

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

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : Intelligence artificielle

Sujet :

Un modèle de sécurité pour les
agents mobiles

Présenté par : BOUZIDI NAIMA
SALI HOUNIDA

Promoteur : D.E Menacer

Soutenu le: 21/10/2006, devant le jury composé de :

Aoussat
Benstiti
Boukhlef

Présidente
Examinatrice
Examineur



- 05/2005-2006-

MIG-004-123-1

Dédicace

*A ma raison de vivre, d'espérer,
à ma source de courage, à ceux que j'ai de plus cher:
mon papa, ma maman,
pour leur amour, leur confiance
et leur sacrifice sans limite,
mes deux adorables frères:
Abdelhakim et Abdelwaheb.
A ma sœur Nawel*

BOUZIDI

*A ma raison de vivre, d'espérer,
à ma source de courage, à ceux que j'ai de plus cher:
mon papa, ma maman,
pour leur amour, leur confiance
et leur sacrifice sans limite,
mes deux adorables frères:
Said et Rafik*

SALI



MIG-004-123-1

Remerciements

Nous tenons à remercier notre promoteur Djamel Eddine Menacer pour ses conseils.

A tous les enseignants de la faculté des sciences exactes de Blida et surtout les enseignants du département de l'informatique

Enfin, nous remercions, de tout cœur, tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Abstract

We talk about mobile agents since years, but the technology of the agents remains rather painful to implement. Safety remains the most difficult problem to solve for the use of the mobile agents. It is also the reason that prevents their diffusion.

Like any distributed system, the mobile agents are prone to the threats of the network like: the corruption of data, the masquerading, the refusal of service, listening, repudiation... etc.

The solutions to these problems exist: coding, authorization, authentication, non-repudiation, etc. these solutions are only adapted to the traditional distributed systems (for example: customer-server).

With regard to the mobile agents, it is not only necessary to protect the visited hosts, which is legitimate, but also the agents against falsifications. In fact, the mobile agents involve new problems of confidentiality, integrity and availability.

The problems of safety relating to the mobile agents are divided into two categories: protection of the sites of reception against the malicious agents and the protection of the agents against the malicious sites.

Current research does not offer general method to protect the agents. The developers of applications to mobile agents must design their own mechanisms of protection according to their needs.

In this work, we will propose a model of security that will answer the concerns that we quoted. Above To illustrate this model, we will implement a standard application e-commerce that uses this model.

Key words: mobile agents, distributed applications, security, encoding, mobility, e-commerce.

Résumé

On parle d'agents mobiles depuis des années, mais la technologie des agents reste plutôt pénible à mettre en œuvre. La sécurité reste le problème le plus difficile à résoudre pour l'utilisation des agents mobiles. C'est aussi la raison qui empêche leur diffusion.

Comme tout système distribué, les agents mobiles sont sujets aux menaces du réseau comme : la corruption de données, le masquerading, le déni de service, les écoutes, la répudiation... etc.

Les solutions à ces problèmes existent : chiffrement, autorisation, authentification, non-répudiation, etc. Mais ces solutions sont adaptées aux systèmes répartis classiques (par exemple : client-serveur).

En ce qui concerne les agents mobiles, il faut non seulement sécuriser les hôtes visités, ce qui est légitime, mais aussi les agents contre les falsifications. En fait, les agents mobiles posent de nouveaux problèmes de confidentialité, d'intégrité et de disponibilité.

Les problèmes de sécurité relatifs aux agents mobiles se scindent en deux catégories : protection des sites d'accueil contre les agents malveillants et la protection des agents contre les sites malveillants.

Les recherches actuelles n'offrent pas de méthode générale pour protéger les agents. Les développeurs d'applications à agents mobiles doivent concevoir leurs propres mécanismes de protection selon leurs besoins.

Dans ce travail, nous allons proposer un modèle de sécurité qui répondra aux préoccupations que nous avons citées. Pour illustrer ce modèle, nous implémenterons une application type e-commerce qui utilise ce modèle.

Mots clés : agents mobiles, applications réparties, sécurité, cryptage, mobilité, e-commerce.

TABLE DES MATIÈRES :

INTRODUCTION GÉNÉRALE	1
CHAPITRE I : LA TECHNOLOGIE DES AGENTS.....	4
INTRODUCTION	4
I. LES AGENTS AUJOURD'HUI, POURQUOI ? POURQUOI FAIRE ?.....	4
II. DÉFINITION D'UN AGENT.....	5
II.1. Définition générale	5
II.2. Quelle définition donnent les experts ?.....	5
▪ AIMA.....	5
▪ Maes	6
▪ KIDSIM.....	6
▪ Hayes-Roth	6
▪ IBM	6
▪ Wooldrige-Jenning	6
▪ Franklin/Graesser	6
III. AGENT ET ENVIRONNEMENT	6
III.1. Caractéristiques des environnements	7
IV. CARACTÉRISTIQUES DES AGENTS.....	7
V. CLASSIFICATION DES AGENTS.....	9
a) Les agents réactifs.....	9
b) Les agents cognitifs.....	9
- Les agents intelligents.....	9
- Les agents collaborants.....	9
- Les agents d'interface(Agents assistants).....	9
- Les agents d'information.....	10
V.1. Agents cognitifs versus agents réactifs.....	11
V.2. Architecture d'un agent	11
VI. LES SYSTÈMES MULTI-AGENTS.....	12
VI.1. Caractéristiques des systèmes multi-agents.....	12
VI.2. Les inconvénients des systèmes multi-agents.....	12
VI.3. Interactions.....	12
VI.4. Communications.....	13
VI.5. Langages de communication inter-agents.....	13
VI.5.1. KIF.....	14
VI.5.2. KQML.....	14
VI.5.3. La FIPA et son ACL.....	15
VII. LES AGENTS INTELLIGENTS.....	15
VII.1. Définition	15
VII.2. Principe de fonctionnement des agents intelligents	16
VII.3. Typologie des agents intelligents intervenant sur Internet	16
VII.3.1. Agents de recherche d'information «OFF LINE »	17
VII.3.2. Consultation « OFF LINE »	17
VII.3.3. Agents de recherche avancés	18
VII.3.4. Agents pour le commerce électronique	18
VII.4. Principaux enjeux des agents intervenant sur Internet.....	18
VIII. CONCLUSION	19

Table des matières

CHAPITRE II : LES AGENTS MOBILES	20
INTRODUCTION	20
I. CODE MOBILE	20
• Les applets Web	20
• Les e-mails dynamiques	20
• Mécanismes anti-piratages	21
I.1. Les catégories de code mobile	21
I.1.1. Évaluation distante	21
I.1.2. Code à la demande	21
I.1.3. Les agents mobiles	22
II. LES AGENTS MOBILES	22
II.1. Définition d'un agent mobile	22
II.2. Les parties d'un agent mobile	22
II.3. Le cycle de vie d'un agent mobile	22
II.4. Particularités des agents mobiles	23
II.4.1. La création des agents mobiles	23
II.4.2. La migration	24
II.4.2.1. La migration forte	25
II.4.2.2. La migration faible	25
II.4.3. La communication entre agents.....	26
II.5. Principe de fonctionnement des agents mobiles.....	26
II.6. Les types d'agents mobiles	27
II.6.1. Les agents de notifications	27
II.6.2. Les agents itinérants	27
II.6.3. Les agents d'adaptation	27
III. AGENT MOBILE VERSUS MIGRATION DE PROCESSUS	27
III.1. Processus	27
III.2. Migration de processus	27
IV. LES SYSTÈMES D'AGENTS MOBILES EXISTANTS	28
• Odyssey	28
• Le système des Aglets	28
• Concordia	28
• Voyager	29
• Agent Tcl	29
• Telescript	29
• Mole	29
IV.1. Comparaison de quelques systèmes étudiés	29
V. JAVA: LANGAGE DE PROGRAMMATION POUR AGENTS MOBILES	30
VI. AVANTAGES ET INCONVÉNIENTS DES AGENTS MOBILES	30
VI.1. Avantages des agents mobiles	30
VI.1.1. Exemple sur la réduction de la charge réseau	30
VI.1.2. Exemple sur l'extensibilité des fonctions d'un serveur	31
VI.2. Les inconvénients des agents mobiles	32
VII. CONCLUSION	32

Table des matières

CHAPITRE III : APPLICATIONS RÉPARTIES ET COMMERCE ÉLECTRONIQUE	33
INTRODUCTION	33
I. APPLICATION RÉPARTIE	33
I.1. Définition d'une application répartie	33
I.2. Caractéristiques des systèmes répartis	34
I.3. Quelques exemples d'applications réparties	34
I.3.1. Applications à flot de données	34
I.3.2. Édition coopérative	34
I.3.3. Téléconférence	35
I.3.4. Télévision interactive	35
I.3.5. Commerce électronique	35
I.4. Formes d'interactions entre les entités d'une application répartie	35
I.4.1. Client –serveur	35
I.4.2. Serveurs coopérants	36
I.4.3. Objets partagés (<i>actifs ou passifs</i>)	36
I.4.4. Flots de communication	36
I.4.5. Code mobile	36
I.4.6. Agents	36
II. COMMERCE ÉLECTRONIQUE	36
II.1. Définition du commerce électronique	37
II.2. Les origines du commerce électronique	37
II.3. Typologie du commerce électronique	37
1. Le B to B (Business To Business, parfois noté B2B)	37
2. Le C to B (Consumers To Business, parfois noté C2B)	37
3. Le B to C (Business To Consumers, parfois noté B2C)	37
4. Le C to C (consumers To Consumers, parfois noté C2C)	38
5. Le B To E (Business To Employees, parfois noté B2E)	38
6. Le B To A (Business To Administration, parfois noté B2A)	38
II.4. L'achat en ligne	38
II.5. Modes de paiement	38
II.6. Le problème du paiement sur internet	39
II.7. La sécurité du paiement	39
II.8. Avantages du commerce électronique	40
II.9. L'avenir du commerce électronique	41
II.10. Utilisation des agents dans le commerce électronique	41
II.10.1. Catégories d'agents pour le commerce électronique	41
II.10.1.1. Les agents acheteurs	42
II.10.1.2. Les agents vendeurs	42
II.11. Négociation en commerce électronique	43
III. PROBLÉMATIQUE	44
IV. CONCLUSION	45
CHAPITRE IV : PROPOSITION D'UNE SOLUTION DE SÉCURITÉ	46
INTRODUCTION	46
I. DÉFINITION DE LA SÉCURITÉ	46
I.1. Protection d'un site hôte envers un agent mobile	47

Table des matières

I.1.1. Quelques solutions	47
1. Bac à Sable	47
2. Signature de code	47
3. Contrôle d'accès	47
4. Vérification du code	48
I.2. Protection d'un agent envers un site hôte	48
I.2.1. Quelques solutions	49
II. UN MODÈLE D'INTERACTION EFFICACE ENTRE LES AGENTS MOBILES	49
II.1. Présentation du modèle Acheteur / Vendeur	49
II.1.1. Principe de fonctionnement du modèle Acheteur / Vendeur	50
II.1.2. Application du modèle Acheteur / Vendeur	51
II.1.2.1. Schéma « SB »	51
III. ARCHITECTURE PROPOSÉE	52
III.1. Vue générale	52
III.2. Description de l'architecture proposée	52
III.3. Scénario de fonctionnement de l'architecture proposée	54
a) Sur le site Acheteur	54
b) Sur le site vendeur	54
IV. LA SÉCURITÉ DU SYSTÈME	55
IV.1. Description détaillée de l'architecture proposée	56
IV.1.1. Architecture d'une place de marché	56
IV.1.2. Bases de connaissances	56
IV.1.3. L'autorité bancaire virtuelle (VBA)	57
V. CONCLUSION	57
 CHAPITRE V : CONCEPTION DU SYSTÈME CESAM	 58
INTRODUCTION	58
I. DESCRIPTION D'UN MODÈLE D'APPLICATION DE E-COMMERCE	58
I.1. Les cas d'utilisation.....	59
1. Les acteurs.....	59
2. Détermination des cas d'utilisation.....	59
2.1. Achat d'un produit.....	60
2.1.1. Recherche d'un produit.....	60
2.1.2. Validation d'achat.....	61
2.2. Vente d'un produit.....	63
2.2.1. Proposition d'un service.....	63
2.3. Diagramme de cas d'utilisation de l'application.....	64
II. DESCRIPTION DU SYSTÈME CESAM À TRAVERS LE MODÈLE D'APPLICATION PROPOSÉ	66
II.1. Achat d'un produit.....	66
II.1.1. Diagramme de séquence	67
Scénario 1 : Traitement de la requête client jusqu'à l'affectation d'itinéraire...	67
Scénario 2 : Migration d'un agent acheteur vers un site MP.....	68
Scénario 3 : Migration d'un agent acheteur vers un site e-shop.....	70
Scénario 4 : Retour de l'agent acheteur	72
Scénario 5 : Validation d'achat.....	73
Scénario 6 : Sécurité d'achat.....	74
Scénario 7 : Procédure du paiement dans l'achat.....	76

Table des matières

Scénario 8 : Modification de catalogue local.....	77
II.2. Vente d'un produit.....	79
II.2.1. Diagramme de séquence.....	80
Scénario 1 : Traitement de la requête vendeur jusqu'à l'affectation d'itinéraire..	80
Scénario 2 : Migration d'un agent vendeur vers un site MP.....	81
Scénario 3 : Migration d'un agent vendeur vers un site e-shop.....	83
Scénario 4 : Sécurité de vente.....	85
II.3. Diagramme de classe général	85
III. CONCLUSION	87
CHAPITRE VI : IMPLÉMENTATION ET TEST	88
INTRODUCTION.....	88
I. CHOIX DU LANGAGE DE PROGRAMMATION.....	88
I.1. Mobilité et portabilité du code	88
I.2. Liaison dynamique.....	88
I.3. Sérialisation.....	89
II. RÉALISATION DU SYSTÈME CESAM.....	89
II.1. Réalisation d'un environnement d'exécution (ASP)	89
II.1.1. Implémentation de l'agent mobile acheteur.....	89
II.1.2. Implémentation de l'agent mobile vendeur.....	90
II.2. Simulation d'une autorité de sécurité (TSA)	91
II.3. Réalisation d'une place de marché.....	91
II.3.1. Implémentation de l'agent statique facilitateur.....	91
II.3.1.1. L'agent facilitateur de l'ASP.....	91
II.3.1.2. L'agent facilitateur de MP.....	92
III. RÉALISATION DE L'APPLICATION.....	93
III.1. La spécification des besoins.....	93
III.2. L'affectation des certificats.....	94
III.3. La place de marché.....	95
III.4. L'affichage des résultats.....	95
III.5. Le paiement.....	95
IV. TEST.....	96
CONCLUSION GÉNÉRALE	102
BIBLIOGRAPHIE	104
ANNEXE	111

Table des matières

LISTE DES FIGURES

Figure I.1 : Interaction entre un agent et son environnement	7
Figure I.2 : Classification des principaux agents cognitifs	10
Figure I.3 : Architecture interne d'un agent cognitif	11
Figure I.4 : Les caractéristiques d'un agent intelligent	16
Figure I.5 : Les composants d'un agent intelligent	16
Figure I.6 : Fonctionnalités des agents de recherche d'information	17
Figure II.1 : Schéma évaluation distante	21
Figure II.2 : Schéma code à la demande	21
Figure II.3 : Les parties d'un agent mobile	22
Figure II.4 : Cycle de vie d'un agent Mobile	23
Figure II.5 : Phase1	24
Figure II.6 : Phase2	24
Figure II.7 : Phase3	24
Figure II.8 : Phase4	24
Figure II.9 : Phase5	25
Figure II.10 : Fonctionnement d'un agent mobile	26
Figure II.11 : Réduction de la charge réseau	31
Figure II.12 : Extensibilité des fonctions d'un serveur	32
Figure III.1 : Zone de négociation entre un acheteur et un vendeur	44
Figure IV.1 : Interaction des agents mobiles à travers une place de Marché	50
Figure IV.2 : Différentes implémentations du schéma SB	51
Figure IV.3 : Schéma général de l'architecture proposée	53
Figure IV.4 : Conception coté client	54
Figure IV.5 : Schéma de la sécurité	55
Figure IV.6 : Création d'un compte bancaire virtuel	57
Figure V.1 : Diagramme de cas d'utilisation général de l'application	59
Figure V.2 : Diagramme de cas d'utilisation général de l'achat	60
Figure V.3 : Diagramme de cas d'utilisation de la recherche d'un produit	61
Figure V.4 : Diagramme de cas d'utilisation de la validation d'achat	61
Figure V.5 : Diagramme de cas d'utilisation de l'achat d'un produit	62
Figure V.6 : Diagramme de cas d'utilisation général de la vente	63
Figure V.7 : Diagramme de cas d'utilisation de la vente d'un produit.	64
Figure V.8 : Diagramme de cas d'utilisation de l'application	65
Figure V.9 : Décomposition de la fonction d'achat	66
Figure V.10 : Diagramme de séquence du traitement de la requête client.	67
Figure V.11 : Diagramme de séquence de la migration d'un agent acheteur vers un site MP	69
Figure V.12 : Diagramme de séquence de la migration d'un agent acheteur vers un site e-shop	71
Figure V.13 : Diagramme de séquence de retour de l'agent acheteur	72
Figure V.14 : Diagramme de séquence de la validation d'achat	73

Table des matières

Figure V.15 : Diagramme de séquence de la sécurité d'achat	75
Figure V.16 : Diagramme de séquence du paiement	76
Figure V.17 : Diagramme de séquence de la modification de catalogue local	78
Figure V.18 : Décomposition de la fonction de vente	79
Figure V.19 : Diagramme de séquence du traitement de la requête vendeur	80
Figure V.20 : Diagramme de séquence de la migration d'un agent vendeur vers un site MP	82
Figure V.21 : Diagramme de séquence de la migration d'un agent vendeur vers un site e-shop	84
Figure V.22 : Diagramme de séquence de la sécurité de vente	85
Figure V.23 : Diagramme de classe détaillé de système	86
Figure VI.1 : La forme de la classe SaBoAgent	90
Figure VI.2 : La forme de la classe agent acheteur	90
Figure VI.3 : La forme de la classe agent vendeur	91
Figure VI.4 : La forme de la classe agent facilitateur de MP	92
Figure VI.5 : La forme de la classe SaBoAgentMessage	92
Figure VI.6 : Interface de l'inscription	93
Figure VI.7 : Interface de spécification des besoins pour les acheteurs	94
Figure VI.8 : Interface de spécification des besoins pour les vendeurs	94
Figure VI.9 : Le format d'un certificat	95
Figure VI.10 : L'interface du paiement	96
Figure VI.11 : L'agent vendeur dans l'ASP	97
Figure VI.12 : L'agent vendeur dans la place de marché	97
Figure VI.13 : La négociation dans la place de marché	98
Figure VI.14 : Retour de l'agent acheteur à l'ASP	98
Figure VI.15 : L'interface d'affichage des résultats d'achat	98
Figure VI.16 : La détermination de nombre d'unité	99
Figure VI.17 : La validation d'achat	99
Figure VI.18 : Information bancaire	99
Figure VI.19 : La fenêtre de crédit du compte acheteur	100
Figure VI.20 : La fenêtre de débit du compte acheteur	100
Figure VI.21 : Modification de catalogue local	101
LISTE DES TABLEAUX :	
Tableau I.1 : Comparaison entre agents cognitifs et agents réactifs	11
Tableaux II.1 : Les agents mobiles versus la migration de processus	28
Tableaux II.2 : Comparaison de quelques systèmes étudiés	29
Tableau VI.1 : Exemple	96

Introduction générale

L'informatique répartie actuelle est principalement fondée sur le modèle client-serveur. Les exemples les plus connus sont probablement le World Wide Web et les systèmes de type Corba [59]. Cependant, deux problèmes importants restent difficiles à traiter :

Efficacité : Le coût des communications entre les clients et les serveurs reste élevé, ce qui explique les efforts des chercheurs pour optimiser les services de communication et pour développer des techniques de cache visant à masquer les lenteurs des réseaux.

