaires.

#### **4.1.7 Adaptation**

Le mécanisme d'adaptation consiste à mettre à jour la base de données CB continuellement suivant les résultats de chaque nouveau cas. Si une requête obtient un bon résultat, le système compare cette requête avec celles existantes dans la base de données CB, si c'est une nouvelle requête, il ajoute les caractéristiques correspondantes de cette requête à la base CB.

#### **4.1.8 Algorithme**

Cet algorithme sert à utiliser la fonction de distance pour trouver les commandes similaires ou les plus proches [Christoph, 95] :

- chercher toutes les commandes qui ont le même type que celle de l'utilisateur;
- évaluer la distance de chaque commande;
- accepter toutes les commandes qui respectent la distance ne touchant pas le seuil déjà défini;
- trier les commandes sélectionnées selon la distance en ordre croissant.

Nous avons implémenté nous-mêmes cet algorithme qui sera appliqué dans le chapitre suivant.

## 4.2 Les agents mobiles

Sujet de recherche depuis plusieurs années, les agents mobiles ne sont toujours pas arrivés dans le domaine du commerce électronique. Nous verrons que l'approche est très puissante, mais qu'elle est aussi exigeante.

Un agent mobile est un programme qui se déplace (physiquement) d'un serveur à un autre sur le réseau. Le code du programme objet est transporté avec ses données et continue ensuite l'exécution sur la nouvelle machine. Nous nous sommes inspirés des travaux proposés dans [White, 95] pour inclure un support de mobilité dans les langages de programmation. [Cabri, 98] offre une comparaison des différentes approches de calcul distribué comprenant le serveur de client, l'évaluation à distance et le code mobile. De nombreuses plates-formes logicielles pour les agents mobiles ont été développées, citons Voyager [Voyager, 97], Aglets [Alf, 02] et D'Agents (autrefois appelé Agent Tcl) [Robert, 01].

Dans les deux sections qui suivent, nous présenterons l'avantage de l'utilisation de la technique des programmes mobiles. Pour cela, nous allons comparer cette technique au modèle client-serveur traditionnel et utilisé dans la plupart des systèmes commerciaux.

### 4.2.1 Exemple en client/serveur

Grâce à la généralisation d'Internet, tout le monde connaît le modèle client/serveur. Vous demandez une page HTML, un serveur vous l'envoie (Figure 13-a). Le serveur peut aussi *calculer* votre page. Par exemple, il peut exécuter un script CGI qui consulte une base de données de chaussures de sport. Ensuite, il vous envoie la liste des modèles qui sont disponibles dans votre pointure (Figure 13-b).

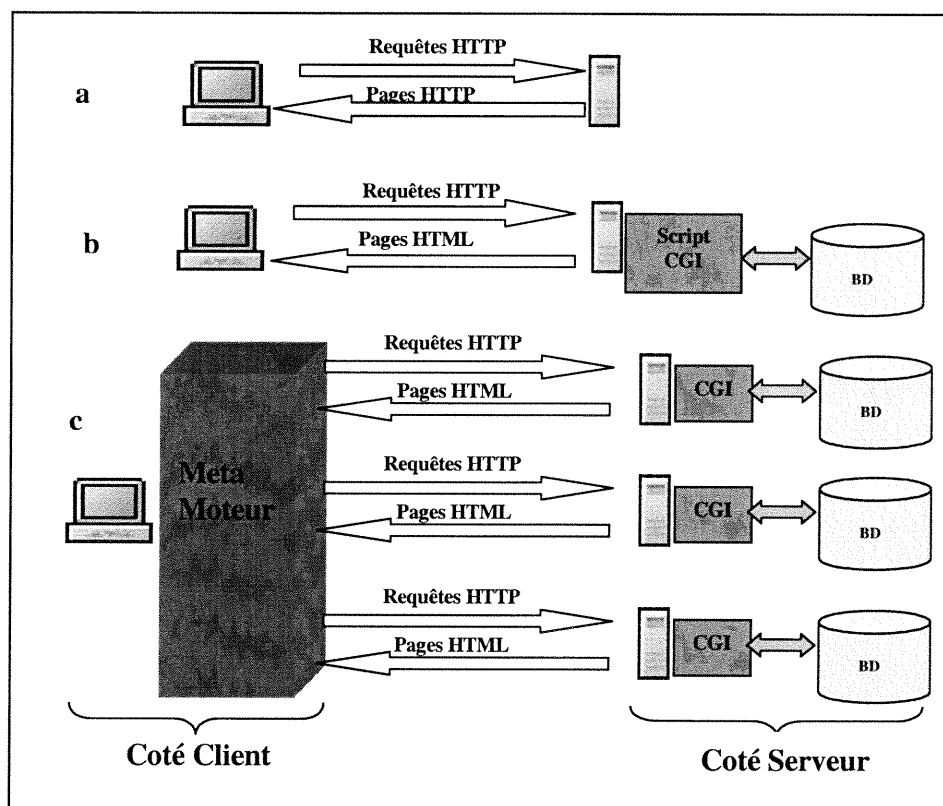


Figure 13 : Modèle Client/Serveur



Pour comparer les prix de plusieurs magasins, vous devez trouver les sites, vous rendre sur chacun d'eux, effectuer vos recherches et noter ce que vous y trouvez. Une telle démarche prend un temps beaucoup trop élevé. Une solution consiste à ajouter un intermédiaire : un meta moteur de recherche qui gère le commerce électronique (Figure 13-c). Ce dernier interroge pour vous chaque magasin dont il connaît les langages d'interrogation et les critères proposés. Après quelques instants de recherche, vous aurez la liste des dix modèles les moins chers.

Cette solution présente encore d'importantes contraintes. La machine client doit être connectée tout au long de la recherche, elle se charge de tous les calculs ce qui emmène une certaine congestion du trafic: la requête est envoyée à chaque fois et les serveurs retournent la totalité des pages complètes de résultats (Copernic.com).

Si l'on déplace le meta-moteur sur un site Web, le client se trouve alors dans une situation beaucoup plus intéressante : la requête est envoyée une seule fois et un nombre restreint de résultats est retourné. Malgré tout, nous ne faisons que déplacer le problème. Le serveur du site qui fera les recherches rencontrera les mêmes contraintes. Un autre type d'architecture est donc à envisager. Nous allons voir comment les agents mobiles peuvent répondre à ce type de problèmes.

#### **4.2.2 Exemples à base d'agents mobiles**

Dans un modèle à agents mobiles, le client n'exécute plus toutes les tâches, il délègue le travail. On considère ici l'agent mobile comme un ensemble code/données autonome capable de se déplacer entre les différents environnements d'exécution des machines hôtes d'un réseau (Figure 14).

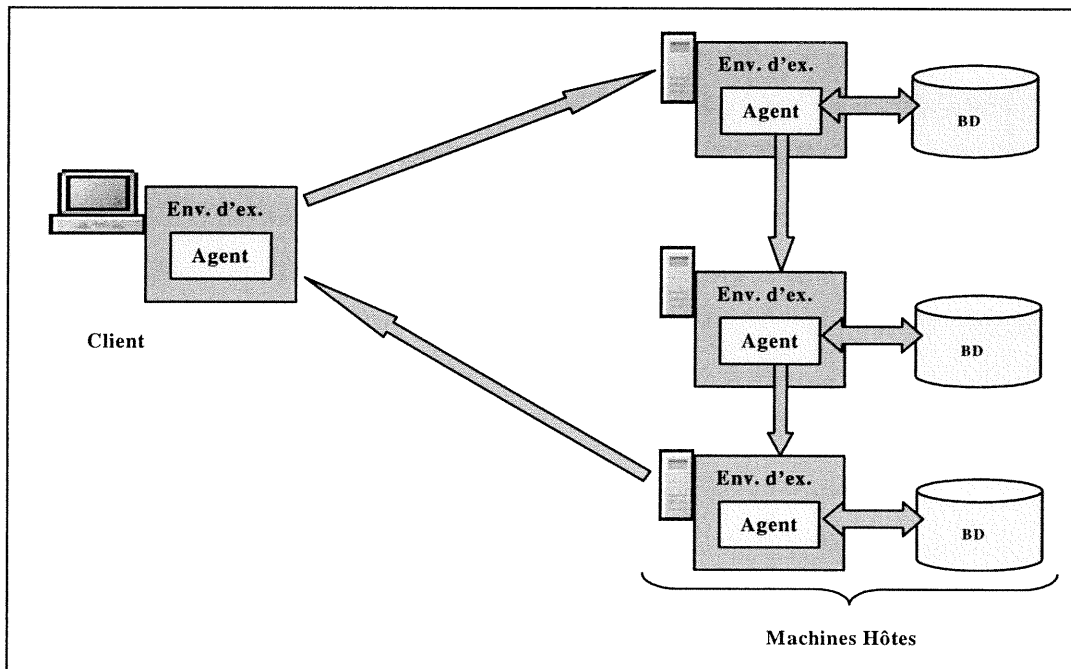


Figure 14 : Modèle à agent mobile

Tout ce qui est fait en mode client/serveur pourra être fait à l'aide d'agents mobiles. Une partie des applications ne nécessitera pas ce nouveau modèle, mais certaines applications se révéleront plus efficaces en l'utilisant. Voici quelques exemples :

- la collecte d'informations avec filtrage (personnalisation de recherche) ;
- la surveillance (si tel évènement se produit sur le réseau, alors l'agent prévient le client) ;
- la distribution d'informations (les données ne partent qu'une seule fois) ;
- la négociation (l'agent part avec une offre du client et revient avec le meilleur prix obtenu) ;
- le calcul parallèle (l'agent peut se dupliquer sur plusieurs serveurs avec les différentes parties d'un calcul à effectuer).

### 4.2.3 Caractéristiques de l'agent mobile

Dans le chapitre 4, nous définirons la notion d'environnement d'exécution ; Considérons pour l'instant qu'il s'agit de l'environnement qui permet à un agent d'être accueilli sur un hôte. Un agent mobile, quant à lui, peut être considéré comme une entité logicielle disposant des facultés suivantes :

- **comportement.** Un agent est pourvu d'un programme qui lui dicte les tâches qu'il doit accomplir. On dit qu'il agit par *délégation* pour le client. Un agent peut ainsi venir avec une compétence particulière sur un hôte, par exemple avec une feuille de style qu'il appliquera au document à envoyer.
- **autonomie.** Un agent agit indépendamment du client. Il décide lui-même là où il va et ce qu'il doit y faire, en fonction du comportement qui lui a été donné.
- **mémoire.** Un agent dispose d'un potentiel de mémorisation permettant par exemple de stocker des informations.
- **communication.** Un agent doit pouvoir interagir avec les différents environnements (le client et les hôtes) ainsi qu'avec les autres agents (locaux ou distants).
- **sécurité.** Un agent doit se protéger des attaques extérieures : par rapport aux autres agents, à ses environnements d'exécution et pendant son déplacement sur le réseau. Son intégrité et la confidentialité des données qu'il transporte doivent être assurés. Par exemple, on ne doit pas pouvoir lui *voler* un numéro de carte de crédit ou modifier son programme. On ne doit pas non plus pouvoir pratiquer la mutation d'un agent à son insu. C'est ainsi que l'on est amené à crypter les agents.