Généricité : Il est difficile pour un serveur de fournir un service satisfaisant tous ses clients potentiels. De plus, si un serveur vise à fournir un service plus générique, cela se traduit souvent par une interface plus complexe impliquant des interactions plus fréquentes entre les clients et le serveur, d'où une perte en terme d'efficacité pour les clients.

Récemment, la recherche en systèmes répartis a vu l'émergence d'un nouveau modèle pour la structuration d'applications réparties sur Internet : la programmation par agents mobiles[79].

Les agents mobiles permettent de répondre aux deux problèmes mentionnés ci-dessus [80]. Un serveur peut fournir un service et une interface très simples et génériques à ses clients. Cette généricité permet aux clients ayant des besoins spécifiques d'adapter le service. Afin de limiter le coût induit par les nombreuses interactions entre le client et le serveur, le client peut programmer un agent mobile qui est envoyé et exécuté sur la machine du serveur. Cet agent mobile implante une extension du serveur qui réalise exactement le service désiré par le client.

Le concept d'agent mobile est en train d'attirer l'attention, ce qui signifie son utilisation dans diverses applications réparties comme le commerce électronique.

Dans le commerce électronique, les agents mobiles commencent à prendre une place importante en vue d'automatiser un certain nombre de tâches. Il s'agit de rendre la comparaison des prix, la fonction d'achat, la fonction de vente, la négociation de plus en plus automatisée. Cette tendance permettra à l'entreprise d'améliorer la connaissance qu'elle a de ses clients pour mieux adapter son offre aux besoins de marchés. Quant au client cela lui permettra de tirer avantage des comparaisons des prix et des produits.

Mais la technologie des agents mobiles est limitée par le problème de sécurité qui est l'un des problèmes les plus délicats.

Les recherches actuelles n'offrent pas de méthode pour protéger les agents. Les développeurs d'applications à agents mobiles doivent concevoir leurs propres mécanismes de protection selon leurs besoins.

Dans ce mémoire. Nous allons proposer un modèle de sécurité qui résoudra le problème de sécurité posé par les agents mobiles.

Motivations et objectifs

La propriété de sécurité et de sûreté de fonctionnement sont essentielles et doivent être prises en considération au plus tôt dans les applications de commerce électronique.

Notre travail a pour but d'assurer la sécurité des agents mobiles notamment dans le domaine de commerce électronique.

Les principaux objectifs que nous nous sommes fixés sont :

1. Etudier la technologie des agents, plus particulièrement les agents mobiles.
2. Présenter le domaine de commerce électronique.
3. Présenter le modèle Acheteur / vendeur.
4. Proposer une architecture qui assure la sécurité des agents mobiles dans le domaine de commerce électronique.
5. Décrire en détail l'architecture proposée à l'aide du langage UML.
6. Mettre en œuvre l'architecture proposée.

Méthodologie suivie :

Pour atteindre nos objectifs, nous commençons par une présentation de l'état de l'art des agents, des agents mobiles et des applications réparties.

Ensuite nous abordons le problème de sécurité posé par l'utilisation des agents mobiles notamment dans le domaine de commerce électronique et nous proposons une architecture capable de résoudre ce problème.

Finalement nous présentons la description détaillée de l'architecture proposée à l'aide du langage UML et sa mise en œuvre en utilisant le langage Java.

Présentation de mémoire

Le présent mémoire se compose de 6 chapitres :

Chapitre 1 : Dans ce chapitre, nous présentons la technologie des agents en insistant sur la notion d'agent intelligent.

Introduction générale

- Chapitre 2 : Dans ce chapitre, nous présentons les agents mobiles comme un nouveau paradigme pour les applications réparties.
- Chapitre 3 : Dans ce chapitre, nous présentons les applications réparties et l'utilité de l'utilisation des agents mobiles dans le domaine de e-commerce. Finalement nous abordons le problème de sécurité posé par l'utilisation des agents mobiles
- Chapitre 4 : Dans ce chapitre, nous présentons les diverses solutions qui existent, et nous proposons une solution basée sur le modèle acheteur / vendeur qui permet de limiter le problème de sécurité posé par les d'agents mobiles dans une application de commerce électronique.
- Chapitre 5 : Dans ce chapitre, nous décrivons l'architecture proposée en détail à l'aide du langage UML.
- Chapitre 6 : Dans ce chapitre, nous proposons une implémentation de l'architecture proposée en utilisant le langage de programmation Java.

Chapitre I

La technologie des agents

INTRODUCTION

L'informatique est en train de changer de manière assez profonde. Tout d'abord, elle devient ubiquitaire. Au départ confinée aux ordinateurs, elle est en train d'investir les objets de la vie courante : téléphones portables, assistants personnels, etc. Elle devient ainsi de plus en plus diffuse et distribuée dans de multiples objets et fonctionnalités qui sont amenés à coopérer.

La décentralisation est donc la règle et une organisation coopérative entre module logiciels est un besoin. De plus la taille, la complexité et l'évolutivité croissantes de ces nouvelles applications informatiques font qu'une vision centralisée, rigide et passive atteint ses limites. On est ainsi naturellement conduit à chercher à donner plus d'autonomie et d'initiative aux différents modules logiciels. Le concept d'agent propose un cadre de réponse à ces deux enjeux complémentaire (et à première vue contradictoire) : l'autonomie et l'organisation [1].

I. LES AGENTS AUJOURD'HUI, POURQUOI ? POURQUOI FAIRE ?

- Les applications sont si riches que les utilisateurs ne les maîtrisent plus :
Les agents, qui vont connaître les intérêts de leur utilisateur vont travailler pour eux.
- Les sources d'information augmentent de façon exponentielle :
Les agents, par des techniques de type " data mining " vont rechercher l'information significative.
- Les bandes passantes des réseaux augmentent, mais on a un temps limité pour les utiliser :
Les agents vont réguler le flot d'information.
- Les puissances machines actuelles permettent de mettre plus d'intelligence locale au service de l'utilisateur.
- Les universités et les centres de recherches sont au seuil de pouvoir mettre sur le marché le fruit de leurs avancées dans des domaines afférents (Intelligence artificielle,...) et beaucoup de ces produits peuvent déjà être obtenus gratuitement sur le réseau. Plusieurs sociétés fournissent des logiciels et des services utilisant des agents.

Si on regarde à titre d'exemple ce qu'une société comme IBM commercialise aujourd'hui on trouve :

- **World Avenue :**
Système d'achat électronique sous Internet qui utilise les agents pour repérer les habitudes d'achat. Ce système utilise un agent, " Intelligent Miner " qui utilise des techniques avancées pour analyser les informations.
- **Lotus Notes :**
Possède des agents permettant l'automatisation de nombreuses tâches (messages automatiques, classements, priorités, recherche automatique périodique sur des mots-clés, ...).
- **WBI (Web Browser Intelligence) :**
Permet à l'utilisateur Web de se rappeler ce qu'il a déjà trouvé, le temps de réponse des sites visités, grâce à des agents intégrés.
- **Aglets :**
Ce mot, inventé suite au mot applet (petites applications écrites en JAVA), référence des agents écrits en JAVA (Agent Applets). De nombreux agents de ce type sont déjà téléchargeables.

II. DÉFINITION D'UN AGENT

D'un point de vue informatique, il n'existe pas encore un consensus sur la définition d'un agent. En plus de la relative jeunesse du domaine, cela s'explique par le fait que diverses communautés revendiquent ce terme avec des problématiques assez différentes même si ces celles-ci sont complémentaires et conduites à se rencontrer à terme [1].

Dans la littérature spécialisée, on trouve une multitude de définition des agents, elles se ressemblent toutes, mais diffèrent selon le type d'application pour lequel est conçu l'agent.

II.1. Définition générale

Un agent est une entité logicielle ou physique à laquelle est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents.

II.2. Quelle définition donnent les experts ?

- **AIMA [26, page 33]**
Tout objet percevant son environnement à travers des détecteurs et agissant sur cet environnement grâce à des actionneurs (une définition proche du robot, ou du programme.....).

- **Maes [27, page 108]**
Systèmes informatiques qui existent dans un environnement complexe et dynamique, perçoivent et agissent de façon autonome dans cet environnement, et de ce fait réalisent un jeu d'objectifs et de tâches pour lesquels ils sont conçus.
- **KIDSIM [28]**
Une entité logicielle persistante, dédiée à des buts spécifiques.
- **Hayes-Roth [29]**
3 fonctions : perception des changements dynamiques de l'environnement, actions sur cet environnement, modèles de raisonnement pour interpréter les perceptions, déduire, définir les actions.
- **IBM [31]**
Entités logicielles qui transportent un jeu d'opérations pour le compte d'un utilisateur ou d'un programme, avec un certain degré d'indépendance et d'autonomie, et, de ce fait, utilisent quelques connaissances ou représentations des objectifs ou désirs de leur utilisateur.
- **Wooldrige-Jenning [32, page 2]**
un système informatique matériel ou logiciel autonome, social, réactif à l'environnement, pro-actifs.
- **Franklin/Graesser :**
Un agent est un système, situé dans un environnement et appartenant à cet environnement, qui perçoit cet environnement, agit sur lui, en poursuivant son propre objectif.

III. AGENT ET ENVIRONNEMENT

Du point de vue d'un agent, l'environnement correspond à tout ce qui lui est extérieur : les autres agents font aussi a priori partie de l'environnement. Cependant, l'environnement est souvent considéré comme une structure centralisée dynamique mais non autonome, qui contient l'ensemble des entités avec lesquelles interagissent les agents. (Voir **Figure I.1**).

En outre, l'agent doit être en mesure de recevoir des informations de son environnement (par des récepteurs) et d'agir sur ce même environnement (par des effecteurs) suivant un comportement établi à partir des observations et du raisonnement de l'agent.

Les modules de communication sont primordiaux, d'autant plus que l'environnement est constitué d'autres agents avec lesquels l'agent peut coopérer pour atteindre son objectif [18].

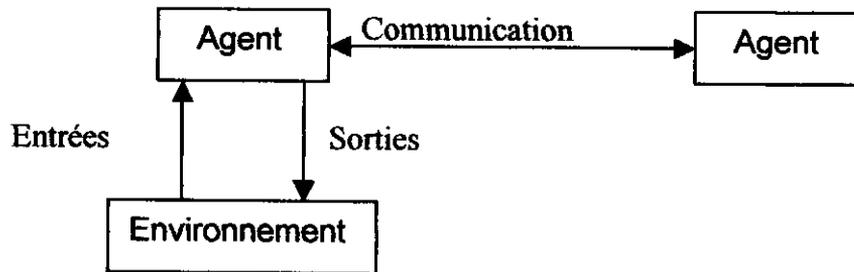


Figure I.1 : Interaction entre un agent et son environnement.

III.1. Caractéristiques des environnements

- **Observable (ou non)**
Dans un environnement observable un agent peut obtenir de l'information complète, précise et à jour sur l'état de l'environnement.
- **Déterministe (ou non)**
Dans un environnement déterministe une action a un effet unique garanti.
- **Statique/Dynamique**
Un environnement statique ne change que sous l'effet des actions d'un agent. Par contre dans un environnement dynamique, les agents ne contrôlent pas entièrement la manière dont l'environnement change.
- **Discret/Continu**
Un environnement est discret s'il y a un nombre fini d'actions bien définies, sinon on dira que l'environnement est continu.

IV. CARACTÉRISTIQUES DES AGENTS

Comme cité précédemment, il n'y a pas de définition standard des agents, mais en partant de l'ouvrage de Wooldrige et Jennings [32], on peut identifier les caractéristiques suivantes pour la notion d'agent :

- **Situé :**
L'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement.
- **Autonome :**
L'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.
- **Proactif :**
Ceci caractérise la capacité d'un agent à anticiper des changements plutôt qu'à réagir au changement.

- **Réactivité :**
Ceci caractérise la capacité d'un agent à effectuer une nouvelle tâche lorsqu'un événement survient.
- **Social :**
L'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.

Bien que presque toutes les caractéristiques ci-dessus des agents puissent être considérées en tant que partage de quelque chose avec le comportement intelligent, les chercheurs ont essayé de donner une différence nette entre les agents logiciels (software) et les agents logiciels intelligents, glissant dans le monde des agents la différence si recherchée entre les programmes et les programmes intelligents.

Donc les propriétés citées précédemment soient communes à toute la communauté de chercheurs en intelligence artificielle [2].

Mais il y a d'autres propriétés qui peuvent être rajoutées. Comme :

- **Négociation et coopération :**
La capacité qu'ont les agents d'effectuer un travail collaboratif pour accomplir une tâche donnée.
- **Adaptatif :**
Un agent possède cette propriété s'il est capable d'utiliser ses connaissances afin de modifier son comportement (augmentation de l'espace disque, détection d'un seuil, etc.).
- **Mobilité:**
La capacité d'un agent de se déplacer d'un ordinateur à un autre pour effectuer des traitements.
- **Communication :**
La possibilité d'échanges de messages entre agents.
- **Délibération :**
L'agent doit être capable de comprendre son environnement et de prendre une seule décision.
- **Perception :**
Un agent possède la capacité de perception qui permet d'acquérir des informations sur son environnement et sur lui-même.
- **Intentionnalité :**
Une intention est la déclaration explicite des buts et des moyens d'y parvenir pour un agent, elle exprime la volonté d'atteindre ses buts ou d'effectuer des actions. Un agent intentionnel est un agent guidé par ces buts [3].

V. CLASSIFICATION DES AGENTS [5] :

Les agents peuvent être classés selon leurs degrés d'autonomie, de coopération et d'adaptation, caractéristiques généralement considérées comme principales en IAD [4] :

1. Le plus haut niveau d'autonomie permet à l'agent de planifier ses actions.
2. Le plus haut niveau de coopération accorde à l'agent des capacités de Négociation.
3. Le plus haut degré d'adaptation permet à l'agent d'adapter, d'acquérir des connaissances.

Un agent possédant ces trois caractéristiques à leur plus haut niveau est appelé Agent intelligent.

Deux types d'agents ont longtemps été distingués dans les SMA [6] :

a) Les agents réactifs

Ces agents ne sont pas doués d'intentions explicites, mais ont des comportements collectifs et leur fonctionnement repose sur la loi stimulus-action.

b) Les agents cognitifs

Ces agents ont une représentation explicite de leur environnement et des autres agents, ils possèdent l'intention de coopérer, et ils ont des comportements plus réfléchis.

D'autres types d'agents, qualifiés d'hybrides, combinant le fonctionnement de ces deux types d'agent, sont ensuite apparus.

Plusieurs types d'agents cognitifs peuvent être distingués, en particulier :

- Les agents intelligents

Combinent les trois caractéristiques (autonomie, coopération, adaptation) à leur plus Haut niveau. Ils sont en général dotés de la capacité d'apprentissage [7].

- Les agents collaborateurs [8] :

Ce sont des agents cognitifs non apprenants, on considère qu'ils sont à la fois fortement autonomes et coopérants, mais peu adaptatifs.

Ils sont surtout utilisés dans les domaines qui nécessitent une décentralisation comme par exemple la maintenance de réseau, ou encore pour simuler le comportement d'organisations humaines ou animales [9].

- Les agents d'interface [10, 11] (Agents assistants [12]) :

Ils ont pour mission de capturer les actions de l'utilisateur, ceci afin de le conseiller vis-à-vis de son activité, ou de modifier l'aspect de l'interface selon le plus souvent un schéma de règles préétablies [13].

Dans l'état actuel des recherches, ces agents possèdent en général une capacité de coopération limitée, ils sont principalement utilisés pour l'assistance à l'utilisateur dans le cas d'interfaces aux fonctionnalités nombreuses et complexes (par exemple, dans certaines suites bureautiques), mais également dans le domaine des systèmes tuteurs intelligents afin de faciliter l'apprentissage humain.

– **Les agents d'information [14, 15] :**

Ces agents qui apparaissent de plus en plus sur le marché sont dédiés à la recherche d'information, principalement sur l'internet. Il faut cependant noter que peu de logiciels de ce type méritent réellement le qualificatif d'agent (au sens de l'Intelligence Artificielle Distribuée), beaucoup ne sont que des programmes se déclenchant à date fixe, sans capacité d'adaptation, ni de coopération (par exemple, les nombreux logiciels de veille qui se déclenchent à heure fixe pour rechercher des informations sur un produit ou un concurrent).

Par contre, un agent d'information (au sens pur du terme) devrait posséder une grande autonomie, en étant capable d'agir seul, soit en fonction d'un calendrier, soit en fonction d'un manque d'information, soit en fonction d'une nouvelle disponibilité d'information, il doit être capable d'adapter son comportement selon les besoins de l'utilisateur ou de la quantité ou pertinence de l'information.

Notons que dans le cas d'une recherche d'information par groupe d'agents, s'il n'y a pas de coordination au sein du groupe, il y a un risque de redondance dans certaines informations rapatriées, source de problème ergonomique.

La figure I.2 positionne globalement ces différents types d'agents cognitifs selon leurs degrés d'autonomie, d'adaptation et de coopération.

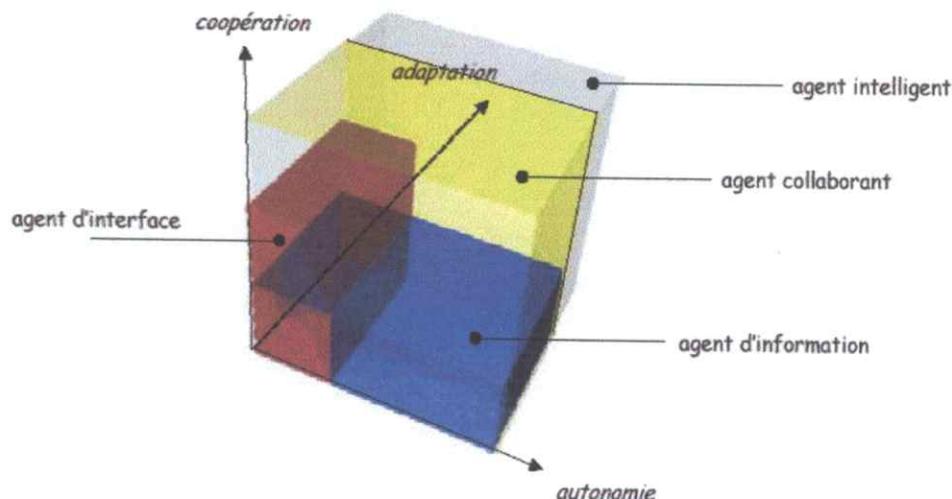


Figure I.2 : Classification des principaux agents cognitifs. [5]

V.1. Agents cognitifs versus agents réactifs

Tableau I.1: Comparaison entre agents cognitifs et agents réactifs.

Agents cognitifs	Agents réactifs
Représentation explicite	Pas de représentation explicite
Architectures complexes	Architectures simples
Organisation explicite	Organisation implicite/induite
Communication explicite	Communication via l'environnement.
Petit/moyen nombre d'agents	Grand ou très grand nombre d'agents
Peut tenir compte de son passé	Pas de mémoire de son historique

V.2. Architecture d'un agent

On peut résumer les différents types d'architecture d'agent cognitif décrits dans la littérature avec le modèle d'architecture d'agent suivant :

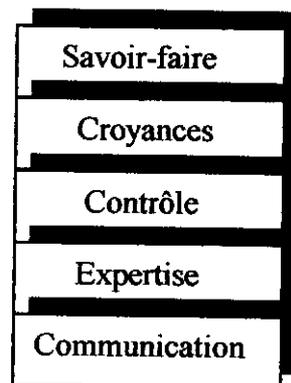


Figure I.3 : Architecture interne d'un agent cognitif. [3]

Savoir-faire :

C'est une Interface de déclaration des compétences et des connaissances d'agent.

Croyances :

Ce sont des connaissances non objectives sur l'environnement et sur les autres agents dans un système multi-agents.

Contrôle :

C'est la connaissance des buts, des intentions, des plans et des tâches de l'agent.

Expertise :

C'est une connaissance sur la résolution d'un problème, sorte de système expert utilisant un formalisme des règles, qui constitue une base de règles.

Communication :

La communication entre agent n'est possible qu'avec la présence d'un protocole de communication qui permet d'interagir avec les autres agents.

VI. LES SYSTÈMES MULTI-AGENTS

Un système multi-agents (SMA) est constitué d'un ensemble d'agents vivants au même moment, partageants des ressources communes et communicants entre eux. Les agents interagissent selon des modes de coopération, compétition, de coexistence. Les agents d'un SMA coopèrent dans l'exécution d'un but commun.

VI.1. Caractéristiques des systèmes multi-agents

Le système multi-agents possède en outre deux caractéristiques importantes : [16].

- **Ouvert ou fermé**
Tous les agents sont développés ou non par le même programmeur.
- **Homogène ou hétérogène :**
Les agents poursuivent tous ou non le même but.

VI.2. Les inconvénients des systèmes multi-agents

La principale difficulté induite par l'utilisation des systèmes multi-agents est qu'il est plus difficile de maîtriser le comportement et l'évolution du système global, constitué d'agents autonomes :

Pour un problème donné, il n'est pas assuré que l'ensemble des agents converge vers une solution commune et unique[17].

VI.3. Interactions

Une des principales propriétés de l'agent dans un SMA est celle d'interagir avec les autres agents. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent. Elles permettent aux agents de participer à la satisfaction d'un but global. Cette participation permet au système d'évoluer vers un de ses objectifs et d'avoir un comportement intelligent indépendamment du degré de complexité des agents qui le composent.

En général, les interactions sont mises en œuvre par un transfert d'informations entre agents ou entre l'environnement et les agents, soit par perception, soit par communication.

- 1) Par la perception, les agents ont connaissance d'un changement de comportement d'un tiers.
- 2) Par la communication, un agent fait un acte délibéré de transfert d'informations vers un ou plusieurs autres agents.

L'interaction peut être décomposée en trois phases non nécessairement séquentielles [6] :

- La réception d'informations ou la perception d'un changement,
- le raisonnement sur les autres agents à partir des informations acquises,
- Une émission de message(s) ou plusieurs actions (plan d'actions) modifiant l'environnement. Cette phase est le résultat d'un raisonnement de l'agent sur son propre savoir-faire et celui des autres agents.

VI.4. Communications

Les communications, dans les SMA comme chez les humains, sont à la base des interactions et de l'organisation. Une communication peut être définie comme une forme d'action locale d'un agent vers d'autres agents. Les questions abordées par un modèle de communication peuvent être résumées par l'interrogation suivante : qui communique quoi, à qui, quand, pourquoi, et comment ? .

VI.5. Langages de communication inter-agents

Depuis quelques années, des recherches ont permis aux programmes informatiques de s'échanger des informations et des connaissances. Les techniques suivantes ont déjà été utilisées, indépendamment des notions d'agent :

- **RPC** (Remote Procedure Call)
Il s'agit d'appeler des procédures à distance à travers un réseau. Ce protocole date de la grande époque d'UNIX, où il s'agissait de faire communiquer des programmes impératifs typiquement écrits en C.
- **RMI** (Remote Method Invocation)
C'est un mécanisme d'appel de méthode sur des objets Java, entre programme Java, qui fonctionne grâce au réseau. Bien qu'intéressant, ce mécanisme est cantonné au langage Java.
- **CORBA** (Common Object Resource Broker Architecture)
Proche par son principe de RMI, ce mécanisme est par contre disponible sur de nombreux langages de programmation. Il permet l'invocation de méthodes à distance et l'échange d'objets sur le réseau.

Cependant, ces techniques sont limitées car elles restent cantonnées au cadre de la programmation impérative ou procédurale : à chaque instant, le programme sait quelle procédure appeler. En ce sens, elles sont mal adaptées aux systèmes multi-agents.

C'est pourquoi, sont apparus les langages de communication entre agents (ACLs⁽¹⁾), qui vont plus loin :

1. Ils manipulent des propositions, des règles et des actions au lieu de simples objets sans sémantique associée.
2. Un message décrit un état dans un langage déclaratif, au lieu d'une simple invocation de méthode ou d'un simple appel de procédure.

Les types de messages des ACLs sont appelés des actes de langage.

Il existe plusieurs catégories de tels actes :

- Assertifs : Pour donner une information.
- Directifs : Pour donner des ordres.
- Promissifs : Pour s'engager sur une action à venir.
- Expressifs : Pour exprimer ses sentiments ou ses croyances.
- Déclaratifs : Pour énoncer un fait.
- Interrogatifs : Pour demander une information.

La compréhension entre agents passe par différents niveaux :

- 1) Tout d'abord, il faut que les agents parlent le même langage, ou bien qu'il existe des traducteurs bijectifs d'un langage dans un autre. On est ici au niveau syntaxique ;
- 2) Mais il faut également que les mêmes objets, les mêmes concepts aient la même signification pour tous les agents. Il est donc nécessaire que tous les agents partagent une ontologie commune ;
- 3) Enfin, les agents doivent être capables de dialoguer entre eux, pas au niveau du simple envoi de messages, mais à un niveau supérieur : ils doivent pouvoir s'informer, se poser des questions, rechercher d'autres agents, etc. Pour ce faire, les agents ne peuvent se contenter d'appeler une procédure ; ils doivent spécifier un état désiré dans un langage déclaratif : c'est ici qu'intervient l'ACL.

VL5.1. KIF⁽²⁾

Le niveau purement syntaxique (niveau 1) peut être pris en charge par KIF, dont la syntaxe est inspirée de Lisp.

VL5.2. KQML⁽³⁾

Le langage KQML se positionne au 3^{ème} niveau de notre classification. Il est complètement indépendant à la fois de la syntaxe et de l'ontologie sous-jacente.

⁽¹⁾ Agent Communication Languages.

⁽²⁾ Knowledge Interchange Format.

⁽³⁾ Knowledge Query and Manipulation Language.

Un message KQML se compose :

- D'un performatif, qui représente l'*en-tête* du message ;
- D'un certains nombres de couples attribut-valeur.

D'un point de vue plus théorique, on peut considérer qu'un message KQML se compose de trois couches :

- Couche de contenu : le contenu du message, c'est à dire la valeur de l'attribut «:content».
- Couche de communication : informations qui permettent l'acheminement du message, par exemple les valeurs des attributs «:sender» et «:receiver».
- Couche de message : composée du performatif et des valeurs des attributs qui décrivent le message, comme «:language» ou «:ontology».

VL5.3. La FIPA et son ACL

La FIPA ⁽⁴⁾ est une association à but non lucratif qui a pour ambition le développement des systèmes à agents.

En 1999, la FIPA a défini un ACL pour répondre principalement à une critique assez vigoureuse contre KQML.

L'ACL de la FIPA ressemble beaucoup à KQML, mais il ne comporte que 22 performatifs. De plus, il est défini de façon beaucoup moins ambiguë : La sémantique des performatifs est décrite dans un langage spécialisé, le langage SL pour Semantic Language (d'après les travaux de Sadek).

SL est donc utilisé pour décrire la sémantique formelle de l'ACL (avec une syntaxe mathématique). Mais il peut revêtir une syntaxe de type Lisp, de façon à être utilisé comme langage de contenu des messages eux-mêmes à la place de KIF.

VII. LES AGENTS INTELLIGENTS

VII.1. Définition

Ce sont des composants logiciels et / ou matériels capables à des degrés différents :

D'être autonome : Fonctionnement automatique.

De communiquer : Echanger des informations avec d'autres programmes ou des hommes.

D'apprendre : Capables de réagir avec un environnement, de s'adapter aux circonstances, de prendre une décision ou d'enrichir eux-mêmes leur propre comportement, sur la base d'observations qu'ils effectuent [20].

⁽⁴⁾ Foundation for Intelligent Physical Agents.

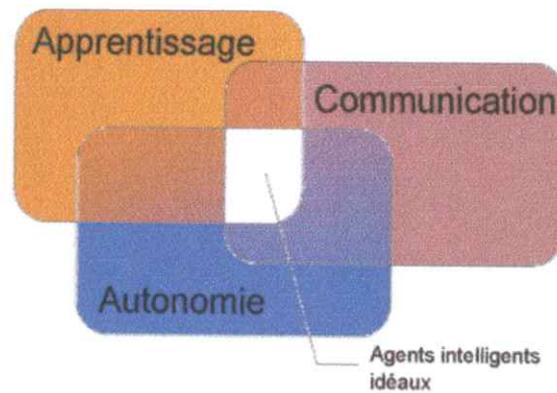


Figure I.4: Les caractéristiques d'un agent intelligent.

VII.2. Principe de fonctionnement des agents intelligents

Les principaux composants de l'agent intelligent sont les suivants:

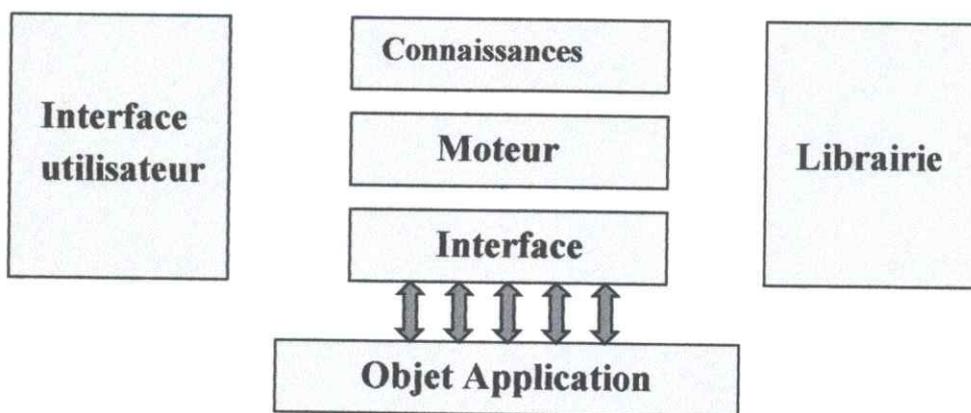


Figure I.5: Les composants d'un agent intelligent.

- Le moteur correspond au cerveau de l'agent.
- Les connaissances représentent ce que l'agent sait, croit et pense, elles sont stockées dans les librairies.
- L'interface utilisateur permet d'établir une communication entre l'agent et l'utilisateur.

VII.3. Typologie des agents intelligents intervenant sur Internet

Les agents intelligents fonctionnant sur Internet n'intègrent pas réellement de caractéristiques types. C'est pour cette raison qu'il paraît plus logique de les classer en fonction de leur mission. Ces agents peuvent se définir en fonction de leur capacité à exécuter des tâches spécifiques.

Par contre, dans le domaine de l'industrie, il est possible d'opérer des distinctions plus précises, notamment en fonction du degré de coopération des agents.

VII.3.1. Agents de recherche d'information «OFF LINE »

- **Méta-moteurs "Off line"**

Ils ont pour rôle de rechercher sur plusieurs moteurs les réponses à une requête, rapatrier les pages trouvées en local, puis classer et gérer les informations, ils passent ensuite à l'indexation des pages en local et l'élimination des doublons, ils peuvent aussi créer des résumés à la volée, organiser les résultats par thème tout en surveillant les modifications des sites avec une périodicité paramétrable afin de mettre en évidence des pages nouvelles par rapport à la dernière recherche.

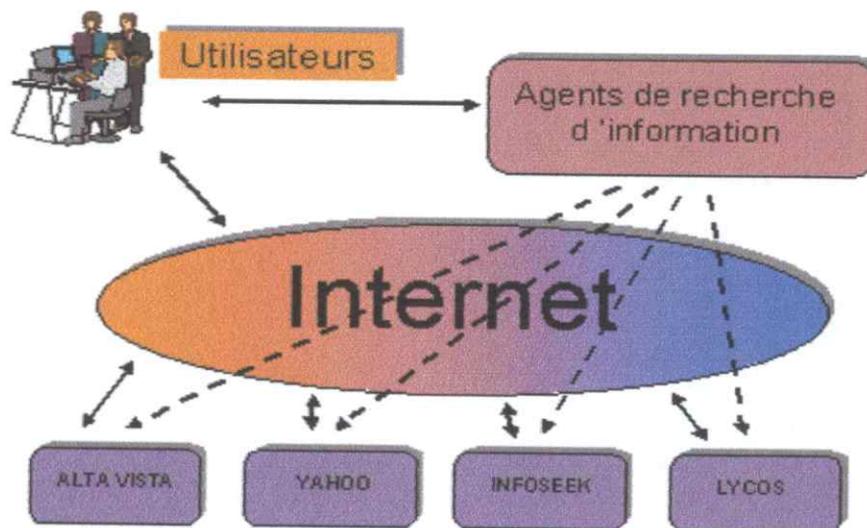


Figure I.6: Fonctionnalités des agents de recherche d'information.

- **Méta-moteurs sectoriels**

Les agents sectoriels sont des outils uniquement spécialisés dans un domaine précis comme, par exemple, la finance, les sciences et techniques, la littérature, etc.

Les agents sectoriels consultent des outils de recherche spécialisés dans des domaines précis.

VII.3.2. Consultation « OFF LINE »

Ces agents ont pour rôle de copier un site Web sur un poste local tout en respectant l'arborescence du site d'origine, en plus ils donnent la possibilité d'inclure ou d'exclure certains format et de spécifier dans l'arborescence le niveau des pages que l'on veut inclure.

VII.3.3. Agents de recherche avancés

Ils ont pour rôle de trouver toutes les pages contenant la requête, afin d'enrichir cette dernière, ils passent ensuite à l'analyse et au filtrage des pages trouvées en rapatriant uniquement les bonnes pages.

VII.3.4. Agents pour le commerce électronique

Ils ont deux missions :

- 1) Faciliter aux consommateurs la sélection de boutiques, de marques ou de produits en se basant sur leurs goûts.
- 2) Permettre aux vendeurs de mieux connaître la demande.

A. Les agents contrôlés par le consommateur, shopping agents ou shopbots :

Le but de ces agents est de rechercher un produit précis, comparer les prix ou les services, enfin acheter ou recommander les meilleurs produits avec les meilleurs prix.

Ce sont des programmes qui parcourent le web pour comparer les prix des produits ou services grand public.

Le premier agent de ce type a été créé en 1995. Bargain Finder, conçu par Andersen Consulting, permettait de comparer les prix des disques vendus sur Internet. Ce fut un échec, mais aujourd'hui avec l'essor du commerce électronique de nouvelles solutions font leur apparition.

B. Les agents d'analyse de la demande des consommateurs :

La fonction principale de ces agents est d'analyser les demandes pour mieux connaître les consommateurs, gérer des profils clients et personnaliser l'offre pour chaque client.

VII.4. Principaux enjeux des agents intervenant sur Internet

Les agents intelligents ne sont pour le moment pas vraiment intelligents. Cependant ils deviennent de plus en plus efficaces parce qu'ils savent prendre des initiatives pour pouvoir collaborer de manière précise avec l'utilisateur ou d'autres applications. Ils peuvent aussi être mobiles pour se déplacer sur tel ou tel site selon le besoin tout en prenant en compte la notion de temps lors du traitement de l'information. Ajoutant la possibilité d'intégrer des outils sophistiqués d'analyse linguistique et d'élaborer des cartographies dynamiques et interactives.

Pour conclure sur les agents intelligents, nous pouvons dire que ces derniers sont largement utilisés dans différents domaines, ils peuvent être caractérisés par leurs degrés d'autonomie, leur aptitude à collaborer, leur capacité à réagir.