Malgré de sérieuses précautions à prendre en terme de sécurité, les caractéristiques des agents mobiles laissent apparaître un potentiel de réponses efficaces à certains types de problèmes.

Tout d'abord, les agents mobiles vont permettre de spécialiser les serveurs pour que ces derniers répondent à des besoins qu'ils ne connaissent pas forcément à l'avance. A titre d'exemple, nous pourrions associer à un agent une tâche d'installation un service donné sur le serveur. Une fois ce service est installé, le serveur peut l'adopter pour une utilisation ultérieure.

Ensuite, les agents mobiles pourraient diminuer l'utilisation du réseau dans des cas particuliers. En général, on aura simplement deux échanges : l'envoi d'un agent et la réception des résultats. De ce fait, les temps de connexion au réseau seront plus courts et on peut très bien imaginer des machines nomades connectées de façon intermittente. Ces dernières seront déconnectées pendant toute la durée du travail de l'agent. Cette approche nous permettra aussi d'alléger la charge de calcul pour le client : l'agent agira par délégation pour le client, les calculs seront distribués sur les différents serveurs hôtes.

En ce qui concerne les performances, des études montrent que dans certains types d'applications et à partir d'un certain seuil (puisqu'il faut tenir compte de la taille de l'agent en tant que tel), les agents mobiles deviennent beaucoup plus intéressants que le modèle client/serveur [Ismail, 98].

### **4.3 Conclusion**

En bref, toutes les activités de l'utilisateur seront stockées dans un profil distinct sur la machine locale. Chaque activité est formée de la commande et des résultats obtenus. Ces activités indiquent le niveau d'expertise de l'utilisateur. En utilisant le CBR, notre système accédera à ce profil pour aider l'utilisateur à choisir des caractéristiques compatibles d'un produit quelconque. Toutes les commandes qui ont abouti à des résultats seront stockées dans une base de données sur le serveur. Notre système, en utilisant aussi le CBR va

chercher et proposer des offres à ceux qui demandent de l'aide. De plus, la mobilité de notre agent le rendra indépendant et autonome de l'utilisateur. L'agent voyage jusqu'à l'hôte pour s'exécuter et dès qu'il a fini sa tâche, il revient avec les données.

Ceci termine la présentation des concepts théoriques sous-jacents à notre solution. Nous pouvons maintenant exposer les détails de l'implémentation du système que nous avons réalisé suite à cette recherche.

## *Chapitre 5*

### **5 Pacha**

L'utilisation des systèmes à agents mobiles intelligents devient de plus en plus populaire dans divers domaines traitant du commerce électronique. Cette progression est due essentiellement aux avantages offerts par ces systèmes comparativement aux systèmes classiques. En effet, l'exécution d'un programme à distance dans un marché virtuel ainsi que l'autonomie de ce programme simplifient beaucoup des tâches difficiles à réaliser dans le paradigme classique client-serveur. En dépit de la croissance des travaux traitant des agents, nous constatons qu'il y a un long chemin à suivre avant d'atteindre le but d'avoir un système *multi-agents* hautement intelligent. Dans ce chapitre, nous présentons notre contribution dans ce domaine, en implantant un système *multi-agents* intelligent tirant profit de la technologie de mobilité des agents. Pacha montre les avantages de l'utilisation du marché virtuel pour accueillir des agents mobiles en provenance d'utilisateurs sur le web. Ces agents communiquent et collaborent avec les agents du système se trouvant dans le marché virtuel afin d'aider l'utilisateur et les commerçants à réaliser leurs objectifs.

L'originalité de ce travail réside dans l'utilisation de la mobilité des agents et l'implantation des techniques assurant un certain niveau d'intelligence. Effectivement, cette mobilité résout beaucoup de problèmes qu'on trouve par contre dans le modèle client-serveur. Notons que la mobilité rend les agents envoyés sur le marché plus autonomes et indépendants des usagers pendant l'exécution à distance. Ainsi, l'intelligence du système facilite le processus de recherche du côté de l'utilisateur en lui offrant des conseils pertinents qui lui permettent de bien choisir les produits convenables à sa demande.

Dans ce chapitre, nous commençons par une présentation globale de l'état de l'architecture du système et le fonctionnement de ses différentes composantes. Ensuite, nous décrivons les protocoles disposés par Pacha et ainsi que la technique utilisée pour l'implémentation de ce système. Enfin, le fonctionnement de ce système sera illustré par un exemple.

## 5.1 Architecture globale de Pacha

Notre système Pacha est constitué principalement de deux parties essentielles « *Partie-Usager* » et « *Partie-Serveur* » (Figure 15). La première contient des modules qui gèrent l'interaction avec l'utilisateur. La deuxième représente le « *Marché Virtuel* ». Elle est composée de modules qui organisent l'hébergement des divers agents et aussi la gestion des bases de données existantes. Ces deux parties communiquent à travers le réseau Internet. Dans cette section, nous décrivons en détails tous les composants de ce système ainsi que l'interaction entre les différents agents. Pour illustrer la structuration de notre modèle, nous commençons par une figure qui représente l'architecture globale du système. Dans les prochaines sections nous expliquerons les différents composants des parties mentionnées en spécifiant les tâches associées à chacun d'eux.

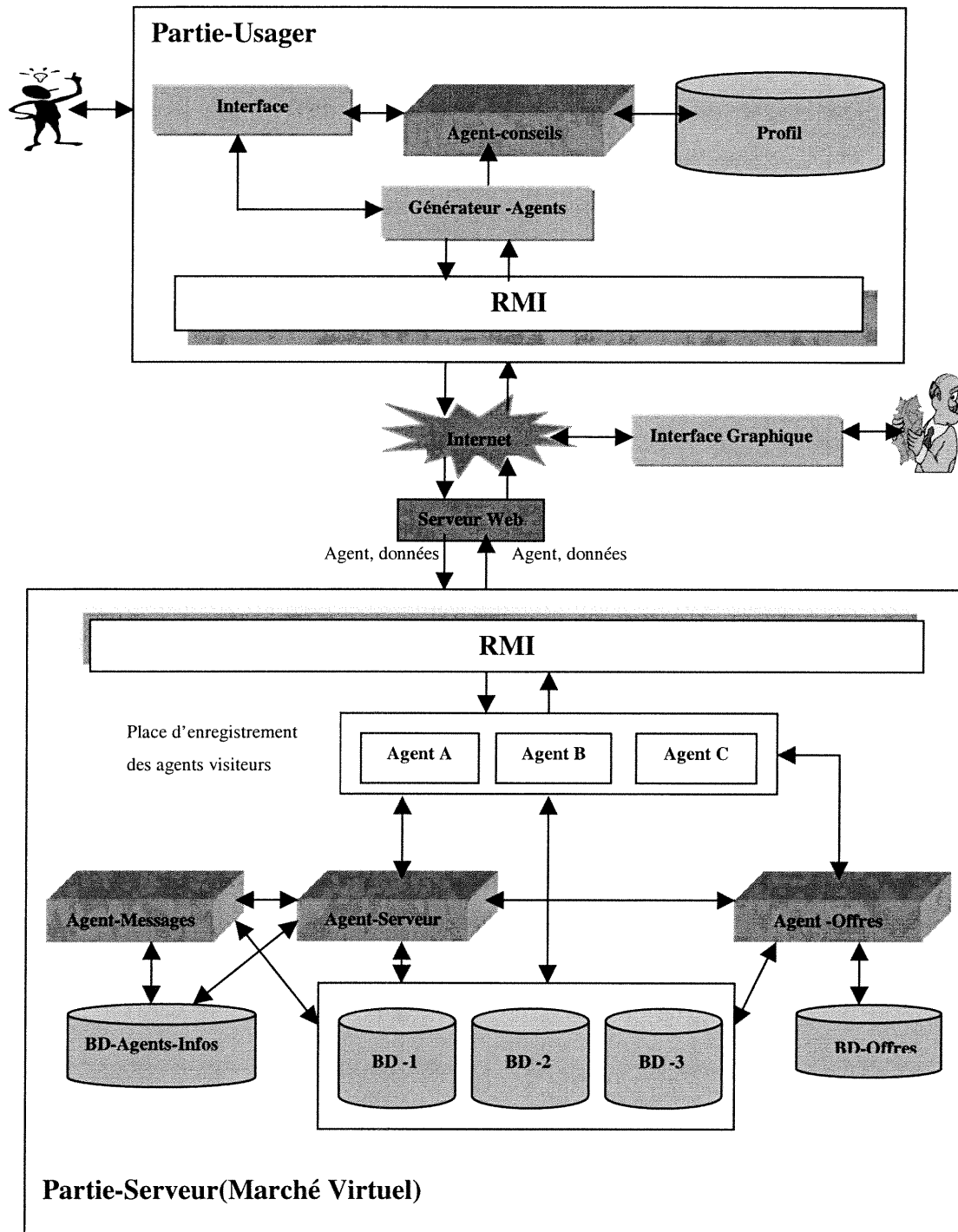


Figure 15 : Architecture du système Pacha.



### 5.1.1 Partie-Usager

C'est un outil qui réside dans la machine locale du client. Il sert à saisir les préférences de l'utilisateur pour créer et envoyer un agent dédié à la recherche de la meilleure offre pour l'utilisateur. Cet outil se compose de quatre éléments :

- **Interface :**

Ce module présente la face visuelle du système pour aider l'utilisateur à bien apercevoir toutes les fonctionnalités qui lui sont fournies. Pour cela, l'interface contient des composants graphiques aidant l'utilisateur à bien préciser ses intérêts. Les tâches associées à ce module sont l'affichage et la présentation des profils, transmission des choix de l'utilisateur vers les autres modules, et aussi, la coopération avec ces modules pour afficher les résultats trouvés. Cependant, les fonctionnalités offertes à l'utilisateur peuvent dépendre de l'application utilisée, comme acheter/vendre un produit ou bien entrer dans une enchère, etc. Notre implantation consiste en une application de recherche des produits spécifiés telle que la « recherche des accessoires informatiques ». Cependant, Pacha peut être adapté à différents types d'applications.