Deux tendances se dégagent actuellement dans le développement des agents :

- Les agents mobiles qui permettent de réduire le trafic sur le réseau.
- Les multi-agents qui sont des systèmes regroupant des agents qui ont de compétences différentes et qui interagissent pour résoudre un problème.

VIII. CONCLUSION

Dans ce chapitre, nous avons souligné l'importance croissante apportée aux nouvelles approches à base d'agents, de systèmes multi-agents.

Nous nous sommes focalisés sur les différentes définitions d'agents, les caractéristiques, les différentes classifications, et finalement nous avons introduit les systèmes multi-agents.

Le chapitre suivant portera sur un des types d'agents très important grâce à leurs propriétés de mobilité, ils sont nommés les agents mobiles.

Chapitre II

Les agents mobiles

INTRODUCTION

Les réseaux à grande échelle, tel l'Internet donnent accès à des quantités de données, de services, de ressources logicielles et matérielles répartis. Dans un tel contexte, le développement et le déploiement d'applications demande des méthodes et des outils adaptés permettant de traiter les problèmes qui résultent de l'hétérogénéité (des données, du logiciel, du matériel), de la répartition des ressources et des utilisateurs, des volumes de données déplacés sur le réseau, de la concurrence, de la sécurité, etc.

En particulier, les applications doivent faire face à de fortes variations des conditions d'exécution (capacités physiques des réseaux et des machines, disponibilité des composants) et il n'est pas possible de faire l'hypothèse sur la qualité des services requis ou rendus. Ainsi les applications doivent être adaptables, si possible de manière dynamique.

Dans ce cadre, les concepts d'agents, de mobilité, et d'implémentation ouverte semblent du plus grand intérêt [33].

I. LE CODE MOBILE

Un « code mobile » est un code dont la source est un système distant, auquel on ne fera éventuellement pas confiance, mais qui sera exécuté localement. Ce concept a plusieurs noms dans la littérature : agents mobiles, code mobile, code téléchargeable, contenu exécutable, contenu actif, et bien d'autres. Tous se rapportent au concept de code téléchargé. Voici quelques exemples de codes mobiles:

- **Les applets Web**

Ce sont de petits programmes écrits principalement en JAVA qui sont automatiquement téléchargés par le navigateur Web puis exécutés à l'affichage d'un document HTML. Un document contient une ou plusieurs applets dont le code source peut résider sur un ou plusieurs serveurs.

- **Les e-mails dynamiques**

Des e-mails pourraient contenir des scripts en Safe-TCL comme composants MIME. Ces scripts pourraient être exécutés soit lors de la réception du mail, soit lors de son ouverture.

- **Mécanismes anti-piratages**

Les entreprises de développement de logiciels se penchent sur les codes mobiles pour trouver une solution simple et efficace au piratage connecté à Internet.

I.1. Les catégories de code mobile

Il existe trois catégories de code Mobile [50]:

I.1.1. Évaluation distante

Dans une interaction par évaluation distante (Remote Evaluation ou **REV**), un composant client envoie un code à un autre site. Le composant récepteur exécute le code de l'application (ce code peut contenir des données), et délivre les résultats du service au composant émetteur [54]. L'unité d'exécution et les ressources sont fixes, seul le savoir-faire est mobile.

La figure II.1 illustre le schéma de l'évaluation distante.

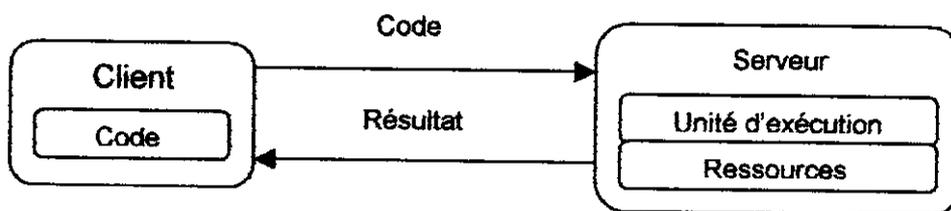


Figure II.1 : Schéma évaluation distante.

I.1.2. Code à la demande

Dans ce modèle du code à la demande (Code On Demand ou **COD**), le composant client interagit avec un composant distant afin de récupérer le code nécessaire à la réalisation d'un service. La requête ne contient pas de données paramètres à utiliser sur le site distant. Le composant distant fournit le code du service qui sera exécuté sur le site client (voir Figure II.2).

Ainsi, en permettant à des composants de récupérer du code, cette technologie permet d'étendre dynamiquement le comportement d'une application.

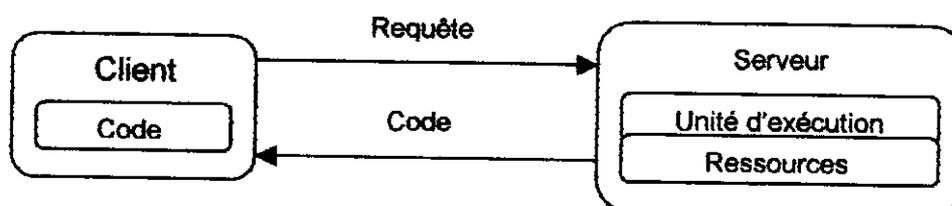


Figure II.2 : Schéma code à la demande.

I.1.3. Les agents mobiles

Les agents mobiles sont inspirés de travaux sur le calcul intensif initiés au sein de Xerox [55]. La notion d'agent mobile a été introduite pour la première fois en 1994 par White[35] qui décrit l'environnement Telescript. Dans cet environnement, des processus pouvaient se déplacer d'eux-mêmes d'un site du réseau à un autre pour interagir localement avec des ressources d'autres sites. Cette technologie est alors apparue comme prometteuse pour la conception d'applications distribuées[57, 58].

II. LES AGENTS MOBILES

II.1. Définition d'un agent mobile

Un agent mobile est un agent capable de se déplacer d'un site physique à un autre.

II.2. Les parties d'un agent mobile

Un agent est composé de son code correspondant à un algorithme, ainsi que d'un contexte incluant des données (Figure II.3) [43]. Ce contexte peut évoluer en cours d'exécution. Le code et le contexte de l'agent sont déplacés avec l'agent lorsque celui-ci visite différents serveurs. Lorsqu'un agent se déplace vers un serveur, il doit poursuivre son exécution sur le site destination.

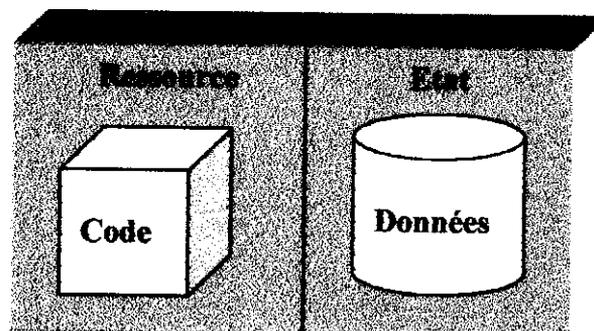


Figure II.3 : Les parties d'un agent mobile.

II.3. Le cycle de vie d'un agent mobile

Un agent possède plusieurs états par lesquels il peut passer, ces états constituent son cycle de vie :

- **Création** : Un agent est créé soit par sa plate-forme, soit par clonage.
- **Clonage** : Un agent peut être cloné de deux manières différentes :
 - ◇ Clonage local : dans le même environnement.
 - ◇ Clonage distant : un agent dans un environnement A se clone dans un autre environnement.
- **Dispose** : Un agent peut être détruit soit par programmation ou par l'environnement.
- **Stocké** : L'agent est arrêté, mais pas détruit.

- **Migration** : L'agent peut migrer d'un environnement A vers un autre environnement.
- **Communication** : L'agent peut communiquer avec d'autres agents, localement ou à distance.

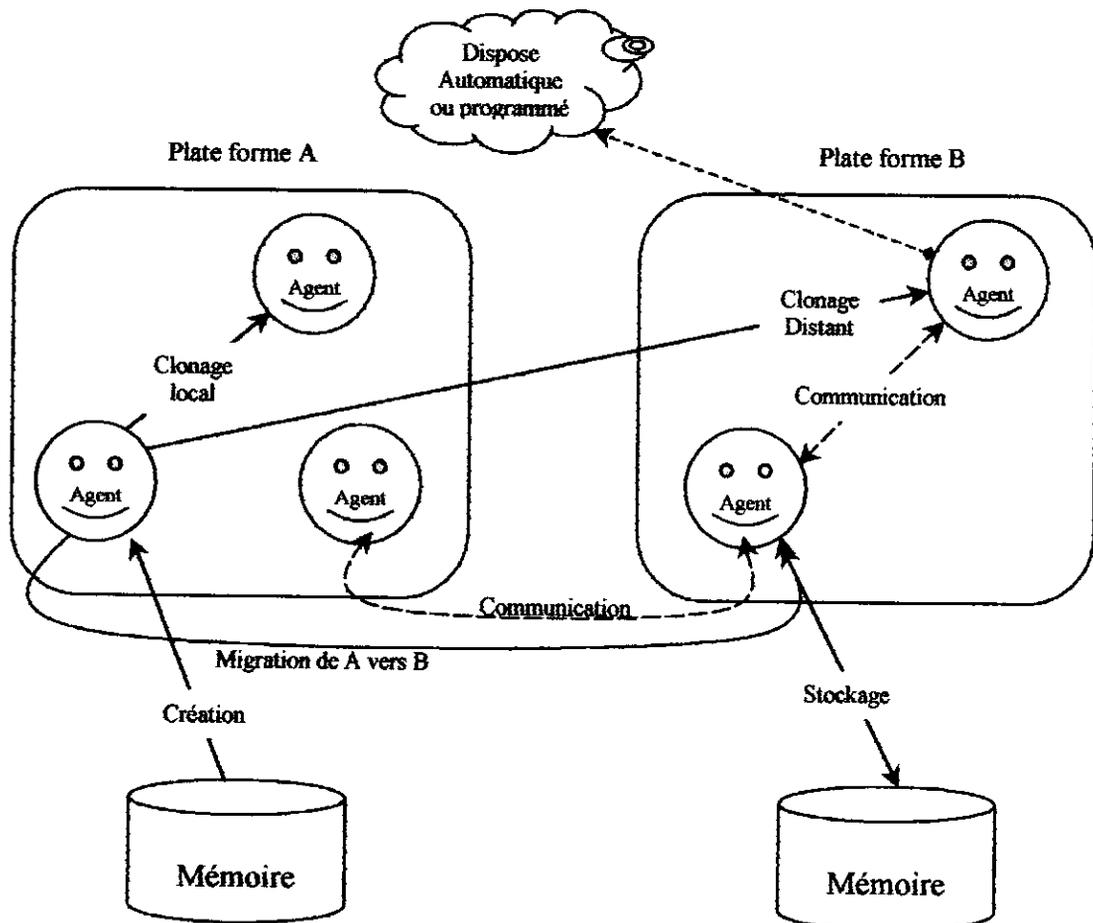


Figure II.4 : Cycle de vie d'un agent Mobile.

II.4. Particularités des agents mobiles

Les particularités des agents mobiles liées à la mobilité sont, sans être exhaustif: leur création (locale ou à distance), leur migration (forte/faible), et enfin leur mode de communication entre eux.

II.4.1. La création des agents mobiles

Dans la mesure où ces agents peuvent se déplacer, on peut se poser la question, lors de leur création, de l'endroit où l'on souhaite qu'ils démarrent leurs activités. En fait, cette propriété n'est pas directement liée à la mobilité, mais à des mécanismes sur lesquels s'appuie la mobilité.

II.4.2. La migration

La migration permet le transfert d'un agent d'un site à un autre à travers le réseau. Le problème principal vient du fait que les agents sont des entités autonomes (pro-actives) qui répondent à des sollicitations externes et internes.

De ce fait l'agent mobile peut :

- Etre sollicité de l'extérieur pour faire une migration (sur ordre de l'administrateur de la machine hôte avant un arrêt de la machine) alors qu'il est en train d'effectuer des traitements (Migration réactive).
- Ou la solliciter lui-même alors qu'il est sollicité de l'extérieur (Migration proactive).

Nous décrivons ci-dessous les différentes étapes de la migration d'un agent mobile :

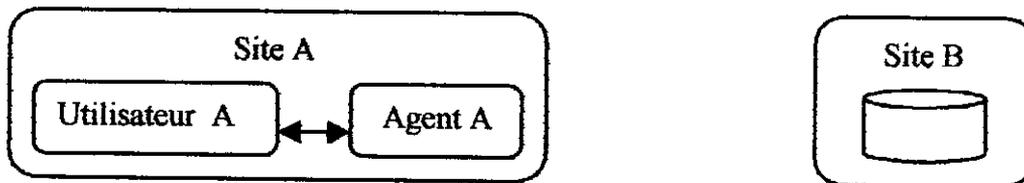


Figure II.5 : Phase 1.

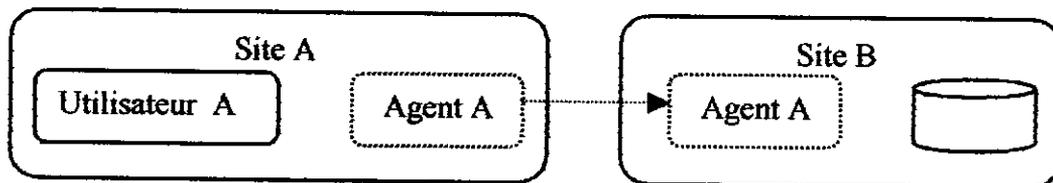


Figure II.6 : Phase 2.

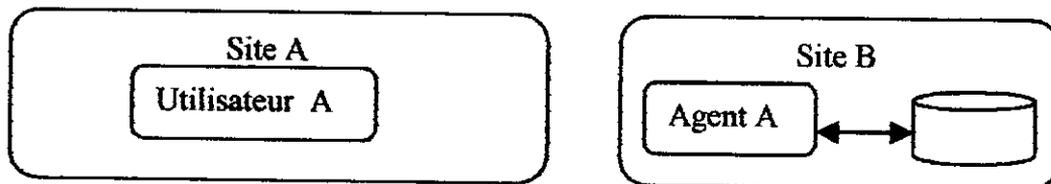


Figure II.7 : Phase 3.

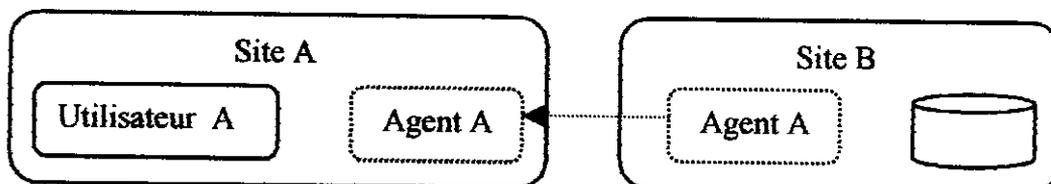


Figure II.8 : Phase 4.

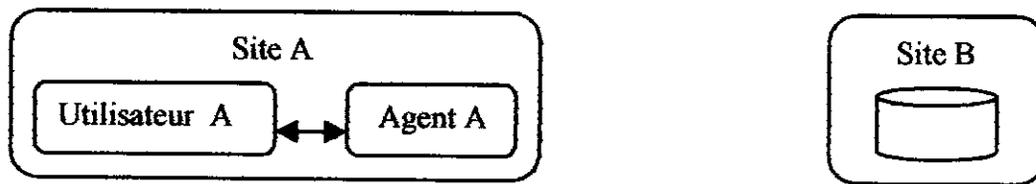


Figure II.9 : Phase 5.

Un utilisateur au niveau de site A veut récupérer des informations d'une base de donnée située au niveau de site B. Pour ce faire, il délègue un agent mobile A (phase1). L'agent A se déplace vers le site B (Phase2), et interroge la base de donnée appropriée (Phase3). Lors de la récupération des résultats, il revient au site A (Phase4) et retourne les résultats rapportés à l'utilisateur (Phase5).

Deux types de migrations ont été proposés dans les plates-formes existantes :

II.4.2.1. La migration forte

Où la totalité de l'agent migre vers le nouveau site, pour cette migration réelle, l'agent est suspendu avant d'être transféré. Une fois arrivé sur le site distant, il redémarre son exécution au point de contrôle précédent, en conservant l'état du processus.

Migration forte = Code + état d'exécution + Données paramètres

Une autre possibilité proposée consiste à stopper l'exécution de l'agent avant la migration puis d'en créer une copie distante identique sur le site distant (migration par réplication).

II.4.2.2. La migration faible

Où seul le code et les données migrent.

Migration faible = Code + Données paramètres

Une fois arrivé sur le site distant, l'agent est ré exécuté. Le programmeur doit donc préserver, dans les données, les informations d'état permettant la poursuite logique de l'exécution.

La migration forte, bien que beaucoup plus exigeante à implanter que la migration faible [53], n'en est pas moins indispensable pour toutes les applications pour lesquelles les notions de fiabilité et de tolérance aux pannes sont primordiales.

II.4.3. La communication entre agents

On peut identifier un grand nombre de moyens de communication entre agents, mais on peut les classer suivant qu'il s'agit d'une communication entre deux agents ou entre un agent et un groupe d'agents (groupe au sens large, c'est-à-dire soit un groupe formellement identifié en tant que tel, soit un ensemble d'agents non formellement identifiés comme par exemple des agents présents à un moment donné dans un lieu donné).

II.5. Principe de fonctionnement des agents mobiles

Un agent mobile peut se déplacer d'un site à un autre en cours d'exécution pour accéder à des données ou à des ressources. Il se déplace avec son code et ses données propres, mais aussi avec son état d'exécution. L'agent décide lui-même de manière autonome de ses mouvements.

Ainsi la mobilité est contrôlée par l'application elle-même, et non par le système d'exécution comme dans le cas de la migration de processus. La figure II.10 illustre ce principe.