- **Profil :**

La « Partie-Usager » fonctionne autour de cette composante essentielle. Le profil est unique pour chaque utilisateur. Il contient deux utilités principales : la première est pour sauvegarder les demandes de l'utilisateur concernant un produit «X » et aussi les résultats de recherche fournis par les agents concernant les produits demandés ou alternatifs provenant du marché virtuel. La deuxième utilité de ce module, est l'accumulation d'expériences antérieures. À titre d'exemple, si l'utilisateur soumet une commande, le profil sauvegarde le résultat de cette commande pour une consultation ultérieure. Cette consultation est faite via l'agent de conseil, qui utilise les données enregistrées dans le profil pour augmenter son niveau d'expertise afin de mieux servir l'utilisateur à la prochaine requête.

- **Agent-Conseils :**

La caractéristique prépondérante qui distingue le système Pacha des autres systèmes traditionnels (qui utilisent ou pas des agents) est celui qui concerne les conseils fournis à l'utilisateur en vue de l'aider à trouver des produits, ceci grâce à un profil personnalisé.

Le traitement des requêtes dans les systèmes traditionnels se fait du côté du serveur. Ce dernier cherche les données relatives à la requête et envoie le résultat. Si la demande contient des informations incompatibles, il sera impossible d'aboutir à des résultats, par conséquent, la réponse sera toujours nulle. L'inconvénient de cette approche est que l'utilisateur peut attendre un temps considérable sans avoir des résultats précis et ou des conseils.

Afin d'améliorer la qualité de recherche, Pacha utilise la notion d'agent dédié aux conseils pour rendre le processus de recherche plus intelligent. Avant le lancement de l'agent, l'utilisateur peut demander l'aide de cet agent qui réside dans la machine locale. Cet agent accède périodiquement au profil technique de l'utilisateur et forme une idée sur son niveau d'expertise qui évolue au fur et à mesure de l'utilisation du système. L'« *Agent-Conseils* » oriente l'utilisateur afin de choisir les meilleures commandes en se basant sur l'historique des recherches déjà effectuées. Il propose également des solutions qui conviennent à l'utilisateur si ce dernier est indécis ou perdu. À la fin, il sauvegarde les caractéristiques de toutes les tâches spécifiées par l'utilisateur dans son profil. Ce mécanisme de conseil est basé sur l'algorithme présenté dans la section 4.1.8.

- **Générateur-Agents :**

Dans le paradigme classique de communication « *client-serveur* », l'utilisateur envoie une requête visant un produit et attend en mode bloquant (IDLE) le retour du résultat. Cependant, ce modèle souffre d'un problème d'attente ennuyeux pour l'utilisateur. Contrairement aux systèmes traditionnels, Pacha permet à l'utilisateur de choisir une requête donnée et en attendant le résultat, l'utilisateur peut faire d'autres requêtes en parallèle sans se soucier de la requête précédente déjà envoyée. Le « *Générateur-Agents* » a pour but

principal d'assurer un mécanisme multi-tâches aux systèmes. Techniquement, cet agent est responsable de préparer, créer, envoyer et recevoir des agents dont leurs tâches sont définies par les préférences de l'utilisateur. Ces agents peuvent être indépendants dans leur but et leur nature. Lors de réception d'un agent, le « *Générateur-Agents* » alerte l'utilisateur en affichant une fenêtre comprenant l'identification de l'agent arrivé (nom, date de départ, date d'arrivée, etc.) et aussi le résultat obtenu. D'autre part, dès le retour d'un agent, le « *Générateur-Agents* » communique avec l'agent de conseil pour lui transmettre les résultats apportés par cet agent. L'agent de conseil sauvegarde ces résultats dans le profil de l'utilisateur pour des utilisations ultérieures.

### **5.1.2 Partie-Serveur « Marché Virtuel »**

Pour la majorité des systèmes à agents, le problème qui se pose est l'hébergement des agents dans le site du marchand. Ce dernier n'accepte dans aucun cas d'accueillir des agents de sources inconnues (le Web) ni de les laisser s'exécuter sans contraintes dans son système. Ce refus revient principalement à des critères de sécurité (problème d'un manque de confiance). Pour éviter ce genre de problème, nous avons adopté le concept d'un marché virtuel. Ce dernier sert à intervenir comme un « third party » entre l'utilisateur et le marchand en prenant la charge d'installer les bases de données des différents marchands. Il est également capable d'héberger des agents provenant de différentes sources à distance. Ainsi, le manque de confiance entre un marchand et les agents est résolu. En effet, le marché virtuel possède un système de sécurité indépendant qui exige des contraintes sur l'exécution des agents visiteurs en les empêchant de modifier les données existantes.

Le marché virtuel est composé essentiellement d'un groupe d'agents coopératifs et d'un ensemble des bases de données. Nous décrivons ces agents ainsi que les tâches associées à chacun d'entre eux.

- *Agents-Visiteurs* :

Rappelons que notre système adopte plusieurs agents (« *Agent-Conseils* », « *Générateur-Agents* », *etc.*) dont la tâche est bien définie. Ainsi, l'agent qui voyage entre les différents sites, nous l'appelons simplement « *Agent-Visiteur* ». Ce dernier a pour but de se déplacer et de s'exécuter à distance en accomplissant les tâches qui lui sont associées. À titre d'exemples, la recherche d'un produit, comparaison des prix, surveillance, enchère et d'autres aussi. Dans notre système, nous avons implanté un agent consacré à la recherche des produits et également à la comparaison des prix. Cependant, l'implantation des autres types d'agents est toujours possible avec Pacha.

La mission de l'« agent-visiteur » est tout d'abord de chercher un produit dans le marché virtuel. Dans le cas de l'indisponibilité de ce produit, l'« agent-visiteur » peut demander de l'aide à d'autres agents dans ce marché. Ces derniers sont dédiés à cette tâche.

Le point prépondérant dans notre système est que l'« agent-visiteur » soit complètement *autonome* et *indépendant* de l'utilisateur qui l'a envoyé. Si ce dernier n'est pas remis en connexion, l'agent peut rester dans le marché virtuel pour un certain délai. Si la connexion est rétablie, l'agent retourne vers l'utilisateur en apportant les résultats obtenus. Ce mécanisme de mobilité des agents est précieux dans plusieurs situations et il différencie notre système Pacha des autres systèmes traditionnels. Nous pouvons récapituler des cas où l'utilisation de ce mécanisme est indispensable. En effet, l'utilisateur peut avoir une panne imprévue dans la connexion ou bien dans sa machine, ce qui empêche la continuité de la phase de traitement de la requête. Le mécanisme de mobilité des agents en Pacha tolère cette panne en permettant à l'utilisateur de récupérer les résultats après le rétablissement de la connexion.

D'autres utilités de ce mécanisme seront discutées dans le chapitre suivant.

- **Agent-Serveur :**

Cet agent est le cœur du marché virtuel. Pour des raisons de sécurité, le marché doit avoir un contrôleur qui supervise la réception des agents visiteurs. Tout d'abord, lors de l'arrivée d'un « *Agent-Visiteur* » au marché, l'« *Agent-Serveur* » l'enregistre dans un emplacement spécifié en lui donnant la permission d'accéder aux bases des données. Ainsi, il récolte quelques informations reliées à cet agent (date d'arrivée, IP adresse de la machine source, e\_mail, date du départ, type de la commande). Ces informations seront stockées dans une base de données particulière « *DB-Agents-Infos* » pour être utilisées par l'administrateur du marché. À la fin de la tâche de l'« *Agent-Visiteur* », l'« *Agent-Serveur* » enregistre les résultats de recherche pour une utilisation future par l'« *agent-Messages* ». Dans le cas où le résultat de la recherche est négatif, l'« *Agent-Serveur* » transmet l'ID d'« *Agent-Visiteur* » à l'« *Agent-Offres* » pour l'aider à trouver une offre similaire à sa demande.

D'autre part, l'« *Agent-Serveur* » surveille la résidence des agents visiteurs dans le marché. Techniquement, la durée de vie de chaque agent visiteur devrait être limitée, car si le nombre des agents qui s'exécutent est amplifié, le serveur risque d'avoir une dégradation de performance et donc un épuisement de mémoire. Pour éviter cette situation, l'« *Agent-Serveur* » surveille et contrôle l'expiration de la date limite de résidence permise pour chaque agent visiteur. Cette durée est définie par l'administrateur du système (à titre d'exemple 24 heures). Si l'« *Agent-Serveur* » détecte une expiration d'un agent donné, il l'efface et informe l'utilisateur via l'« *Agent-Messages* ».

- **Agent-Offres :**

Après avoir reçu un « *Agent-Visiteur* », l'« *Agent-Offres* » commence à chercher des offres possédant des caractéristiques similaires à la demande de l'« *Agent-Visiteur* ». Cette recherche est effectuée en se basant sur l'algorithme de CBR.

Dès que l'« *Agent-Visiteur* » reçoit une offre similaire, il commence à chercher les produits dans les bases de données des marchands existants dans le marché virtuel. Mais

il est probable que l'« *Agent-Visiteur* » ne trouve pas de résultats de l'offre proposée (produits épuisés) alors l'agent demande une autre offre à l'« *Agent-Offres* » qui, à son tour propose une autre offre mais de niveau de similarité inférieure. Ce processus se répète jusqu'à que l'« *Agent-Visiteur* » trouve des résultats ou bien l'« *Agent-Offres* » n'a plus d'offres à proposer. À noter que l'« *Agent-Serveur* » sera informé par l'« *Agent-Visiteur* » à propos des résultats de sa recherche. Au cas où l'« *Agent-Visiteur* » n'a pas trouvé sa demande, l'utilisateur est informé plus tard par courrier électronique dès que le produit sera disponible. Cette dernière tâche est une des responsabilités de l'« *Agent-Messages* ».

- ***Agent-Messages*** :

La tâche principale associée à cet agent est d'informer l'utilisateur de tout événement relié à l'état de son agent. Cette communication est faite directement par le *courrier électronique*. Rappelons que l'agent a deux possibilités pour terminer sa recherche : l'établissement de la connexion et la déconnexion avec l'utilisateur. Dans le premier cas, l'« *Agent-Visiteur* » retourne à sa machine de départ et la tâche est terminée tout simplement. Dans le deuxième cas, l'« *Agent-Visiteur* » a accompli la mission de recherche. Après un certain délai, l'« *Agent-Messages* » informe l'utilisateur de l'état de terminaison de la tâche de son agent. Si l'utilisateur ne s'est pas remis en connexion (ce qui est le cas) pour récupérer son agent, ce dernier sera effacé du marché après le dépassement d'une certaine date fixée par l'administrateur. À ce moment, l'« *Agent-Messages* » informe aussi l'utilisateur de la suppression de son agent.

De plus, l'« *Agent-Messages* » a une mission de veille sur les bases de données afin de surveiller tout changement qui peut être appliqué sur ces données. En cas d'échec d'une recherche d'un agent donné, quand l'« *Agent-Messages* » détecte la disponibilité des produits en recherche, il informe l'utilisateur de cette nouvelle disponibilité. En conséquence, l'utilisateur peut répéter le processus de recherche en suivant les étapes déjà mentionnées.

Outre les agents, le système Pacha est composé de :

- une interface dédiée à l'administrateur permettant à ce dernier de gérer tous les agents du marché virtuel. Elle permet aussi de visualiser les résultats des activités entreprises par les agents visiteurs.
- une couche Java RMI nécessaire à la connexion des deux parties principales du système, de plus qui permet aux usagers de contrôler et de visualiser les activités de leurs agents dans le marché virtuel. Cette couche est décrite à la section (5.4.2).

À noter que Pacha fournit en plus une interface graphique pour aider le marchand à contrôler et manipuler sa base de données dans le marché virtuel.

## 5.2 Protocoles de communication

Dans cette section, nous présentons trois protocoles pour aider à clarifier le cheminement de l'information afin d'illustrer l'interaction et la coopération entre les différents composants. Le premier protocole sert à illustrer le mécanisme dont *Pacha* dispose pour conseiller et aider l'utilisateur à bien choisir sa demande. Ensuite, le deuxième nous donne une image claire sur la situation des agents qui voyagent au marché virtuel et la façon dont ils réagissent et communiquent avec les autres agents disposés par le marché virtuel. Enfin, le troisième présente le mécanisme de communication du *Pacha* avec l'utilisateur qui a envoyé son agent au marché où la connexion n'existe pas entre les deux. Ce dernier protocole donne également une idée de la façon dont le serveur contrôle et supervise les agents visiteurs en supprimant ceux qui ont terminé leur mission et ont dépassé une date limite spécifiée.

## 5.2.1 Processus pour conseiller l'utilisateur

L'objectif principal de ce processus est de donner une idée sur la technique adoptée par Pacha pour conseiller et orienter l'utilisateur afin de bien choisir les caractéristiques du produit à chercher. La Figure 16 nous montre le modèle d'interaction entre les différents acteurs afin de choisir un produit.

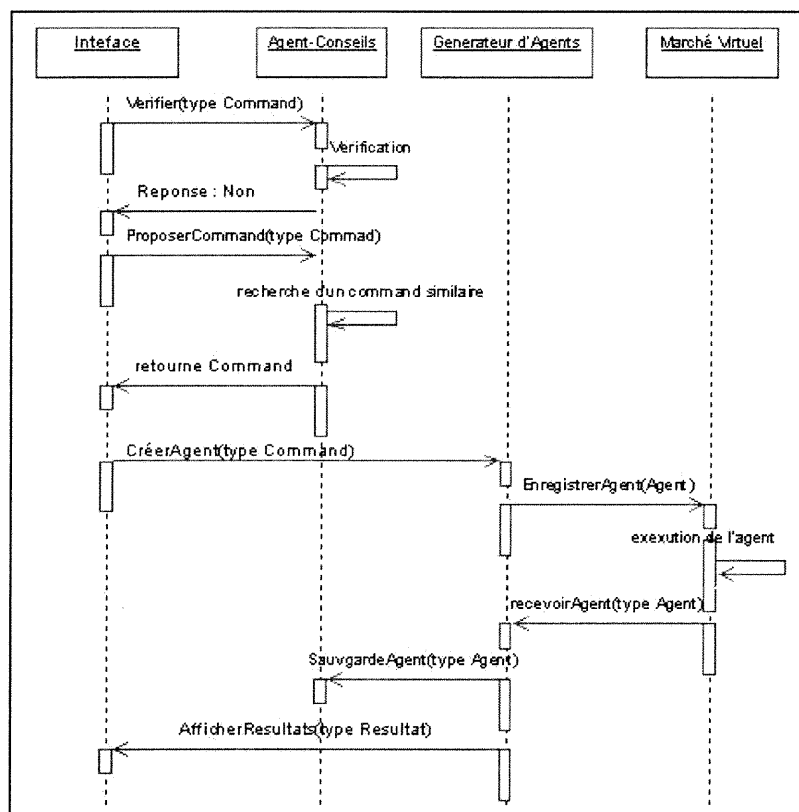


Figure 16 : Conseil et choix d'un produit

Le point de départ est l'objet « *Interface* » qui prend les préférences de l'utilisateur pour créer une instance de type *Command*. Ensuite, pour vérifier la validité de ce *Command*, l'« *Agent-Conseils* » dispose d'une méthode « *Vérifier (type Command)* ». Cette méthode parcourt toutes les commandes déjà sauvegardées dans le profil de l'utilisateur. Elle



retourne la valeur *Faux* si la commande existe déjà mais sans avoir abouti à des résultats. Elle retourne *Vrai* dans les autres cas. Si la réponse est négative, l'« *Interface* » demande à l'« *Agent-Conseils* » de lui proposer une autre commande similaire à celle déjà choisie. À ce moment, l'« *Agent-Conseils* » fournit la méthode « *Proposer (type Command)* » pour chercher d'autres commandes similaires ayant obtenu des résultats. En général, l'« *Interface* » peut choisir entre les commandes proposées ou bien insister sur son premier choix (algorithme 1).

---

#### **Algorithme 1** Proposer (Commande)

---

**Begin**

*TableauCommandes* = *ChercherCommandes*(*Commande.Nom*())

**for all** *CommandeTest* ∈ *TableauCommandes* **do**

*Distance* = *EvaluerDistance*(*Commande*, *CommadeTest*)

**if** *Distance* > *seuil* **then**

*TableauCommandes.EffacerCommande*(*CommadeTest*)

**end if**

**end for**

*TrierCommandes* (*TableauCommandes*)

**return** *TableauCommandes*

**End**

---

L'« *Interface* » envoie la commande au « *Générateur-Agents* » en lui demandant de préparer l'agent au *marché virtuel*. Pour réaliser cette tâche, le « *Générateur-Agents* » fournit la méthode « *CréerAgent (type Command)* ». Pour transmettre cet agent, le « *Générateur-Agents* » appelle la méthode « *EnregistrerAgent (type Agent)* » fournie par le marché virtuel à distance. Cette méthode sert à enregistrer l'agent dans le marché, pour qu'il puisse s'exécuter et chercher le produit demandé. Si la mission de l'agent est terminée, alors le marché livre l'agent qui sera reçu par le « *Générateur-Agents* » au moyen de la méthode « *RecevoirAgent (type Agent)* ». Le « *Générateur-Agents* »

demande à l'agent de conseil de sauvegarder la commande et les résultats de l'agent dans le profil de l'utilisateur. Ces résultats peuvent être utilisés dans d'autres commandes ultérieures. Enfin, le « Générateur-Agents » envoie les résultats de l'agent à l'« Interface » pour les afficher, en utilisant la méthode « *AfficherResultats (type Resultats)* ».

## 5.2.2 Processus de recherche d'un produit

La (Figure 17) nous montre l'interaction entre les différents objets qui se trouvent dans le marché virtuel.

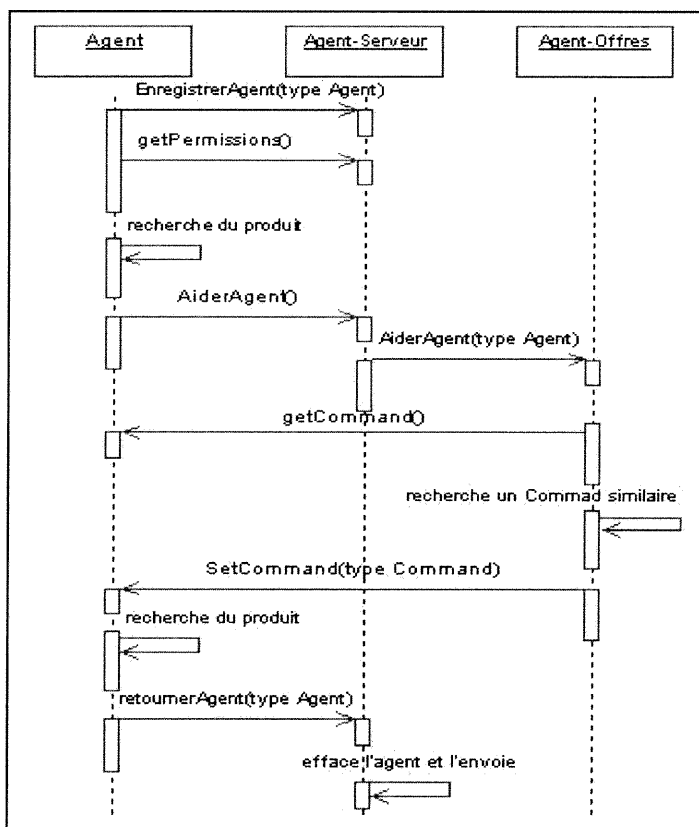


Figure 17 : Recherche d'un produit

Quand l'Agent « *Agent-Visiteur* » arrive au site du marché, il appelle la méthode « *EnregistrerAgent (type Agent)* » fournie par l' « *Agent-Serveur* » pour enregistrer les agents visiteurs. L'Agent demande à l' « *Agent-Serveur* » la permission d'accéder aux bases de données des marchands. Après avoir obtenu les permissions nécessaires, il commence à chercher le produit demandé dans les bases de données des marchands.

Si les données en question ne sont pas disponibles, l'Agent demande l'aide de l' « *Agent-Serveur* » en appelant la méthode « *AiderAgent ( )* » fournie par l' « *Agent-Serveur* ». Cette méthode permet la communication entre cet Agent et l' « *Agent-Offres* » via la méthode « *AiderAgent (type Agent)* » de l' « *Agent-Offres* ». Suite à l'appel de cette méthode, l' « *Agent-Offres* » demande à l'agent la commande recherchée, en utilisant la méthode « *getCommand ( )* » et procède pour trouver un objet ayant les caractéristiques les plus proches à celles de l'objet spécifié dans la commande initiale (à voir : algorithme 1 dans la section 4.1.8). Ainsi l'Agent prend en considération les suggestions de l' « *Agent-Offres* » et commence une autre session de recherche. Tant que l'Agent n'a pas trouvé de résultats, il demande d'autres solutions à l' « *Agent-Offres* ». Finalement, quand l'Agent est satisfait des résultats de ses recherches, il appelle la méthode « *RetournerAgent ( )* » offerte par l' « *Agent-Serveur* » qui sauvegarde des informations sur l'activité de l'Agent et ensuite, l'envoie à son point de départ.