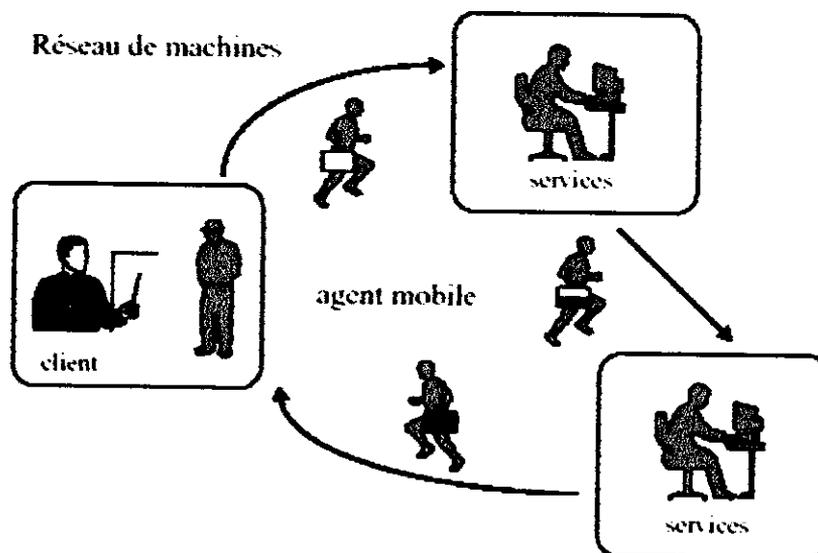


Figure II.10 : Fonctionnement d'un agent mobile [49].

Un client donne une mission à un agent qui pour la réaliser se déplace dans le réseau de machines accédant localement aux services offerts par ces machines. On peut distinguer trois phases :

- L'activation de l'agent mobile avec la description de sa mission.
- L'exécution de la mission par l'agent qui se déplace pour accéder aux services.
- La récupération éventuelle des résultats de l'agent mobile.

II.6. Les types d'agents mobiles

Il existe trois types d'agents mobiles :

II.6.1. Les agents de notifications

Ils attendent une information ou un événement et préviennent l'utilisateur ou déclenchent une action (exemple : surveillance de connexion ou d'activité sur une machine . . .).

II.6.2. Les agents itinérants

Ils réalisent une suite d'interactions avec les serveurs. On privilégie, par exemple, les accès locaux. Par exemple, l'agent va consulter un premier serveur qui lui fournit une liste de livres concernant les agents et il va ensuite consulter un deuxième serveur qui lui fournit une liste de prix. L'agent réalise ensuite la jointure.

II.6.3. Les agents d'adaptation

Ils étendent les fonctions du service, ou s'adaptent aux besoins spécifiques des clients. Par exemple, un client demande un document à un serveur et lui fournit également un algorithme de compression et de chiffrement. L'agent de compression et de cryptage s'exécute sur le serveur et renvoie au client le document chiffré et compressé.

III. AGENT MOBILE VERSUS MIGRATION DE PROCESSUS

III.1. Processus

Le processus est un concept clé de tous les systèmes d'exploitation. La notion de processus fournit un modèle pour représenter l'activité résultante de l'exécution d'un programme sur une machine [47].

III.2. Migration de processus

La migration de processus est le déplacement d'un processus en cours d'exécution d'une machine source vers une machine destination, ces deux machines étant reliées par un réseau de communication et n'utilisant pas de mémoire partagée.

La migration de processus d'une machine source vers une machine destination consiste à interrompre le processus qui s'exécute sur la machine source pour extraire son état d'exécution, à transférer cet état vers la machine destination, à créer, sur la machine destination, un nouveau processus auquel on affectera l'état d'exécution transféré et à mettre à jour les liens de communication avec les autres processus. Une fois ceci fait, le processus sur la machine source doit être détruit tandis que le processus sur la machine destination est lancé et représente le processus déplacé [48].

De nature, les agents mobiles traitent la distribution en interne. Un agent mobile est donc un processus migrant volontairement. Il peut se déplacer de sites en sites en suivant un itinéraire en fonction des tâches à réaliser.

Le tableau ci-dessous illustre la différence entre les agents mobiles et la migration de processus.

Tableaux II.1 : Les agents mobiles versus la migration de processus.

	Migration de processus	Agents mobiles
But de la migration (1)	Partage de charge	Naviguer en fonction de l'application
But de la migration (2)	Survivance aux arrêts de la machine	Supporter des opérations déconnectées
Initiative	Système d'exploitation	Application
Prise en compte de la répartition	Implicite	Explicite
Portée	Lan	Wan

IV. LES SYSTÈMES D'AGENTS MOBILES EXISTANTS

En 1994, la société General Magic [35] a proposé le premier système supportant les agents mobiles. Ce système propriétaire était peu adapté à un environnement tel que l'Internet. L'intérêt suscité par la technologie à base d'agents mobiles a contribué ensuite à un nombre important de développements de plates-formes. Une liste [36] répertorie plus d'une soixantaine de plates-formes à ce jour, certaines d'entre elles sont utilisées dans le cadre de travaux de recherche. D'autres sont fournies sous la forme de produits commerciaux. Ci-dessous, on décrit les systèmes les plus réponsus.

Odyssey ([21])

Odyssey est le successeur de Telescript, développé par General Magic, basé sur JAVA et il en reprend les concepts fondamentaux à savoir les notions de places, tickets et autorités. Il est largement inspiré par Telescript, mais n'a pas exactement la même fonctionnalité.

De plus, il est indépendant du protocole de transport utilisé et peut donc s'adapter à RMI (Sun), CORBA (OMG).

Le système des Aglets ([22])

Le « *Aglet Workbench* » d'IBM est sans doute un des systèmes les plus connus, utilisent la notion de référence distante (AgletProxy), le passage de messages (multicast, broadcast) et le protocole ATP développé par IBM (Agent Transfert Protocol [23]). Il offre en outre une certaine notion de sécurité en limitant les ressources allouées aux aglets.

Concordia ([24])

Concordia est un système basé sur java, développé par le Horizon Systems Laboratory de Mitsubishi [41], il se caractérise par sa simplicité mais aussi par une faible communication entre les agents, et un mécanisme de sécurité faible.

Voyager ([25])

Il a été développé par la société *objectspace*, il semble être le système d'agent le plus complet puisqu'il intègre entre autres les références distantes, le support CORBA, et un ramasse miettes distribué.

Agent Tcl (D'Agent)(<http://www.cs.dartmouth.edu/~agent>)

Agent-Tcl est l'un des premiers systèmes d'agents mobiles (première publication en 1995), initialement écrit en Tcl, un langage portable conçu au Dartmouth College. Il a été depuis réécrit pour supporter divers langages - Tcl, Java et Scheme -. D'Agent est le nouveau nom d'Agent-Tcl.

Telescript [37]

Il a été développé par la société américaine *General Magic* proposant un système fermé pour le commerce électronique, il était un des seuls systèmes supportant le transfert de l'état d'exécution des agents[42].

Mole (www.informatik.uni-stuttgart.de/ipvr/vs/projekete/mole.html)

Le groupe dédié aux systèmes distribués, au sein de l'Institute of Parallel and Distributed High-Performance Systems à l'Université de *Stuttgart*, dirigé par *Kurt Rothermel*, était le premier à développer une plate-forme basée sur JAVA supportant les agents mobiles. Récemment, une troisième version de leur système, appelé *Mole*, a été publiée sur le Web. *Mole* est une plate-forme à agents mobiles à usage général.

IV.1. Comparaison de quelques systèmes étudiés

Le tableau ci-dessous illustre la comparaison entre quelques systèmes étudiés :

Tableaux II.2 : Comparaison de quelques systèmes étudiés

	Mobilité	Langage
Telescript	Migration forte	Proche de LISP
Aglets	Migration faible	Java
Agents TCL	Migration forte	Script TCL
Voyager	Migration faible	Java
Concordia	Migration faible	Java

V. JAVA: LANGAGE DE PROGRAMMATION POUR AGENTS MOBILES

Plusieurs langages supportent la mobilité du code [39] (par exemple : Java, Telescript), les plates-formes à agents mobiles s'appuient sur ces langages.

Toutefois, le langage java reste aujourd'hui le plus répandu pour le développement d'applications distribuées, en partie du fait que :

- La machine virtuelle java est portée sur la plupart des systèmes courants.
- Elle permet une indépendance envers le réseau physique et les systèmes d'exploitation des sites hôtes.
- En java, les agents peuvent migrer entre des sites hétérogènes.
- Java permet la mobilité du code et des objets grâce au mécanisme de sérialisation[56].

VI. AVANTAGES ET INCONVÉNIENTS DES AGENTS MOBILES

VI.1. Avantages des agents mobiles

La mobilité d'agent permet de rapprocher client et serveur et en conséquence de réduire le nombre et le volume des interactions distantes (en les remplaçant par des interactions locales), de spécialiser des serveurs distants ou de déporter la charge de calcul d'un site à un autre.

La mobilité du code offre un premier niveau de flexibilité aux applications, et elle ne se substitue pas aux capacités de communication des agents (la communication distante reste possible) mais les complète; afin de satisfaire aux contraintes des réseaux de grande taille ou sans fil (latence, non-permanence des liens de communication), les agents communiquent par messages asynchrones.

On va donner ci-dessous des exemples sur les avantages cités précédemment :

VI.1.1. Exemple sur la réduction de la charge réseau

Un utilisateur au niveau de poste C (application) veut récupérer des numéros de téléphone de quelques personnes, la liste des noms se trouve au niveau de poste A (service) et celle des numéros de téléphone au niveau de poste B (annuaire).

La figure II.11 illustre l'efficacité de l'utilisation des agents mobiles par rapport au modèle client / serveur.

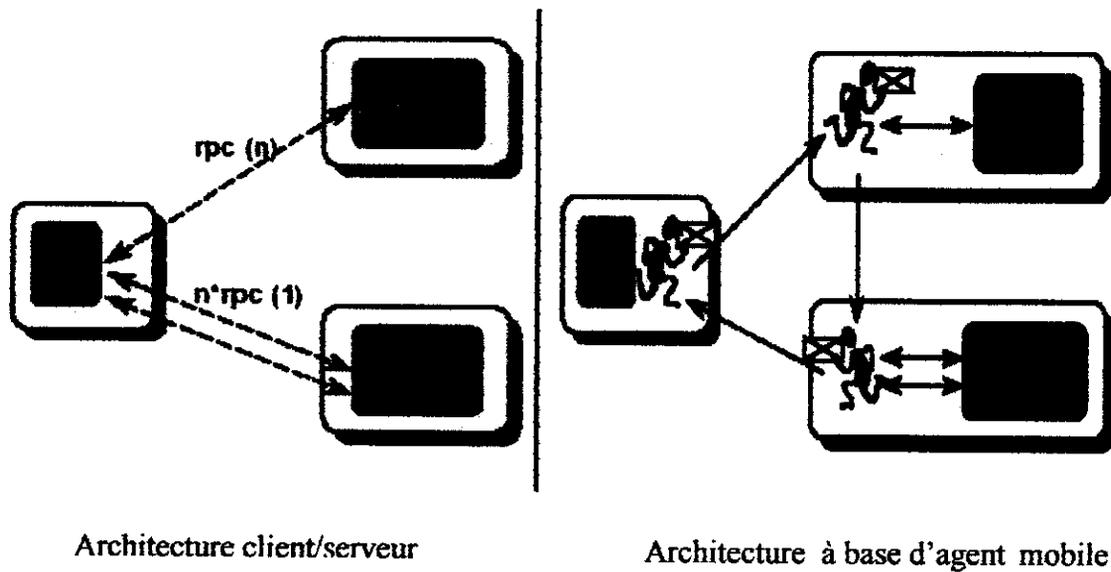


Figure II.11 : Réduction de la charge réseau.

VL1.2. Exemple sur l'extensibilité des fonctions d'un serveur

Le Web est très souvent utilisé pour récupérer des documents de toute sorte. Nous considérons dans cet exemple l'accès à un serveur exportant une interface permettant à un client d'obtenir une copie d'un document.

Le client demande au serveur un document de nom donné et récupère un flot de données correspondant au contenu du document.

Cependant, si le client désire optimiser ce transfert de données en utilisant un outil de compression des données, il est nécessaire d'étendre le serveur pour y ajouter cette fonction de compression. De plus, différents clients peuvent utiliser différents outils de compression, nécessitant des extensions particulières pour chaque client.

Le paradigme d'agents mobiles offre une réponse élégante à ce problème. En effet, le client peut envoyer un agent sur le site du serveur. Cet agent interroge le serveur pour obtenir le document, compresse le document obtenu et revient sur le site du client avec le document compressé. Sur le site client, l'agent peut décompresser le document et le délivrer au client. De cette manière, le client réalise une extension dynamique du serveur en fonction de ses besoins spécifiques sans changer l'interface de ce dernier.

La figure II.12 explique brièvement cet exemple [43].

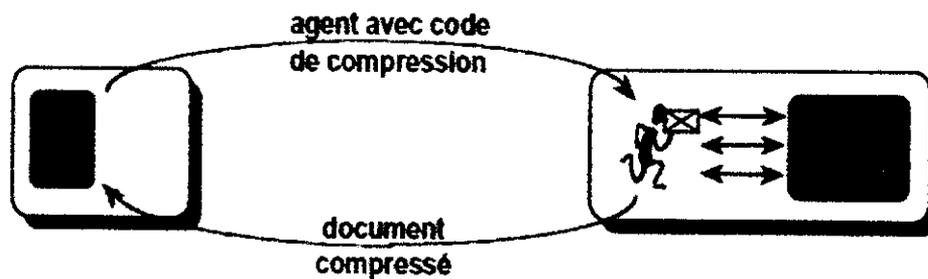


Figure II.12 : Extensibilité des fonctions d'un serveur.

VL2. Les inconvénients des agents mobiles

- Manque d'applications où les agents mobiles apportent un avantage certain.
- Manque d'infrastructure et de standards : aucun standard n'est réellement appliqué et leur conception varie encore beaucoup d'un système à l'autre malgré les efforts d'organismes normalisateurs comme la (FIPA). De même, les difficultés non résolues empêchent la formation d'une infrastructure suffisamment fournie pour permettre le développement d'applications économiquement intéressantes.
- 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.

VII. CONCLUSION

Les agents mobiles, nouveau paradigme, semblent ouvrir une nouvelle voie pour la conception des futures applications réparties grâce aux multiples avantages qu'ils offrent, notamment leurs qualités asynchrones et leur utilisation modérée des ressources du réseau.

D'où découle l'objectif de chapitre suivant : pouvoir impliquer les agents mobiles dans le commerce électronique.

Chapitre III

Applications réparties et commerce électronique

INTRODUCTION

Les réseaux à grande échelle, tel l'Internet, donnent accès à des quantités de données, de services, de ressources logicielles et matérielles répartis, et provoquent l'apparition de nouvelles applications telles que : commerce électronique, courrier électronique, recherche d'information, etc.

Dans ce chapitre nous aborderons la notion d'application répartie, nous donnerons quelques exemples d'applications réparties et nous étudierons en détail l'une des applications les plus importantes : *le commerce électronique*.

I. APPLICATION RÉPARTIE :

I.1. Définition d'une application répartie :

Une application informatique est dite *répartie* lorsqu'elle met en jeu des parties qui s'exécutent sur plusieurs machines reliées par un système de communication. Les premières applications réparties ont été développées en utilisant directement les mécanismes de bas niveau (messages) fournis par le système de communication. A partir de ce stade initial, les efforts ont principalement visé à :

1. Dissimuler autant que possible la répartition, pour se ramener aux schémas connus de programmation centralisée;
2. Fournir des mécanismes d'un plus haut niveau d'abstraction que les messages, pour la communication et pour le partage d'information;
3. Faciliter la maintenance des applications, et en particulier leur évolution pour s'adapter aux nouveaux besoins et aux nouveaux environnements;
4. Faciliter la réutilisation des applications et parties d'applications existantes.

La poursuite des objectifs (1 et 2) a fait l'objet de nombreux travaux depuis les années 80. Elle a conduit à l'émergence de plusieurs modèles d'organisation des applications (client-serveur, communication par événements, partage d'objets) et des mécanismes nécessaires à leur mise en oeuvre (appel de procédure à distance, bus logiciel, mémoire virtuelle répartie).

I.2. Caractéristiques des systèmes répartis

Un système informatique réparti peut être caractérisé par les propriétés suivantes :

- Le système est constitué d'un ensemble de composants matériels (ordinateurs, organes d'entrée-sortie, processeurs spécialisés, dispositifs de commande ou de mesure, etc.), interconnectés par un réseau de communication.
- Les composants du système ne fonctionnent pas de manière indépendante, mais coopèrent pour l'exécution de tâches communes.
- Le système peut continuer à fonctionner (éventuellement en mode dégradé) malgré les défaillances partielles des composants ou du réseau de communication.

I.3. Quelques exemples d'applications réparties

Nous présentons une brève description de cinq classes d'applications réparties. Sans couvrir l'ensemble du domaine, ces exemples sont représentatifs d'une catégorie assez large d'applications, dans lesquelles la répartition joue un rôle intrinsèque.

I.3.1. Applications à flot de données

On désigne sous le nom de flot de données ("*workflow*") une classe d'applications caractérisées par l'exécution coordonnée d'un ensemble de tâches sur différents organes de traitement. La coordination spécifie d'une part le mode de séquençement des tâches, d'autre part les données échangées entre les tâches. Les applications à flot de données ont initialement été introduites pour automatiser les procédures administratives dans lesquelles des documents circulent entre divers services, l'arrivée d'un document pouvant déclencher des actions, et les documents pouvant eux-mêmes être modifiés.

I.3.2. Édition coopérative

Un éditeur coopératif permet à plusieurs auteurs géographiquement distribués de coopérer à la rédaction d'un document. Les documents sont structurés : la structure d'un document définit une organisation en parties (chapitres, sections, etc.). Le document commun est un objet abstrait, dont chaque utilisateur peut avoir une vue différente. Ces vues diffèrent par :

- Les parties (ou fragments) du document qui sont rendues visibles à l'utilisateur,
- Les droits de l'utilisateur sur chacun de ces fragments (lecture, écriture, etc.).

Pour chaque fragment du document, un utilisateur joue le rôle de gérant, et peut attribuer des droits aux autres utilisateurs sur ce fragment. Le découpage du document et les droits des utilisateurs peuvent être redéfinis dynamiquement, en respectant des contraintes fixées (par exemple un seul gérant et un seul rédacteur par fragment).

I.3.3. Téléconférence

Un système de téléconférence vise à permettre à un ensemble d'utilisateurs géographiquement répartis de participer à une "réunion virtuelle", dans laquelle on souhaite restituer aussi fidèlement que possible des conditions d'une réunion réelle. Les utilisateurs communiquent entre eux via des informations partagées accessibles à tous (tableau blanc, documents partagés), ainsi que par des canaux audio et vidéo.