### 5.2.3 Processus de notification de l'utilisateur

Pour bien illustrer ce protocole, nous présentons deux figures qui nous permettent la visualisation et le déroulement de communication entre l' « *Agent-serveur* » et l' « *Agent-Messages* », lors d'une connexion entre l'utilisateur et le marché. Souvenons-nous, que le serveur est en face de deux possibilités lors d'une déconnexion.

La (Figure 18) explique la première possibilité où la connexion avec l'utilisateur est rétablie sans l'expiration de la date limite de résidence de l'agent visiteur dans le marché virtuel.

Le point de départ, l'« *Agent-Serveur* » demande à l'« *Agent-Visiteur* » sa date d'arrivée via la méthode « *retourneDate()* » fournie par l'« *Agent-Visiteur* ». Suite à la réception de cette date, l'« *Agent-Serveur* » vérifie si cet agent excède la date limite de résidence permise. Dans le cas où l'« *Agent-Visiteur* » ne dépasse pas cette date, l'« *Agent-Serveur* » demande à l'« *Agent-Messages* » d'avertir l'utilisateur à propos de la fin de la mission de son agent. Cette avertissement est faite via la méthode « *avertiAgent(type Adresse)* » fournie par l'« *Agent-Messages* ». En conséquence, l'utilisateur reçoit un e-mail à ce propos.

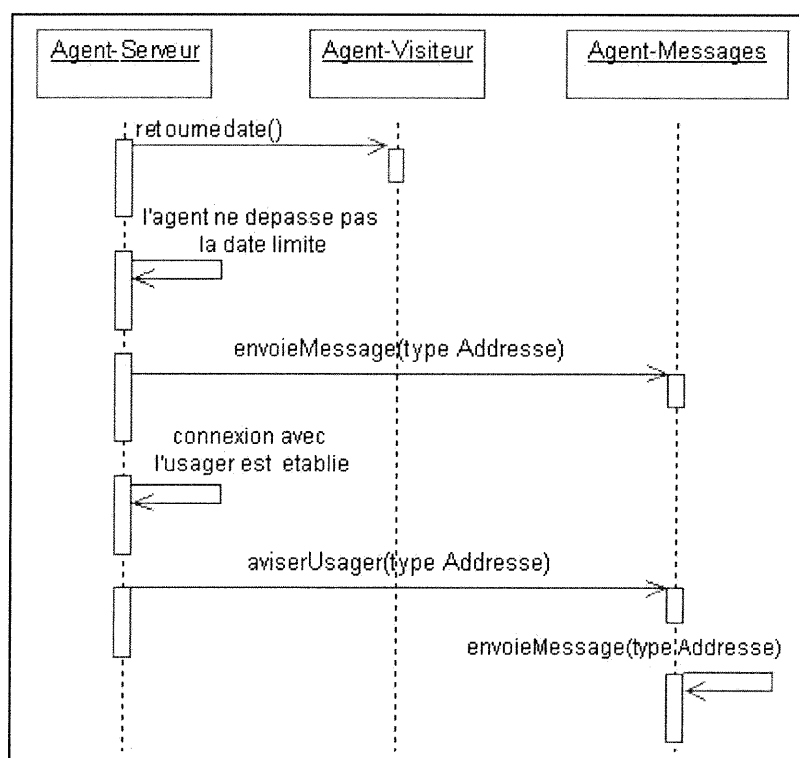


Figure 18 : Avertissement de l'utilisateur

Dans cette figure, nous supposons que l'utilisateur répond à cet e-mail en établissant la connexion avec le serveur du marché. Dans ce cas l'« *Agent-Serveur* » retourne l'agent à sa machine de départ.

La (Figure 19) simule le cas où l'utilisateur n'a pas rétabli la connexion après l'avertissement de l'« *Agent-Messages* ».

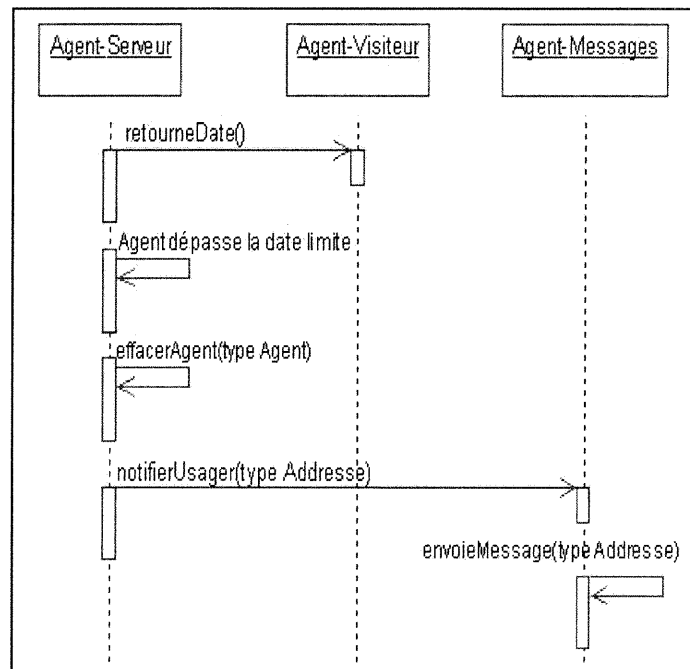


Figure 19 : Notification de l'utilisateur.

Ce dernier détecte à un moment donné l'expiration de l'agent en question, alors il l'efface en utilisant la méthode « *effacerAgent(type Agent)* » et également demande à l'« *Agent-Messages* » d'aviser l'utilisateur à propos de cette action et l'état final de son agent.

### 5.3 Déroulement de Pacha

Cette section présente le scénario d'interactions entre les différents composants du système Pacha (« *Partie-Client* » et « *Marché-Virtuel* »). Elle a pour objectif de donner une idée du déroulement et du fonctionnement du système. Suite à ce que nous avons présenté dans les sections précédentes, nous procédons à expliquer le déroulement du système en utilisant un exemple concret. Cet exemple induit une séquence d'étapes à réaliser des deux côtés client et serveur. Dans les prochaines sections, nous identifions ces étapes avec des illustrations des interfaces qui représentent notre système.

### 5.3.1 Côté client (choix des produits et transmission des agents)

Un usager désire acheter par exemple un processeur spécifique. Avec notre système, l'usager peut tout simplement déléguer à un agent « A » la responsabilité de trouver la liste des processeurs correspondant aux caractéristiques spécifiées par l'usager. Pour construire l'agent A, l'usager choisit la tâche de recherche d'un processeur (Figure 20).

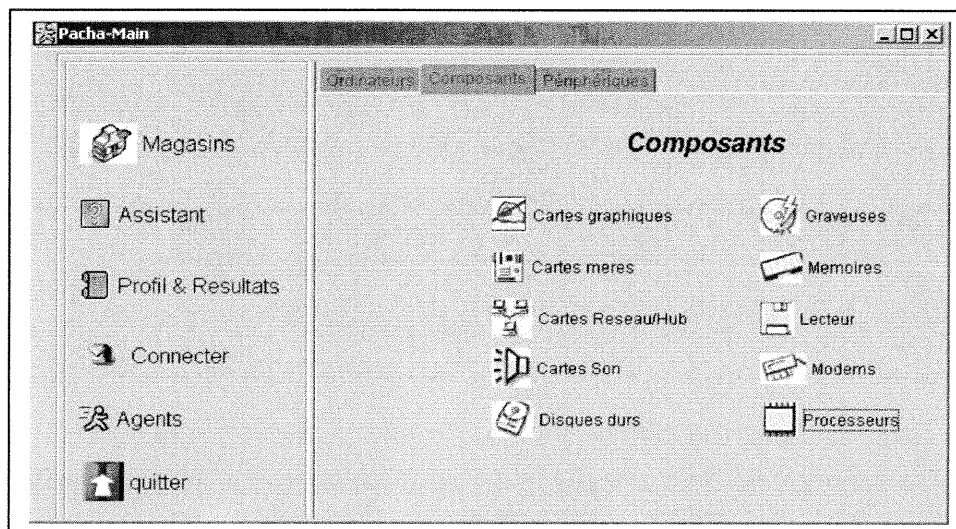


Figure 20 : L'usager choisit la tâche de son agent.

Ensuite il commence à compléter les formes appropriées décrites plus tard dans cette section. À titre d'exemple, ces formes peuvent contenir les propriétés d'un processeur (type, cadence, budget maximal) (Figure 21). A noter que l'usager peut toujours consulter les magasins qui sont disponibles au marché virtuel. Toutes les commandes et leurs résultats seront sauvegardés dans le profil de cet usager qu'il peut, à son tour, les consulter pour des utilisations ultérieures. Enfin, l'usager peut toujours interrompre la connexion avec le serveur durant le choix de sa commande.

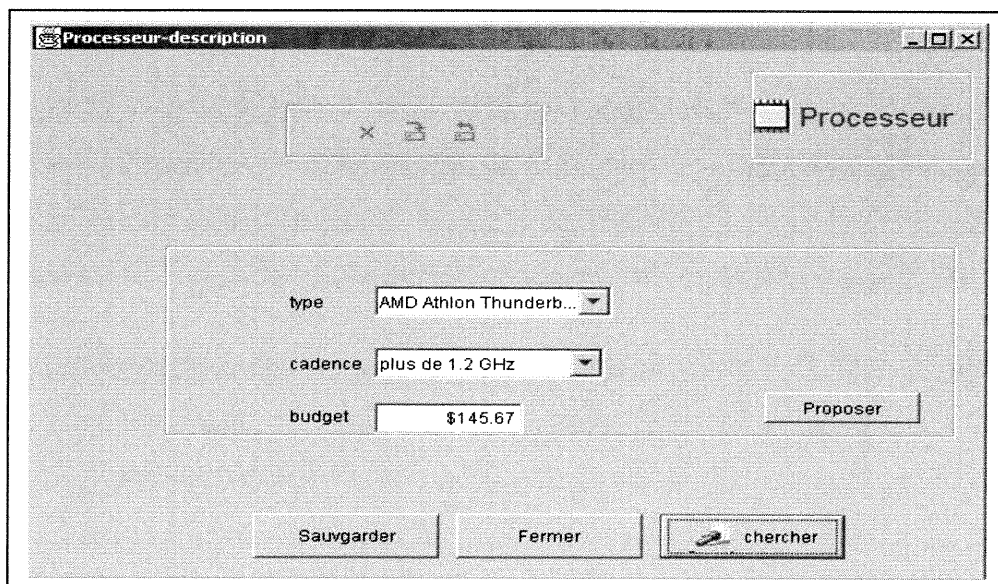


Figure 21 : L'utilisateur définit ses préférences.