Le système doit remplir les fonctions suivantes :

- Présenter à chaque utilisateur une image de l'environnement commun,
- Fournir à chaque utilisateur une interface permettant de manipuler l'environnement commun et une interface de communication,
- Fournir des fonctions de coordination entre les utilisateurs (protocole d'arrivée, de départ, de gestion du temps de parole, etc.),
- Assurer la gestion cohérente des données partagées.

I.3.4. Télévision interactive

Un réseau de télévision interactive vise à fournir à ses clients un ensemble de services : diffusion de programmes de télévision choisis à la demande, télé-achat, jeux interactifs. Chaque client (abonné au réseau) dispose d'un terminal TV (*set-top box*) qui lui présente une interface analogue à celle d'un poste de télévision ordinaire ; le boîtier de télécommande possède une fonction de pointage qui permet l'utilisation de menus.

I.3.5. Commerce électronique

L'objectif d'un système de commerce électronique est de permettre à un ensemble de clients et de fournisseurs connectés à un réseau d'exécuter des transactions commerciales : recherche d'un produit ou d'un service par consultation de catalogues, commande simple ou groupée, paiement, demande de livraison, etc.

I.4. Formes d'interactions entre les entités d'une application répartie :

Les interactions entre les entités qui composent une application peuvent prendre des formes très diverses, dont nous donnons quelques exemples.

I.4.1. Client –serveur :

Le schéma client-serveur a été l'un des premiers modèles proposés pour la structuration des systèmes répartis et il reste le plus répandu. Il met en jeu deux entités : un *client*, qui demande l'exécution d'un service, et un *serveur*, qui réalise ce service. Le schéma de base est celui de l'exécution synchrone : un processus client envoie une requête à un serveur et se bloque, en attente de la réponse ; le serveur exécute le service demandé et renvoie la réponse. L'appel de procédure à distance est un outil qui permet de faciliter la réalisation du schéma client-serveur.

I.4.2. Serveurs coopérants :

Pour augmenter les performances et la disponibilité d'un service, celui-ci peut être réalisé par plusieurs serveurs coopérants. Divers schémas peuvent être utilisés pour cette coopération : maître-esclaves, groupe de serveurs utilisant un protocole de groupe pour se coordonner entre eux, et désignés ou non par un nom unique.

I.4.3. Objets partagés:

Un mode indirect d'interaction entre plusieurs processus répartis est l'utilisation d'objets partagés. Les schémas diffèrent par leurs modalités de mise en oeuvre : protocole d'accès aux objets; mode de désignation des objets (nom interne universel ; nom symbolique avec passage par un service de désignation, etc.) ; persistance ou non des objets.

I.4.4. Flots de communication

Le mode direct de communication utilise des flots de communication, parmi lesquels on peut distinguer des flots discrets et des flots continus. Des exemples de flots discrets sont les messages utilisés pour le transfert de fichiers, le courrier électronique, la mise à jour de données, la coordination entre tâches. Des exemples de flots continus sont ceux utilisés pour transporter des images animées ou de la voix.

I.4.5. Code mobile

Un autre mode d'interaction consiste à transporter avec des données les programmes nécessaires à leur traitement. On évite ainsi d'être tributaire de l'existence de ces programmes sur le lieu de réception, ce qui permet de déterminer au dernier moment le lieu d'exécution des programmes. L'hétérogénéité des machines peut être masquée par la réalisation d'une machine virtuelle capable d'interpréter le code transporté.

I.4.6. Agents

Enfin, on peut noter le recours à des "agents", programmes qui agissent pour le compte d'une personne ou d'une organisation.

II. COMMERCE ÉLECTRONIQUE

L'arrivée du troisième millénaire est caractérisée par une révolution technologique autrefois inimaginable. Cet ensemble de modifications informatiques et électroniques se fait ressentir sur l'économie et plusieurs autres domaines.

Dans cette section nous aborderons la notion de e-commerce.

II.1. Définition du commerce électronique

Il n'y a pas de définition universelle du commerce électronique en raison du grand nombre de marchés et d'acteurs sur Internet et de l'évolution rapide de leurs rapports complexes. [38]

Le commerce électronique est un phénomène grandissant qui est observé avec intérêt par de nombreux milieux d'affaires. Réduit à sa plus simple expression le commerce électronique consiste en l'utilisation de moyens électroniques pour échanger de l'information, mener à bien des activités et effectuer des transactions [44]. Il s'applique tant à la vente de produits qu'à la prestation de services. Il peut même aller jusqu'à la création des produits et services qui sont intangibles (ex : des livres). Ces échanges sont effectués par deux grandes catégories d'entités : les organisations (les entreprises et le gouvernement) et le consommateur/individu.

II.2. Les origines du commerce électronique

En 1838, le télégraphe inventé par Morse a permis de réaliser de nombreuses transactions commerciales et c'est en 1910 que l'activité commence à se structurer avec la vente de bouquets de fleurs par une firme américaine FTD Inc. (Florist Telegraph Delivery Group). Cette dernière entreprise est pionnière dans l'utilisation du commerce électronique par voie des télégraphes [45].

En 1991, le gouvernement américain annule les dernières restrictions sur l'utilisation d'Internet à des fins commerciales et Tim Berners-Lee à Genève développe la première application graphique qui permet aux ordinateurs de se connecter au réseau Internet. Ce genre d'application est connu sous le nom de *browser*. De nombreux pionniers dans la scène du e-commerce sont apparus, tels que AMAZON.COM qui se dédie à la vente en ligne de livres et CD de musique [45].

II.3. Typologie du commerce électronique

Le commerce électronique prend plusieurs formes. Il existe une terminologie américaine pour les désigner : les e-business qui prend les formes suivantes :

Le B to B (*Business To Business*, parfois noté *B2B*)

Désigne les activités commerciales entre entreprises, destinées aux professionnels.

Il est important de mentionner que le commerce électronique « interentreprises » (B2B) domine nettement et représente actuellement 80 % des activités recensées. [60,73]

Le C to B (*Consumers To Business*, parfois noté *C2B*)

Représente le regroupement de clients entre eux, pour des achats en gros, en communauté, afin de bénéficier de prix dégressifs intéressants. [73]

Le-B-to-C (*Business-To-Consumers*, -parfois-noté-*B2C*)

Désigne une relation commerciale d'entreprise à particulier (consommateur final). Ce type d'échange est très connu mais il est moins important que le B2B. [73]

Le C-to-C (consumers-To-Consumers, parfois noté C2C)

Désigne une relation commerciale entre particuliers. C'est une extension du B2C. [73]

Le B To E (Business To Employees, parfois noté B2E)

Désigne la relation entre une entreprise et ses employés.

Le B To A (Business To Administration, parfois noté B2A)

Regroupe les échanges des entreprises avec l'administration.

II.4. L'achat en ligne

Faire du commerce électronique sur Internet est très semblable à la façon de faire du commerce dans la vie de tous les jours, chez un commerçant local, dans un supermarché ou encore dans un grand magasin. Les grandes étapes d'un achat peuvent être résumées de la façon suivante. [61]

- Il y a tout d'abord la phase d'**achat** au cours de laquelle l'acheteur parcourt différents magasins à la quête d'un ou plusieurs articles. Il va consulter plusieurs magasins, comparer plusieurs articles et, éventuellement, chercher des conditions d'achat intéressantes.
Internet permet de parcourir l'ensemble des magasins et d'examiner leurs articles sans qu'il soit nécessaire de se déplacer.
- Une fois la décision d'achat prise avec ses conditions connues, l'acheteur procède à la **commande**. L'acheteur et le vendeur doivent se mettre d'accord sur le mode de paiement : comptant, carte bancaire, carte de crédit privative, chèque, etc.
- Le mode de paiement ayant été choisi, l'acheteur procède au **paiement** proprement dit. Il doit vérifier le montant, donner les informations de paiement (pour une carte de crédit, le numéro de carte et la date d'expiration) et signer son paiement de façon électronique. Ce paiement, suivant son montant, peut faire appel à une demande d'autorisation auprès de la banque de l'acheteur.
- Le paiement, ou son autorisation, ayant été effectué, le vendeur doit assurer la **livraison** de la marchandise si celle-ci ne peut pas être distribuée sous forme électronique, comme c'est souvent le cas.

II.5. Modes de paiement

On retrouve sous Internet les différents modes de paiement existant dans la vie courante aujourd'hui, notamment dans la vente à distance.

Le paiement peut bien sûr être fait à réception de la marchandise (contre remboursement). Le colis contenant l'achat n'est délivré qu'en échange d'un chèque de paiement. Dans cette partie, nous nous intéressons seulement aux **paiements effectués en ligne** qui peuvent prendre les formes suivantes : [61]

- Le paiement comptant sous forme d'argent virtuel stocké dans un porte-monnaie électronique. Celui-ci se comporte comme un véritable porte-monnaie : une fois rempli, auprès d'une banque par exemple, on peut procéder à des achats de faible ou moyenne valeur ;
- Le chèque peut être utilisé pour n'importe quel montant. Ce document atteste que l'on dispose d'un crédit auprès d'une banque ; une fois rempli avec son montant, il doit être signé. Il existe un équivalent électronique de la signature dont la fiabilité est bien meilleure que la signature manuelle ;
- La carte bancaire ou la carte de crédit. C'est un moyen de paiement de plus en plus courant grâce notamment à sa facilité d'utilisation.

II.6. Le problème du paiement sur internet

Les paiements sont notamment limités par les lois du pays qui n'autorisent pas forcément le libre chiffrage des informations. Les internautes sont encore très frileux pour la consommation sur Internet, car ils ne savent pas ce qu'ils font de leur numéro de carte de crédit lorsqu'ils le donnent.

En règle générale, les sites de vente proposent soit un paiement traditionnel (par chèque), soit un paiement en ligne (par carte de crédit). Les inconvénients du paiement traditionnel sont évidents en terme de délais et d'échange de devises avec les pays étrangers. [62]

Deux possibilités existent pour le paiement en ligne :

- La première possibilité est le porte-monnaie électronique, qui est géré par un organisme tiers et qui correspond à une carte virtuelle sur laquelle on dépose de l'argent. Cette solution est généralement utilisée pour les produits de faible coût.
- La deuxième solution est le paiement directement avec la carte de crédit, comme tout autre achat. C'est là que les problèmes de sécurité commencent et que la peur des consommateurs se fait sentir.

II.7. La sécurité du paiement

Les risques sont multiples. Le commerçant peut modifier le montant à débiter ou vendre un produit qui n'existe pas et que le client ne recevra jamais. Le client, lui-même, peut utiliser une carte qui n'est pas la sienne, contester avoir passé une commande ou avoir un découvert à la banque.

Il s'agit donc de sécuriser les échanges en s'assurant qu'ils sont chiffrés (confidentialité), que ceux qui y participent sont bien ceux qu'ils disent être (authentification), que les données n'ont pas été modifiées (intégrité). Il faut également pouvoir certifier que les échanges ont bien eu lieu (non-répudiation) et que le client peut payer. [62]

Il existe plusieurs mécanismes pour assurer une certaine sécurité :

SSL : Secure Socket Layers : c'est de loin le plus utilisé, il assure le chiffrement des échanges mais ne garantit pas que le marchand va vous livrer, ni que le client peut payer.

SET : Secure Electronic Transaction : chiffrement des données de la carte de crédit, signature des messages et authentification des différents acteurs de l'échange.

C-SET : Chip Secure Electronic Transaction : C'est une extension de SET avec un lecteur de carte. Ces deux systèmes sont compatibles, mais C-SET permet de contrôler davantage de chose de façon physique (vérification de la carte, etc...). Ce système est aussi sûr qu'un paiement par carte bancaire dans un magasin.

D'autre mécanisme de sécurité existe mais ne devrait pas être utilisés pour le paiement.

II.8. Avantages du commerce électronique

Le commerce électronique est l'une des applications les plus importantes. Les possibilités et les avantages de ce médium électronique sont nombreux.

Pour l'entreprise productrice de biens et services (B2B) ou encore pour le commerçant (B2C), le commerce électronique : [63]

- Permet la réduction des coûts opérationnels :
Les coûts d'exploitation (marketing, production et inventaire), de distribution et de livraison peuvent être réduits significativement avec l'usage du commerce électronique.
- Augmente l'efficacité en raison des gains en précision et en rapidité.
- Offre un accès aux marchés locaux, nationaux ou internationaux.
- Permet d'avoir une offre de produits et services personnalisés.
- Permet d'offrir un marketing spécialisé en raison des nombreuses banques de données disponibles.

Pour l'entreprise consommatrice de produits ou encore le consommateur, le commerce électronique :

- Donne accès à un plus grand choix de produits et services.
- Donne accès à des produits à des prix moins élevés.
- Procure une certaine commodité pour faire les transactions ou obtenir de l'information (éliminant des déplacements inutiles, etc.).
- Assurer un service de qualité, 7 jours sur 7, 24 heures sur 24.

II.9. L'avenir du commerce électronique

La révolution du savoir et le commerce électronique transformeront la société et son organisation. Voici les grandes orientations : [64]

- Création d'entreprises virtuelles.
- Modification des relations entre l'entreprise, ses clients et ses fournisseurs.
- Modification du processus d'achat des consommateurs et des entreprises.
- Adoption de masse de la facturation en ligne et de leur gestion par les consommateurs.
- Augmentation du travail à distance et du partage d'information via Internet.

II.10. Utilisation des agents dans le commerce électronique

Le commerce électronique devient de plus en plus une réalité. Or, à chaque fois qu'un marchand ou un client souhaite d'échanger des produits via l'Internet, il épuise plusieurs heures pour trouver une bonne solution car l'information est décentralisée, et les champs d'expertise sont fonctionnellement distribués dans différents domaines. Par exemple, afin de vendre un produit, on doit tenir compte de la connaissance du domaine auquel il appartient, les stratégies de la société commerciale qui le vend, et de bien d'autres facteurs.

D'autres part, le commerce électronique est quasiment fait par nos interactions: c'est nous qui décidons quand nous allons acheter des produits ou combien nous souhaitons de payer pour chaque produit, etc. Donc, la négociation joue un rôle important pendant les échanges des produits ou des services.

En résumé, il nous faut un système qui cherche les produits auxquelles nous nous intéressons (une classification selon certains critères), et qui négocie avec les différents marchands d'une manière automatique. Aujourd'hui, seules les applications agents correspondent à ce besoin.

II.10.1. Catégories d'agents pour le commerce électronique

On distingue ici deux types d'agents :

- Des agents qui interviennent coté client (agents acheteurs),
- Des agents qui interviennent coté vendeur (agents vendeurs).

II.10.1.1. Les agents acheteurs

L'intervention des agents coté clients (agents acheteurs) permettra au client de comparer les prix et d'acheter des produits à des prix avantageux.

Les agents acheteurs permettent au consommateur d'effectuer des recherches par marque, par catégorie ou encore par distributeur.

L'idée, à terme, est de fournir une présélection automatique d'articles en fonction des préférences du consommateur avec un comparatif sur les prix, fonctionnalités, services, et avis d'autres utilisateurs, etc.

Les agents acheteurs interviennent donc pour faciliter le processus D'achat [66]. Plus précisément ces agents peuvent nous renseigner : [67]

- Sur la disponibilité d'un produit en menant une recherche par marque ou par catégorie (produits + accessoires).
- Sur l'identification des distributeurs : localisation d'un distributeur précis, liste intégrale ou sélective de distributeurs (en fonction des services qu'ils offrent : garantie, facilité de paiement, ...).
- En traitant les informations collectées, par exemple grâce à des tableaux comparatifs des offres présentées sur divers critères (prix, services, avis d'autres consommateurs).
- En établissant une présélection automatique d'articles en fonction des préférences du consommateur (priorité au rapport qualité / prix, aux services, aux avis des autres consommateurs, ...).
- En réalisant la transaction :
 - De façon automatique (achat répétitif d'un panier de produits, achat dès qu'un modèle est en solde).
 - Ou semi-automatique : Recommandation/suggestion,

Exemples d'agents acheteurs :

- **BargainbotSearch Agent** : cet agent est spécialisé dans l'édition. A partir d'un titre de livre ou du nom de l'auteur, il adresse simultanément des recherches à libraires et présente de brèves fiches descriptives sur les références trouvées. [67]
- **The Movie Critic** : donnez votre avis sur 12 films et l'agent vous fera des recommandations sur les films que vous n'avez pas encore vus.
- **Firefly** : indiquez à cet agent vos goûts musicaux et cinématographiques et il vous fera lui aussi des recommandations assez pertinentes. [72]

II.10.1.2. Les agents vendeurs

Si les agents les plus spectaculaires sont ceux qui s'adressent aux consommateurs, il existe une autre famille d'agents, ce sont les agents les plus utiles aux directeurs marketing et commerciaux. Ceux qui analysent la demande globale pour adapter leur offre aux besoins du marché.

Ces agents permettent aux vendeurs la possibilité de rendre leur produit immédiatement visible sans avoir à investir dans des coûteux systèmes de gestion de base de données. Les agents vendeurs présentent les biens et les services aux clients (qu'ils considèrent comme des agents) et peuvent même être programmés pour négocier.

Un agent vendeur ayant un produit à commercialiser va traverser le réseau à la recherche des clients intéressés par ce produit. Lorsque l'agent vendeur rencontre un agent client intéressé par ce type de produits, une transaction est alors négociée entre les deux agents.

Les différentes fonctions assurées par les agents vendeurs sont : [68]

- Enregistrement du profil et des préférences de l'acheteur.
- Enregistrement des demandes successives de l'acheteur afin d'enrichir, d'affiner, de faire évoluer son profil.
- Calculer des recommandations sur l'évolution de l'offre commerciale grâce à des statistiques sur la demande globale des consommateurs.

Exemples d'agents vendeurs :

- **The BroadVision** : Créé en 1993, cette société se consacre au développement de systèmes intégrés de gestion des relations de marketing individualisé, ou marketing one-to-one.
- **SelectCast**
- **AgentWare**

II.11. Négociation en commerce électronique

La négociation est un processus permettant de s'accorder (diminution des incohérences, des incertitudes) sur des situations communes ou des plans par des échanges d'informations. Dans le cadre d'un système multi-agents, la négociation est largement utilisée pour : [70]

- L'allocation de tâches ou de buts,
- La résolution de conflits,
- La définition de structures organisationnelles,
- La cohérence du système,....la planification.

Comme décrit dans le schéma ci-dessous, dans un contexte de négociation l'acheteur et le vendeur définissent un prix optionnel. Le prix optionnel de l'acheteur est strictement inférieur au prix optionnel du vendeur. L'intervalle de négociation est l'intervalle qui sépare le prix optionnel de l'acheteur de celui du vendeur.