Après avoir choisi les caractéristiques du processeur, les caractéristiques de cet item seront toujours vérifiées et validées par le système qui, à ce moment, avertit l'utilisateur au cas où son choix est déjà sélectionné mais n'a pas abouti à des résultats (Figure 22). L'utilisateur peut sélectionner un processeur ou bien maintenir à son premier choix.

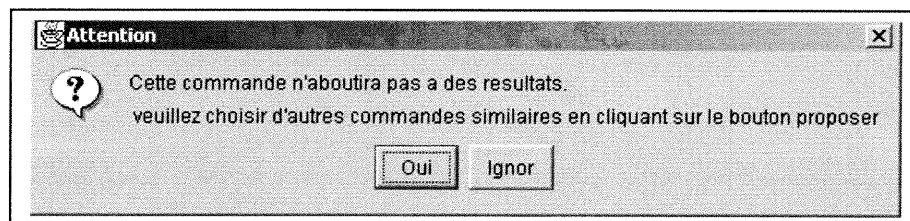


Figure 22 : Vérification de la commande.

Dans le premier cas, le système se charge de présenter un ensemble d'items plus proches de celles que l'utilisateur vient de sélectionner (Figure 23). À noter que cet ensemble de processeurs est trié selon le niveau de similarité par rapport au processeur en question.

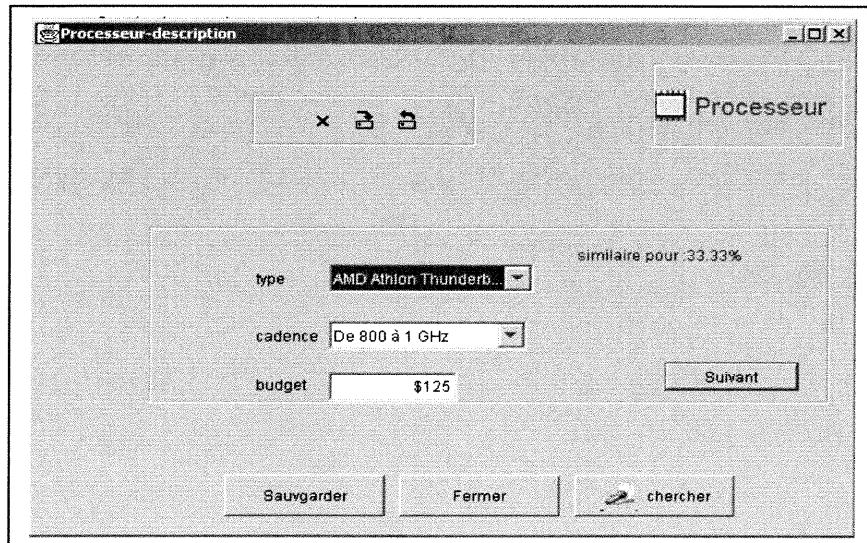


Figure 23 : Item similaire

Avant le lancement de l'agent, l'utilisateur se trouve devant une autre forme (Figure 24) qui lui demande de sélectionner d'autres préférences. Ces dernières seront les tâches spécifiées à l'agent qui va voyager au marché.

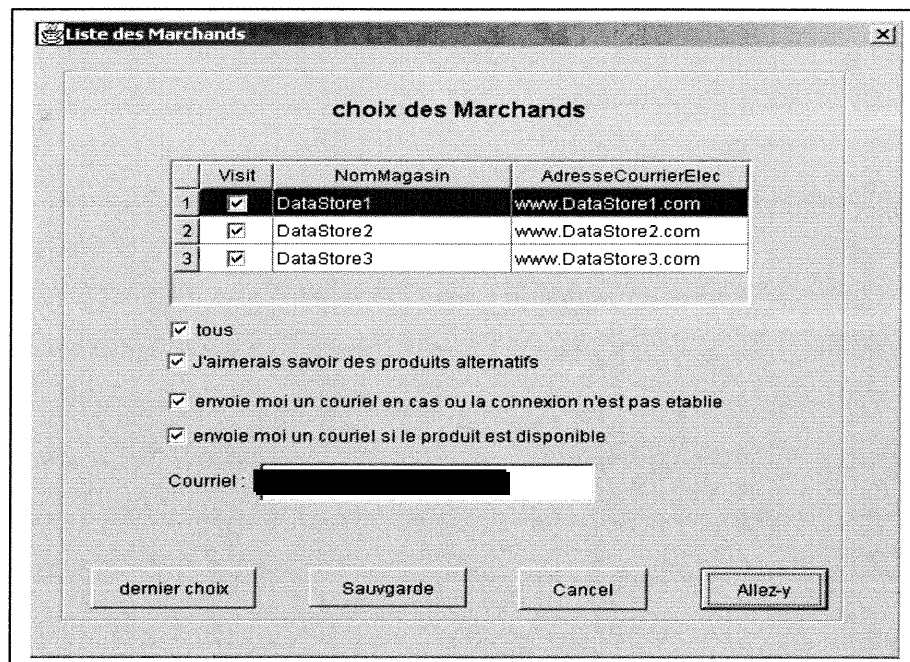


Figure 24 : Liste des marchands.



À partir de cette forme, l'utilisateur a plusieurs choix :

- choisir les marchands dans lesquels son agent doit chercher les items ;
- accepter des produits alternatifs ;
- recevoir des courriers électroniques concernant les produits disponibles ou bien l'état de son agent à distance.

Dès que l'utilisateur choisit son item, il peut alors demander de créer et d'envoyer l'agent au marché virtuel (bouton : « *Allez-y* »). Le système prépare et lance l'agent A en tenant compte des préférences de l'utilisateur. Il est important de noter qu'une fois que l'agent est lancé, l'utilisateur peut effectuer d'autres tâches comme par exemple, débrancher ou éteindre sa machine, fermer le système ou annuler tout simplement la tâche de l'agent. Cependant l'utilisateur peut à tout moment se reconnecter au marché virtuel pour poursuivre l'activité de ses divers agents qui sont déjà envoyés.

### **5.3.2 Côté serveur (interaction des agents dans le marché virtuel)**

Cette étape explique et illustre comment les agents visiteurs s'exécutent et interagissent avec d'autres agents dans le marché virtuel. L'objectif de cette interaction est d'aider les agents visiteurs à trouver des données pertinentes. Étant donné un agent A qui arrive au marché virtuel cherchant un produit qui n'est pas disponible, il entame une coopération avec les agents du marché virtuel. Suite à cette coopération, les agents lui offrent un autre produit similaire. La (Figure 25) illustre le processus de cette négociation affiché sur le serveur.

*AgentServeur : Agent A est inscrit dans le marché virtuel.*

*AgentServeur : Agent A prend les permissions aux bases des données des marchands.*

*Agent A : je parcours les bases des données des marchands afin de trouver mon produit.*

*Agent A : j'en ai pas trouvé des résultats, alors je demande l'aide de l'AgentServeur.*

*AgentServeur : je demande à l'Agent-Offres d'aider l'agent A.*

*Agent-Offres : je prends les caractéristiques de la commande de l'agent A, et je cherche les commandes similaires.*

*Agent-Offres : je trouve un Processeur similaire à 66.67%.*

*Agent A : j'essaye cette commande.*

*Agent A: je trouve des résultats.*

*AgentServeur : je n'ajoute pas cette commande dans la base de données des offres car cette commande existe déjà.*

*AgentServeur : je sauvegarde quelques informations de cet agent comme (date d'arriver, date départ, type de produit, quantité trouvée,...).*

*Agent A : Ok, j'ai fini et la connexion avec l'utilisateur est établie alors je pars.*

*AgentServeur: Agent A est enlevé du marché virtuel et il est retourné à sa station de départ.*

Figure 25 : Scénario de négociation avec les agents du marché virtuel.

Rappelons-que, si l'utilisateur ne s'est pas remis en attente de l'agent A, cet agent peut l'attendre dans le marché virtuel. Quand la connexion est rétablie, l'agent A retourne à la machine de départ en apportant les résultats obtenus à l'utilisateur.

Les résultats obtenus suite au processus de recherche par l'agent, sont affichés à l'utilisateur dans une fenêtre spécifiant les descriptions de produit trouvé, l'adresse électronique du marchand qui a ce produit et le pourcentage de niveau de similarité dans le cas d'un produit alternatif (Figure 26). Ces résultats sont triés selon leur prix. Cependant, il faut mentionner que ces résultats seront enregistrés dans le profil de l'utilisateur pour une consultation future comme nous l'avons décrit dans les sections précédentes.

Résultats est environ : 66.67%

	resultatdesc	resultatdetails	resultatmarchand	resultatprix
1	INTEL Celeron ii : plus	Processor--> & type : INTEL Ce	www.Store1.com	700
2	INTEL Celeron ii : plus	Processor--> & type : INTEL Ce	www.Store2.com	725
3	INTEL Celeron ii : plus	Processor--> & type : INTEL Ce	www.Store3.com	750

OK

Figure 26 : Résultats obtenus par l'agent.

### 5.3.3 L'interface de l'administrateur

Pacha offre à l'administrateur du système une interface permettant de réaliser deux tâches principales: gérer les agents visiteurs du marché virtuel et aussi, visualiser une liste des événements faite par les agents visiteurs (Figure 27).