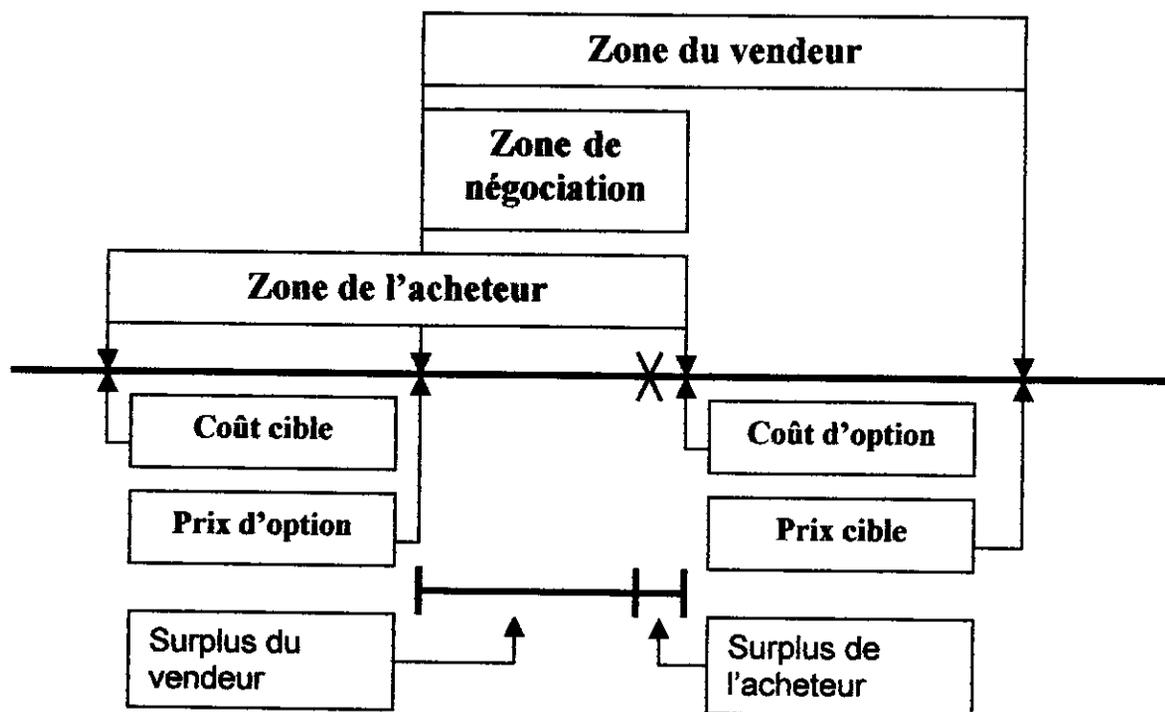


Figure III.1 : Zone de négociation entre un acheteur et un vendeur. [71]

III. PROBLÉMATIQUE

Aujourd'hui, la technologie "agent mobile" est de plus en plus attractive vu les avantages qu'elle offre pour la réalisation des tâches complexes et répétitives dans des systèmes ouverts et dynamiques. Néanmoins, l'utilisation de cette technologie dans des applications réelles telle que commerce électronique reste limitée, à cause de ses inconvénients déjà cités dans le chapitre II, mais la sécurité est un des problèmes les plus délicats posés par les agents mobiles, cette propriété est souvent citée comme la raison majeure qui en a limité la diffusion dont les causes sont :

- La mobilité des agents rend leur localisation et contrôle difficile.
- La difficulté de distinguer un agent d'un ver (Exemple : ver de Morris).

Le problème de la sécurité des agents mobiles s'énonce simplement par deux problèmes à résoudre :

- 1) La protection du système vis-à-vis des agents s'exécutant dans le système ;
- 2) La protection de l'agent vis-à-vis du système où il s'exécute.

Pour toutes ces raisons, des applications réparties à base d'agents mobiles, par exemple le e-commerce, peinent à voir le jour.

Dans le domaine de e-commerce, la sécurité devient un argument pour la promotion d'un site commercial, où elle est mise en avant pour inciter le client à effectuer ses achats en toute confiance. Mais souvent ce ne sont que des arguments, et les infrastructures

derrière sont loin d'être à la hauteur, soit par insouciance, soit par incompetence de l'administrateur. Parmi les problèmes de sécurité dans le e-commerce, le problème de paiement qui ralentisse d'une manière remarquable la diffusion de cette application.

IV. CONCLUSION

La technologie d'agent mobile naît de l'évolution de celui de système réparti, après le développement d'Internet et la généralisation de son utilisation, malgré les avantages qu'elle offre, la diffusion de cette technologie reste limitée

Un frein à l'utilisation réelle de cette technologie réside cependant dans les problèmes de sécurité qu'elle introduit. La mise en place d'une politique de sécurité peut demander, d'une part la protection des ressources et des données des machines hôtes, et d'autre part, la préservation de l'intégrité et de la confidentialité des agents eux-mêmes et de leurs communications.

Dans le chapitre suivant, nous allons proposer un modèle de sécurité qui répondra aux préoccupations que nous avons citées. Pour illustrer ce modèle, nous présenterons une application type e-commerce qui utilise ce modèle.

Chapitre IV

Proposition d'une solution de sécurité

INTRODUCTION

La sécurité déjà mentionnée dans le chapitre III, reste le problème le plus difficile à résoudre. C'est aussi la raison qui empêche la diffusion des agents mobiles. Comme tout système distribué, les agents mobiles sont sujets aux menaces du réseau comme : la corruption de données, le masquering, le déni de service, les écoutes, la répudiation... etc.

Les solutions à ces problèmes existent : chiffrement, autorisation, authentification, non-répudiation, etc. Mais ces solutions sont adaptées aux systèmes répartis classiques (par exemple : client-serveur).

Dans ce chapitre nous allons présenter un modèle d'interaction entre les agents, modèle autour duquel nous définirons une architecture optimale pour les agents mobiles dans le domaine de e-commerce.

I. DÉFINITION DE LA SÉCURITÉ

La sécurité consiste à empêcher les accès non autorisés aux informations [78]. Elle met en jeux deux types d'éléments :

- Les sujets (Utilisateurs, processus).
- Les objets (ressources, données).

Et elle se décompose en trois propriétés :

- **La confidentialité**

Dans la littérature, la confidentialité semble être la qualité la plus importante d'un système sûr. Assurer la confidentialité des données consiste à faire en sorte que les informations restent secrètes et que seules les personnes autorisées y aient accès. Les utilisateurs du système ont besoin d'avoir la garantie que leurs informations ne risquent pas d'être divulguées à des personnes non autorisées.

- **L'intégrité**

Outre la confidentialité des informations, l'intégrité évite la corruption, l'altération et la destruction des données dans le réseau de manière non autorisée.

- **La disponibilité**

Assurer aux utilisateurs légitimes une continuelle disponibilité des informations, des services et des ressources dans le réseau (les temps de réponse, la tolérance aux fautes, le contrôle de concurrence, le partage équitable de ressources, etc.). En somme, veiller à la disponibilité d'un système, c'est prévenir contre les perturbations et les interruptions de son fonctionnement et aussi contre toute utilisation abusive des services et des ressources du système.

La sécurité est souvent assimilée aux seules propriétés de confidentialité et d'intégrité des objets. Afin de garantir les propriétés de confidentialité et d'intégrité, il convient de protéger les objets, les ressources et les canaux de communication contre d'éventuelles attaques. Pour cela, des mécanismes d'authentification, de cryptographie (chiffrement) et de contrôle d'accès [81] (pare-feu) sont généralement utilisés.

I.1. Protection d'un site hôte envers un agent mobile

Un système d'exécution, s'il n'est pas protégé, devient vulnérable dès qu'il exécute du code venant de l'extérieur.

Lorsqu'un agent accède à la mémoire ou aux fichiers de la machine hôte, il peut facilement violer les propriétés de confidentialité ou d'intégrité.

Mais la protection des sites contre des attaques menées par des agents malveillants est un problème qui est aujourd'hui bien maîtrisé. En effet, plusieurs solutions permettent maintenant de se prémunir d'éventuelles attaques et voici les méthodes les plus connues :

I.1.1. Quelques solutions

1. Bac à Sable

La technique du bac à sable consiste à exécuter un agent à l'intérieur d'un environnement restreint. Ainsi, un site peut exécuter un agent douteux dans le bac à sable sans trop se soucier des problèmes de sécurité. Cette approche peut facilement se mettre en place en utilisant des interpréteurs de code dont leurs possibilités sont limitées.

2. Signature de code

La signature de code intervient lors de la création d'un agent, son créateur le signant numériquement afin qu'il puisse s'identifier durant ses déplacements. Cette technique permet d'obtenir une authentification de haut niveau pour les sites. Les applets Java adoptent désormais ce mode de fonctionnement. Ainsi si une applet est signée, elle est considérée comme un code de confiance et peut accéder à toutes les fonctionnalités de Java. Elle sera placée dans un bac à sable dans le cas contraire.

3. Contrôle d'accès

Pour améliorer les deux précédentes techniques, on met en place une politique de contrôle d'accès plus complexe. On peut la voir comme un raffinement d'une politique de bac à sable générale vers une politique spécifique à chaque application ou classe d'agents. En fonction des agents, le site pourra autoriser l'accès à un ensemble précis de fonctionnalités. Le contrôle d'accès permet de mixer les deux

premières techniques en offrant aux agents signés plus de fonctionnalité qu'un simple bac à sable sans pour autant accéder à toutes les fonctionnalités.

4. Vérification du code

La vérification de code permet d'obtenir une garantie sur la sémantique d'un code à travers l'analyse de sa structure, ou de son comportement pour un agent, en fonction d'une politique de sécurité donnée.

I.2. Protection d'un agent envers un site hôte

Les environnements d'exécution des agents sur les sites doivent avoir accès au code et à l'état des agents afin de pouvoir les exécuter. Durant l'exécution sur un serveur, le code, l'état et les données de l'agent sont donc exposés à des problèmes de confidentialité et d'intégrité.

Les attaques possibles ne sont pas encore toutes clairement identifiées[82], mais pour comprendre ce que risque un agent lors de son exécution sur un site malveillant, nous pouvons référencer les éléments transportés pouvant être cible d'attaque [75] :

Le Code : ensemble des instructions composant la tâche de l'agent.

Les données statiques : données ne changeant pas durant les déplacements.

Les données collectées : ensemble des résultats obtenus au cours des déplacements réalisés par l'agent depuis son lancement.

L'état courant : ensemble de données servant à l'exécution courante de l'agent.

Nous pouvons citer trois grandes catégories d'attaques que les sites sont capables de réaliser : l'inspection, la modification et le rejeu [76]. L'inspection consiste à examiner le contenu de l'agent afin de récupérer des informations critiques transportées par l'agent. La modification se réalise en remplaçant certains éléments de l'agent dans le but de conduire une attaque. En remplaçant le code, l'agent effectuera des opérations malveillantes sur les futurs sites à visiter. Le rejeu s'obtient en clonant l'agent puis en exécutant le clone dans plusieurs configurations pour retrouver le savoir de l'agent. Différentes techniques sont en cours d'étude pour garantir aux agents qu'ils peuvent accéder à des nœuds dans lesquels ils peuvent avoir toute confiance ou pour détecter les agents corrompus. Cependant, aucune d'elle n'apporte un niveau de sécurité suffisant comme celui proposé pour protéger les sites.

Ce que nous pouvons dire pour conclure sur les problèmes de sécurité liés à l'utilisation des agents mobiles, c'est qu'il s'agit encore d'un champ d'investigation complet. Bien que la protection des sites soit à présent maîtrisée, la protection des agents n'est, pour l'instant, toujours pas satisfaisante.

Les recherches actuelles n'offrent pas de méthode générale pour protéger les agents. Les développeurs d'application à agents mobiles doivent concevoir leurs propres mécanismes de protection selon leurs besoins.

I.2.1. Quelques solutions

1. Chiffrer les données d'agents afin de les protéger sur un itinéraire, Young et Yung [83] ont proposé un moyen de chiffrer les données accumulées et non réutilisées sur l'itinéraire d'un agent.
2. L'utilisation de signatures permet de prouver qu'un site a été néfaste envers un agent. Ces solutions portent principalement sur l'intégrité et consistent à prouver, a posteriori qu'il y a eu falsification de l'agent.
 - Vigna [84] suggère l'utilisation d'une trace d'exécution des agents afin de détecter, a posteriori, les falsifications de l'exécution.
 - L'authentification partielle des résultats [40] permet de détecter, a posteriori, la falsification sur les données.
3. Pour assurer la confidentialité des parties de codes et des données de l'agent, il existe des solutions partielles qui chiffrent le code et les données tout en les gardant exécutables.
 - Le brouillage de code [34] cherche à protéger l'agent contre une analyse de l'attaquant.
 - Sander et Tschusdin [85] ont proposé une idée théorique afin de préserver la confidentialité de parties de code d'un agent. L'idée est de chiffrer certaines fonctions de l'agent et les résultats ne peuvent être déchiffrés que sur le site client.

II. UN MODÈLE D'INTERACTION EFFICACE ENTRE LES AGENTS MOBILES : LE MODÈLE ACHETEUR / VENDEUR

II.1. Présentation du modèle Acheteur / Vendeur

Suite aux développements d'Internet, les entreprises s'aperçoivent de l'intérêt du commerce électronique pour la promotion de leurs produits. Par exemple : Un utilisateur n'accède pas directement aux différents magasins, mais délègue la tâche de rechercher le meilleur prix d'un produit à un agent « acheteur ». L'agent acheteur tente de négocier avec les différents « Vendeurs ». Ce processus de négociation devrait converger vers des solutions optimisant les gains des différents intervenants. Selon l'exemple, nous aurons affaire à deux types d'agents mobiles :

1. Les agents qui représentent les clients ou *agents acheteurs* de services.
2. Les agents qui représentent les services ou *agents vendeurs* de services.

Ajoutons les sites de rencontre (ou rendez-vous) des agents mobiles où ils pourront interagir localement.

II.1.1. Principe de fonctionnement du modèle Acheteur / Vendeur

Les agents interagissent localement selon des schémas de coopération, de négociation et même de compétition et forment un système multi-agents.

Les agents vendeurs ne sont pas obligés de migrer systématiquement. En particulier, lorsqu'une transaction est conclue entre un agent acheteur et un agent vendeur, l'agent vendeur seul interagira avec son site d'origine pour livrer le service demandé.

De plus ce modèle n'exclut pas le recours aux interactions plus classique telles que l'interaction distante ou la migration directe d'un agent sur un hôte.

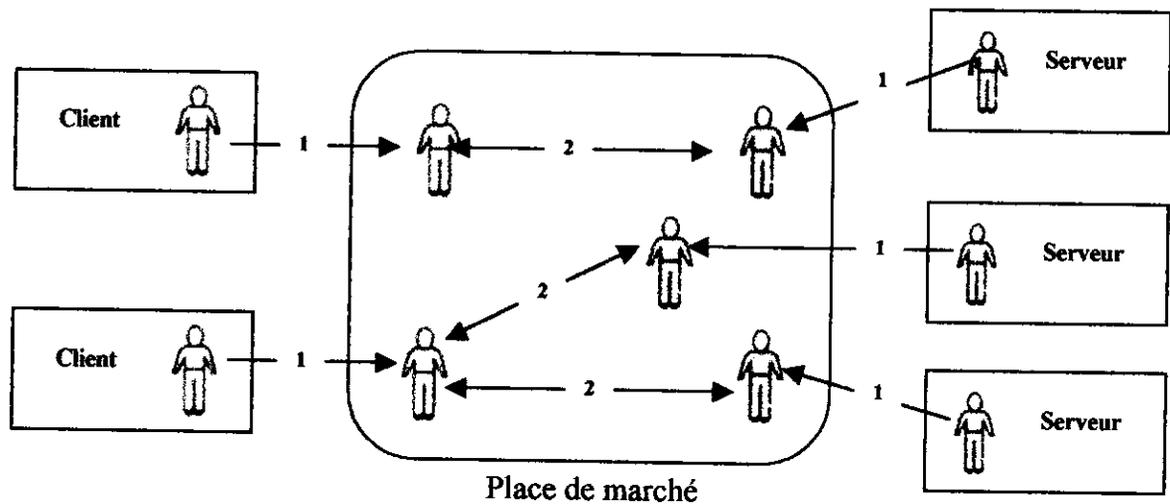
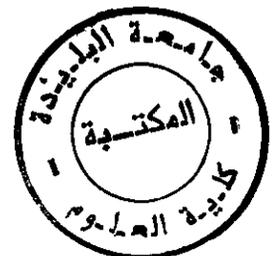


Figure IV.1 : Interaction des agents mobiles à travers une place de Marché. [77]

Les différentes notations utilisées dans la figure précédente sont indiquées dans le tableau ci-dessous :

	Agent acheteur	1	Migration
	Agent vendeur	2	Interaction locale



Il existe plusieurs stratégies d'interaction dans le modèle Acheteur/Vendeur :

- Interaction locale (IL).
- Interaction distante (ID).
- Migration directe ou complète (MD).
- Migration indirecte ou partielle (MI).

II.1.2. Application du modèle Acheteur / Vendeur

Il existe trois implémentations possibles du modèle acheteur-vendeur ou « SB » (Seller-Buyer) :

1. « SB0 » : Les deux composants (client et serveur) sont statiques.
2. « SB2 » : L'acheteur est mobile (c'est un agent) mais le vendeur (serveur) est statique.
3. « SB3 » : L'acheteur et le vendeur sont mobiles.

Dans la première implémentation, les solutions de sécurité sont éprouvées parce que dans les systèmes distribués client-serveur, les problèmes de sécurité ont des solutions partielles largement employées, par contre dans la seconde implémentation se pose le problème de sécurité de l'agent mobile (Acheteur). En ce qui concerne la dernière implémentation, le problème de sécurité est totalement réglé grâce à la mise en œuvre des lieux de rencontre (places de marché) hautement sécurisés dont l'accès est strictement contrôlé.