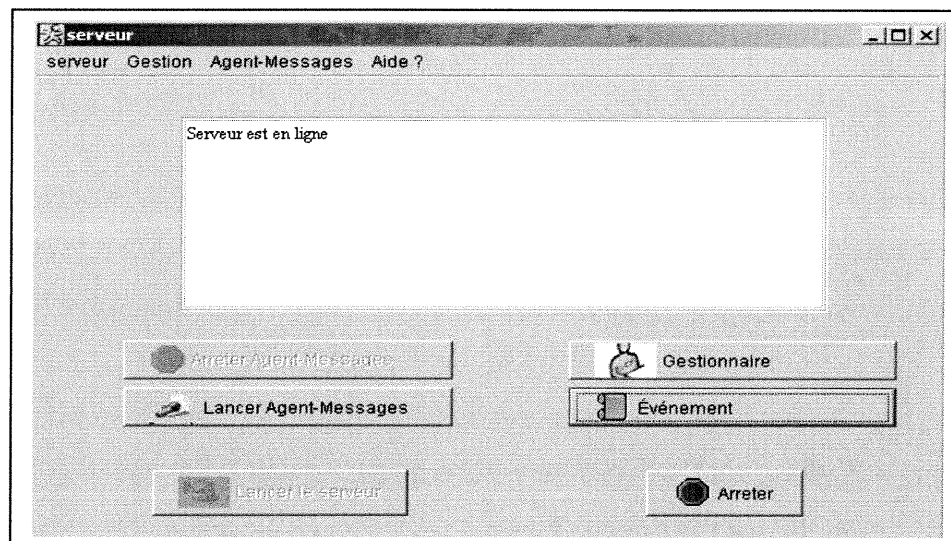


Figure 27 : Interface principale de l'administrateur.

La gestion des agents visiteurs est la responsabilité de l'administrateur du marché (Côté serveur). Pour cette fin Pacha offre une interface graphique permettant de visualiser les agents actifs dans le marché virtuel en affichant quelques informations sur chaque agent dans le marché (telles que IP adresse de la machine de départ, date de résidence, etc.). Ainsi, pour des raisons de performance, l'administrateur est capable de vérifier le statut de tous les agents existants (en exécution, en attente, etc.) et il peut toujours effacer un agent dans une situation critique.

Cependant, le système fournit une autre tâche permettant à l'administrateur en tout temps d'intervenir pour éteindre ou démarrer le serveur.

Du point de vue conceptuel, notre système peut avoir un point vulnérable si un usager malhonnête envoie de nombreux agents simultanément au système. Dans ce cas, le système pourrait avoir une forte dégradation au niveau de la performance. Pour éviter cette situation, l'administrateur peut superviser les sources des agents existants (Figure 28). À un moment donné, s'il trouve une telle problématique, il peut toujours bloquer l'adresse source des agents en empêchant l'usager émetteur de se reconnecter au système.

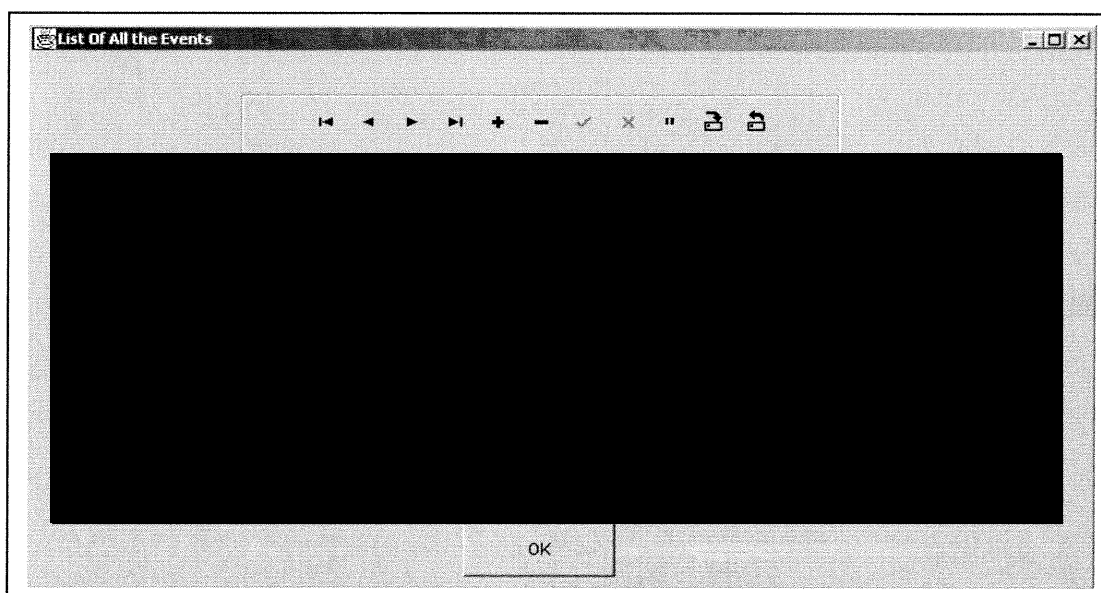


Figure 28 : liste des événements.

## 5.4 L'environnement d'exécution

Toute machine hôte (serveur) qui accueille un agent doit disposer d'un environnement d'exécution qui permette aux agents de s'exécuter et d'accéder aux différents services de l'hôte, par exemple à une base de données ou à un algorithme de cryptage. L'environnement d'exécution pose le problème de sa cohérence sur le réseau. Il nécessite une grande homogénéité des configurations logicielles sur chacun des hôtes. Même si un standard existait, il faudrait encore gérer les différences entre les versions. Dans le domaine de l'interopérabilité entre les agents, il existe trois standards : Mobile Agent System interoperability Facility (MASIF) de l'OMG (l'organisme qui gère la norme CORBA), Knowledge Query and Manipulation Language (KQML) et Foundation of Intelligent Physical Agents (FIPA).

Si l'agent est un exécutable (par exemple du langage C compilé), il faudra en plus gérer l'homogénéité des plates-formes matérielles et des systèmes d'exploitation. *Java* et les langages interprétés ne souffrent évidemment pas de cette contrainte.

### 5.4.1 Implémentation

Nous avons entièrement développé le système Pacha en Java 2 (JDK 1.3). L'un des avantages majeurs de ce choix est la portabilité apporté par Java, caractéristique indispensable d'un système qui serait implanté dans un environnement potentiellement hétérogène. De plus, *Java* possède tous les outils nécessaires pour la construction d'agents intelligents et autonomes [Bigus, 98] et ce dans une trousse de développement complète, gratuite et éprouvée. L'un de ces outils est Java RMI qui a été utilisé comme méthode de communication.

### 5.4.2 RMI

Le système “Remote Methode Invocation” (RMI) de Java permet à un objet exécuté sur une machine virtuelle Java (JVM) d’invoquer des méthodes d’un objet exécuté sur une autre JVM. RMI procure donc un moyen de communication dans un système d’objets distribués utilisant le langage Java. Du point de vue du programmeur, RMI permet par exemple à des objets d’une application serveur tout comme s’il s’agissait d’invocations de méthodes locales, en camouflant toute la mécanique nécessaire au transfert des données (paramètres et résultats) à travers le réseau.

Une habileté unique de RMI, constituant sa principale force, est de permettre à une application de télécharger le code d’un objet d’une autre application en cours d’exécution [woltrath, 00]. Ainsi, les types de données et les comportements, au départ disponibles seulement sur une application, peuvent être transmis à une autre application (pouvant se trouver sur une autre machine virtuelle, distante) et de cette façon, en accroître les capacités dynamiquement. La « *mobilité* » des agents de notre système est donc réalisée de manière conceptuelle, les agents étant des objets pouvant, grâce à RMI, voyager avec les données et ainsi modifier leur comportement dynamiquement.

Finalement, les bases de données utilisées pour stocker toutes les informations sur les produits, les commandes, les agents et aussi le profil de l’usager sont des bases de données relationnelles “JDataStore” fournie par “ Borland ” et reliées à notre application via le driver “com.borland.datastore.jdbc.DataStoreDriver”. “JDataStore” a beaucoup d’avantages : d’abord elle est écrite en Java, compatible à SQL et accécible via JDBC. À noter que cette base de données est intégrée avec “JBuilder”.

### 5.4.3 Sécurité

L’environnement d’exécution doit aussi se protéger des attaques, par exemple d’agents virus. Il nous faut donc un mécanisme d’authentification des agents pour gérer les droits

qu'ils auront sur l'environnement. Cela permettra notamment à un hôte d'interdire l'accès à certaines informations pour tous les agents n'appartenant pas à un groupe autorisé.

Pour répondre à cette question de sécurité, RMI offre un gestionnaire spécial de sécurité et le chargeur de classes "*Class Loader*". Ce dernier, nous donnent l'opportunité de protéger le marché virtuel contre les agents visiteurs. L'administrateur de serveur peut définir des politiques d'exécution de sécurité telles que l'accès aux fichiers en mode lecture/écriture et aux ports du contact, etc. ( Figure 29).

```
grant {
  /* Socket Permissions */
  permission java.net.SocketPermission "*:1024-65535", "connect,accept,listen";
  permission java.net.SocketPermission "*:80", "connect";

  /* Utility Permissions */
  permission java.util.PropertyPermission "*", "read";
  permission java.util.PropertyPermission "java.library.path", "write";

  /* Runtime Permissions */
  permission java.lang.RuntimePermission "loadLibrary.*";

  /* File Permissions */
  permission java.io.FilePermission "/pacha/DBs/*", "read";
  permission java.io.FilePermission "/pacha/Info1/*", "read";
};
```

Figure 29 : Exemple de configuration du gestionnaire de sécurité.

Après avoir configuré notre gestionnaire de sécurité, les agents visiteurs ont seulement la permission de lire les bases de données existantes dans le marché virtuel.

## 5.5 Conclusion

Nous avons présenté et expliqué en détails les différentes composantes de notre système. Nous avons présenté l'algorithme clé et décortiqué les modules principaux : Conseil, communication, Offres et environnement de négociation. Le prochain chapitre vise à déterminer si nos objectifs ont été atteints ou non. Ce dernier chapitre ouvrira finalement la porte sur les possibilités futures de notre architecture et sur les améliorations qui pourraient lui être apportées.



## *Chapitre 6*

### **6 Conclusion et perspectives**

#### **6.1 Conclusion**

La technologie du commerce électronique a toujours été en constante évolution mais celle-ci est particulièrement fulgurante depuis l'arrivée de l'Internet qui joue un rôle important face aux applications coopératives sur le Web. Au fur et à mesure que la technologie Internet se développe tout en devenant un canal principal pour la vente et l'achat des biens et des services, le besoin d'accroître l'efficacité des processus devient primordial. Unaniment, on s'entend pour dire que l'utilisation de la notion des agents intelligents améliore le traitement des transactions dans les applications reliées au domaine du commerce électronique. En dépit de la croissance des travaux traitant sur l'utilisation des agents, peu réussissent à conceptualiser et à implanter un système *multi-agents* intelligent. Malheureusement, on note que dans les travaux qui pourraient s'avérer intéressants, le niveau d'intelligence associé à leurs systèmes est modéré. Ce manque d'intelligence se traduit par la difficulté, au niveau du consommateur, de choisir un produit dû au manque de connaissances requises pour ce faire. Par conséquent, ceci entraîne pour le vendeur l'impossibilité d'offrir des produits et des services personnalisés et adaptés aux besoins des consommateurs.

La problématique décrite plus haut a grandement motivé l'élaboration de notre système dont l'objectif initial était de conceptualiser et d'implanter un système *multi-agents* intelligent et mobile. D'une part, nous voulions développer un système qui tient compte du niveau d'expertise de l'utilisateur afin de lui fournir une aide pertinente: accessible pour les uns et spécialisée pour les autres. Ce processus permet aussi au marchand de personnaliser leur offre afin de combler les besoins réels du client. D'autre part, nous tenions à mettre un accent sur la mobilité des agents de manière à ce que l'utilisateur puisse se déconnecter pendant que l'agent s'occupe de sa requête.

Les trois propriétés qui caractérisent notre système (décrit en détails au chapitre 4) sont directement liées aux objectifs que nous nous fixions: intelligence, multi-agents, autonomie. La première caractéristique rend notre système capable de donner aux utilisateurs des conseils relatifs à leurs besoins et de les orienter selon leur niveau d'expertise. D'autre part, cette propriété permet au marchand de personnaliser une offre similaire lorsque les données demandées ne sont pas disponibles. Ce processus est effectué via des agents appliquant l'algorithme CBR ("*Case Based Reasoning*"). La deuxième propriété, multi-agents, consiste en une plate-forme appelée marché virtuel permettant d'accueillir des agents visiteurs référés par les utilisateurs. Ce marché est composé de plusieurs agents qui interagissent et collaborent ensemble afin de guider un agent visiteur et l'aider à trouver les produits recherchés. La troisième propriété, l'autonomie, différencie vraiment notre système des systèmes actuels. Les tests effectués sur Pacha montrent bien que les agents de notre système détiennent un haut degré d'autonomie. Une perte de connexion entre l'utilisateur et le serveur n'interrompt point le traitement des tâches associées aux agents qui sont envoyées. En effet, l'autonomie qui démarque ces agents leur permet de continuer la phase de recherche sans avoir à demeurer en connexion avec leur source. Les tests montrent que cette caractéristique permet à l'utilisateur de récupérer les résultats de recherche après un rétablissement de la connexion.

Avec ces trois propriétés nous pouvons conclure que Pacha rencontre bien les objectifs initiaux. Malgré le fait que ce dernier soit un système pilote mené dans le cadre d'un sujet

de recherche, nous croyons qu'il est envisageable de l'améliorer pour une utilisation efficace dans des situations réelles.

## 6.2 Perspectives

Bien entendu, plusieurs améliorations pourraient être apportées à Pacha. L'architecture globale du système étant générique, le support de plusieurs marchés virtuels dans lesquels les agents pourront voyager est toujours possible. Nous pourrions éventuellement avec une modification simple envisager l'utilisation de ce système dans d'autres contextes où le marché virtuel consisterait en une salle d'enchère, vente, etc. Nous croyons également qu'une amélioration de la mesure de sécurité est indispensable. Finalement, il est possible de perfectionner notre système servant à orienter l'utilisateur selon son niveau d'expertise. Nous demeurons cependant ouverts à d'autres concepts tels les agents réactifs.

## 7 Bibliographie

- [Aïmeur, 00] [Aïmeur and Vezeau, 2000] Aïmeur, E., and Vezeau, E.: *Short-Term Profiling for a Case-Base Reasoning*, proceedings of the 11th European Conference on Machine Learning, Barcelona, Catalonia, Spain, pages 23-30 May/June 2000.
- [Aha01] Aha, D.W., Breslow L.A., and Muñoz-Avila H.: *Conversational case-based reasoning*. Applied Intelligence, pages 9-32, 2001.
- [Amazon, 02] Amazon: <http://www.amazon.com>. [Last access at August 2002].
- [Anthony, 96] Anthony C. and Pattie M. Kasbah: *An agent marketplace for buying and selling goods*, in Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, pages 30-40, April 1996.
- [Alf, 02] Alf IngeWang: *Using JavaSpaces to implement a mobile Multi-Agent System*, Dept. of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2002.
- [Bergmann, 99] Bergmann, R., Schmit S.: *Product Customization in an Electronic Commerce Environment using Adaptation Operators*, in Proceedings of the 7th German Workshop on Case-Based Reasoning, Würzburg, pages 25-30, March 1999.
- [Bigus, 98] Bigus, J.P.: *Constructing Intelligent Agents Using Java*. John Willey & Sons, Inc., 1998.
- [Cabri, 98] Cabri G., Leonardi L., and Zambonelli V.: *The Impact of the Coordination Model in the Design of Mobile Agent Applications*, the Twenty-Second Annual International Computer Software and Applications Conference, Vienna, Austria, August 1998.

- [Christoph, 95] Christoph Globig, Stefan Wess: *Learning in Case-Based Classification Algorithms*, in Workshop on Algorithmic Learning for Knowledge-Based Systems, kaiserslautern, Germany, 1995.
- [Coen, 94] Coen M. H., Sodabot: *A Software Agent Environment and Construction System*, MIT AI Technical report 1493, June 1994.
- [Collins, 98] Collins J., Tsvetovat M., Mobasher B. and Gini M.: *MAGNET: A Multi-Agent Contracting System for Plan Execution*, in Workshop on Artificial Intelligence and Manufacturing, State of the Art and State of Practice, AAAI Press, Albuquerque, NM, pages 63-68, August 1998.
- [eBay, 02] eBay, Available at <http://www.eBay.com>. [Last access at August 2002].
- [Forrester, 02] Forrester Technographics:  
<http://www.forrester.com/er/press/release/0,1769,177,ff.html>. [Last access at August 2002].
- [Gunghao, 98] Gunghao YAN, Wee-Keong NG, Ee-Peng LIM: *Toolkits for a distributed, agent-based web commerce system*, IFIP Conference, 1998.
- [Guttman, 98a] Guttman R. and Maes P.: *Agent-mediated Integrative Negotiation for Retail Electronic Commerce*, in Workshop on Agent Mediated Electronic Trading, May 1998.
- [Guttman, 98b] Guttman R., Moukas A., and Maes P.: *Agent-mediated Electronic Commerce: A Survey*, Knowledge Engineering Review, June 1998.
- [Harrison, 95] Harrison C. G., Chess D. M., and Kershenbaum A.: *Mobile Agents Are they a good idea?*, Technical report, IBM T. Watson Research Center, 1995.
- [Industry, 02] Industry Privacy Failures Hurting E-Commerce:  
[http://www.privacytimes.com/newwebstories/indus\\_priv\\_9\\_9.htm](http://www.privacytimes.com/newwebstories/indus_priv_9_9.htm). [Last access at August 2002].

- [Ismail, 98] Ismail L. et Hagimont D., *Spécialisation de serveurs par des agents mobiles, Colloque International sur les Nouvelles Technologies de l'information (NOTERE'98)*, Montréal, Canada, Novembre 1998.
- [J. E., 94] J. E. White, *Telescript Technology: The Foundation for the Electronic Marketplace*, General Magic White Paper, 1994.
- [Jango, 02] Jango. Available at <http://www.jango.com>. [Last access at August 2002].
- [Kuokka, 98] Kuokka D. and Harada L., *Matchmaking for Information Agents, Readings in Agents*, M. N. Huhns & M. P. Singh editors, pages 239-245, 1998.
- [Leake, 96] Leake, D. B.: *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/MIT Press, Menlo Park, CA. 1996.
- [Michael, 99] Michael V. and Stefan P.: *An agents-based Market supporting Multiple Auction protocols*, 4th International Conference on Electronic Business Engineering, 1999.
- [Michael, 00] Michael V., Stefan P.: *Using autonomous agents to expand business models in electronic commerce*, 2000.
- [MySimon, 02] MySimon Buyers Guides: <http://www.Mysimon.com>, 1999. [Last access at August 2002].
- [OCDE, 99] OCDE, *Définir et mesurer le commerce électronique, OCDE Workshop*, Avril 1999:  
[http://www.oecd.org/dsti/sti/it/ec/act/agenda\\_ECworkshop.htm](http://www.oecd.org/dsti/sti/it/ec/act/agenda_ECworkshop.htm).  
[Last access at August 2002].
- [Perkowitz, 97] Perkowitz M. and Etzioni O.: *Adaptative Web Sites: an AI challenge*, In 15 th International Joint Conference on Artificial Intelligence, Nagoya, Aichi, Japan, August, 1997.

- [PersonnaLogic, 02] PersonnaLogic. Available at <http://www.personaLogic.com>. [Last access at August 2002].
- [Priceline, 02] Priceline. Available at <http://www.priceline.com>. [Last access at August 2002].
- [Rakesh, 93] Rakesh A., Tomasz I., and Swami A. N.: *Mining Association Rules between Sets of Items in Large Database*,. SIGMOD Conference, 1993.
- [Robert, 01] Robert S. Gray, George Cybenko, David Kotz, Ronald A. Peterson and Daniela Rus: *D'Agents: Applications and Performance of a Mobile-Agent System*, Thayer School of Engineering / Department of Computer Science, Dartmouth College, Hanover, November 2001.
- [Shoham, 90] Shoham Y., *Agent-Oriented Programming*, Technical Report STAN-CS-90-1335, Stanford University, 1990.
- [Sierra, 99] Sierra C.: *Agent-mediated electronic commerce a European viewpoint*, In 7 th Journées Francophones pour l'Intelligence Artificielle Distribuée et les Systèmes Multi-agents, La Réunion, November 1999.
- [Turban,02] Turban, Efraim, *Electronic commerce: a managerial perspective*, Prentice Hall, pages 3-13, 2002.
- [Voyager, 97] Voyager Core Package: Technical White Paper. ObjectSpace Incorporated, March 1997.
- [Weiss, 99] Weiss G., Wooldridge M.: *Multi-Agent Systems: A modern approach to DAI*, June 1999
- [White, 95] White J., *Mobile Agents*, General Magic, Telescript Technical Whitepaper, October 1995.

- [Wolrath, 00] Wolrath A., Waldo J., *the java tutorial- RMI. Website*, Octobre 2000: <http://java.sun.com/docs/books/tutorial/rmi/index.html>. [Last access at August 2002].
- [Wurman, 98] Wurman P. R., Wellman M. P. and Walsh W. E., *The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents*, in 2nd International Conference on Autonomous Agents, Minneapolis, USA, May 1998.