II.1.2.1.. Schéma « SB »

Voici la présentation schématique des trois implémentations décrites précédemment :

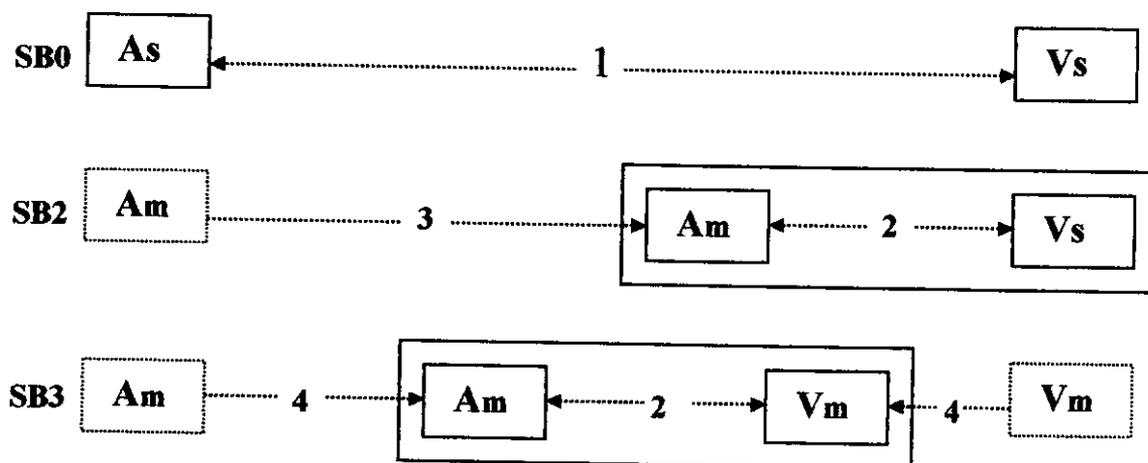


Figure IV.2 : Différentes implémentations du schéma SB. [77]

Les différentes notations utilisées dans le schéma précédent sont indiquées dans le tableau ci-dessous.

1	Interaction distante	As	Agent acheteur statique
2	Interaction locale	Am	Agent acheteur mobile
3	Migration directe	Vs	Agent vendeur statique
4	Migration indirecte	Vm	Agent vendeur mobile

III. ARCHITECTURE PROPOSÉE

Dans cette section nous allons décrire une architecture pour la sécurité des agents mobiles.

III.1. Vue générale

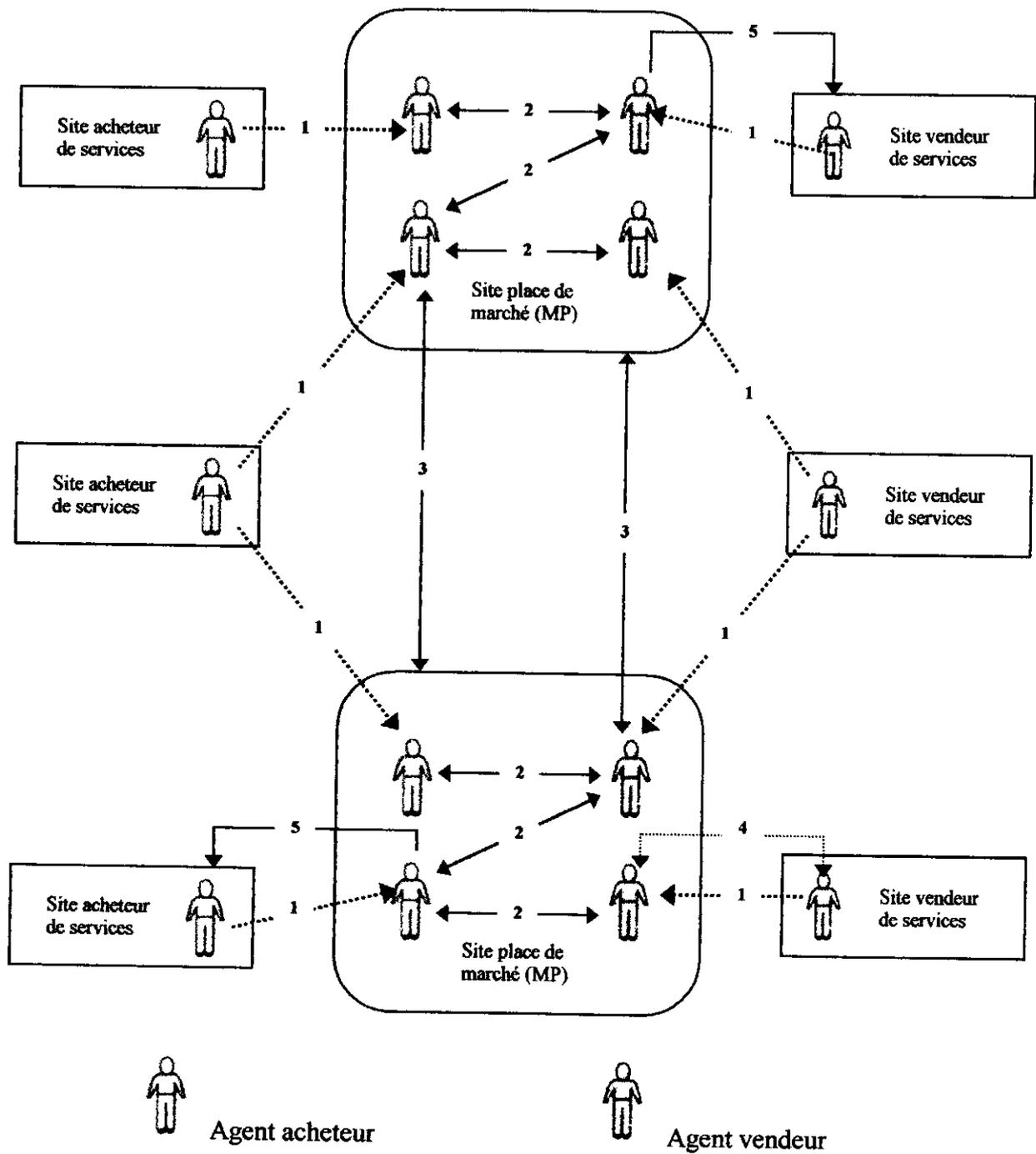
L'architecture est décrite autour du modèle acheteur/vendeur, elle est basée sur le schéma « SB3 », et permet d'assurer la sécurité des agents mobiles en limitant leurs déplacements à des sites de rencontre (places de marché) hautement sécurisé.

III.2. Description de l'architecture proposée

L'architecture comporte :

- 3 Types de sites
 1. Les sites acheteurs de services.
 2. Les sites vendeurs de services.
 3. Les sites places de marché ou site MP.

- 3 Types d'agent
 1. Les agents mobiles acheteurs (BMA).
 2. Les agents mobiles vendeurs (SMA).
 3. Les agents statiques facilitateurs (SFA), qui stimulent la rencontre, sur un site MP, entre un agent acheteur et un agent vendeur, Cet agent tient le rôle de gestionnaire du site MP.



- 1 Migration indirecte(ou partielle)
- 2 Interaction locale
- 3 Migration vers un autre site MP
- 4 Interaction distante
- 5 Migration retour vers le site d'origine

Figure IV.3 : Schéma général de l'architecture proposée. [77]

III.3. Scénario de fonctionnement de l'architecture proposée

a) Sur le site Acheteur :

Le client exploite l'interface utilisateur pour exprimer ses besoins, et pour les réaliser, il délègue un agent acheteur qui se chargera de l'achat à sa place. L'agent acheteur migre vers le site MP le mieux adapté, et après authentification, il expose ses besoins à l'agent facilitateur, ce dernier consulte sa base de connaissances et met l'agent acheteur en contact avec un agent vendeur intéressant ou lui recommande de migrer vers un autre site MP.

L'agent acheteur négocie avec l'agent vendeur, si la négociation aboutit, une transaction s'effectue, sinon il contacte un autre agent vendeur présent sur le même site MP ou il migre vers un autre site MP.

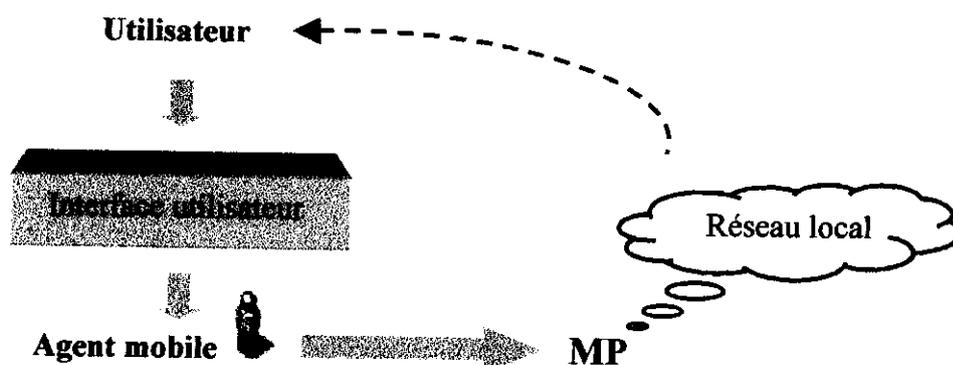


Figure IV.4 : Conception coté client.

b) Sur le site vendeur:

Le site vendeur délègue un agent mobile pour promouvoir la vente de ses produits, l'agent vendeur migre vers un site MP, expose ses services (après authentification) à l'agent facilitateur, et sur recommandation de ce dernier, il interagit avec des agents acheteurs intéressés par ses services.

IV. LA SÉCURITÉ DU SYSTÈME

Nous présentons ci-dessous le modèle de sécurité basé sur l'architecture proposée.

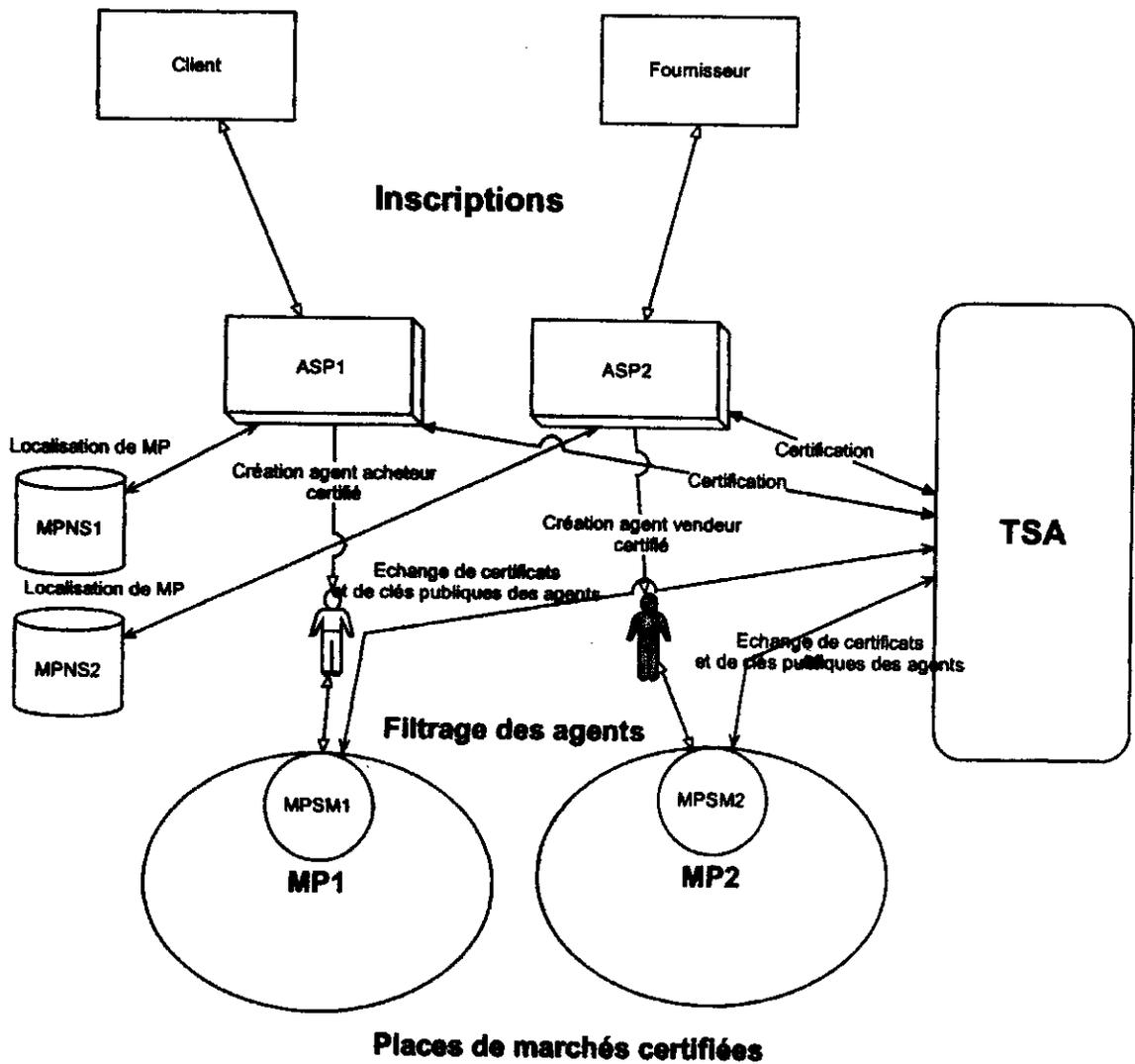


Figure V.5 : Schéma de la sécurité.



IV.1. Description détaillée de l'architecture proposée:

Notre architecture est basée sur les composants suivants :

- Les acheteurs de services(CBS ou Client Buyer of Services) ;
- Les vendeurs ou fournisseurs de services(PSS ou Provider Seller of Services) ;
- **MP** (MarketPlaces ou places de marché) : Des lieux de transactions qui englobent les e-shops qui offrent des produits (ou services) à vendre (par les agents vendeurs) pour les clients potentiels(les agents acheteurs). [87]

Pour que ces composants puissent interagir, nous définissons les composants suivants :

- **ASP** (Agent Service Provider ou des fournisseurs de service agent) Ils offrent aux clients (acheteurs ou vendeurs) un environnement de création et de gestion d'agents logiciels. [87]
- **MPNS** (Market place Naming service ou des services de nommage) Ils offrent des informations sur les market places sur le réseau Internet. [87]
- **TSA** (Trust & Authority Service) Qui gère la sécurité de migration des agents ainsi que la sécurité des places de marché visitées. [87,19]

Pour la sécurité, nous utiliserons la technologie PKI [PKI] (public Key Infrastructure) qui fournit des clés de cryptage publiques et privées, largement utilisées dans le réseau Internet. Le cryptage étant basé sur l'algorithme RSA.

Pour le paiement nous intégrerons une autorité bancaire virtuelle (VBA) qui génère un VAN (virtual Account Number) pour chaque utilisateur inscrit sur l'ASP.

IV.1.1. Architecture d'une place de marché

Une place de marché est un Intranet, c'est à dire un réseau local de plusieurs hôtes. Certains hôtes jouent un rôle particulier, nous trouvons :

- **MPSM** (MP Security Manager) : Il gère la sécurité de la place. Il agit aussi comme un Pare-feu.
- Un agent facilitateur : Il gère la place de marché.
- Les e-shops : Ils sont des hôtes hébergeant un service agent (ou agent facilitateur) qui accueille les agents vendeurs de produits et les agents acheteurs correspondants.

IV.1.2. Bases de connaissances :

Notre architecture définit plusieurs bases de connaissances qui facilitent la localisation des services et ressources nécessaires pour les agents mobiles. Nous distinguons :

- La base annuaire MPDS (MP Directory Service) : stockée sur chaque serveur MP, cette base contient :

- 1) Les adresses IP des e-shops de la MP.
 - 2) Le catalogue de domaine (DC) qui contient les informations sur les services et ressources disponibles sur les e-shops.
- La base e-shop ou catalogue local (LC) : Il contient toutes les informations détaillées sur tous les services et ressources disponibles au niveau de l'e-shop, il est limité à un seul e-shop.

IV.1.3. L'autorité bancaire virtuelle (VBA)

Chaque utilisateur inscrit à l'ASP possède un compte bancaire virtuel (VAN) dans la banque VBA.



Figure IV.6 : Création d'un compte bancaire virtuel.

V. CONCLUSION

Dans ce chapitre, nous avons montré que dans un environnement avec infrastructure de rencontre, la sécurité devrait être accrue. En effet, tous les échanges entre les fournisseurs et consommateurs (ou leurs agents respectifs) de services se feraient localement dans un endroit « sûr ». Pour cette fin, les fournisseurs et consommateurs seraient contrôlés avant d'être autorisés à entrer. De plus, les consommateurs et fournisseurs pourraient avoir un visa d'accès, qui devrait contenir diverses informations comme leurs identités, la date d'émission et d'expiration de leurs visas, etc. Nous avons ainsi montré grâce à l'architecture proposée que les exigences de sécurité des agents mobiles sont maintenues.

Dans le chapitre suivant, nous présentons notre système CESAM et nous décrivons en détail les aspects de sécurité dans l'architecture proposée à l'aide du langage UML.

Chapitre V

Conception du système CESAM

INTRODUCTION

Dans ce chapitre nous avons choisi d'utiliser le langage UML pour la description de notre système CESAM⁽¹⁾ qui permet de faire des transactions commerciales entre un acheteur et un vendeur par l'intermédiaire d'agents mobiles en toute sécurité.

I. DESCRIPTION D'UN MODÈLE D'APPLICATION DE E-COMMERCE

Nous allons décrire un modèle d'application simple pour servir d'exemple d'application répartie pour le système CESAM. Cette application est de type e-commerce.

L'application de commerce électronique assure deux objectifs :

- Achat de produit.
- Vente de produit.

L'application contient :

- Un site utilisateur, à partir de ce site l'utilisateur lance les différentes requêtes : achat si l'utilisateur est un acheteur ou vente si l'utilisateur est un vendeur.
- Un site de commerce électronique où l'on trouve les produits à acheter ou services à offrir pour les clients potentiels.

Dans les sites de commerce électronique, les produits disponibles sont stockés dans des catalogues électroniques.

Le paiement est souvent assuré par un autre tiers (une banque).

Pour mieux comprendre le fonctionnement de l'application, on va décomposer les fonctions principales en sous fonctions.

⁽¹⁾ : Commerce électronique sécurisé à base d'agents mobiles.

I.1. Les cas d'utilisation

Les besoins peuvent être exprimés sous forme de cas d'utilisation, dans un langage très proche des utilisateurs.

1. Les acteurs

Un acteur représente un rôle joué par une personne ou un système qui interagit avec le système en question, les acteurs se déterminent en observant les utilisateurs directs du système. [86]

Les acteurs se répartissent dans les catégories suivantes :

Acteur principal :

- L'utilisateur : toute personne qui manipule le système.

Autres systèmes :

- La banque.

2. Détermination des cas d'utilisation

L'application présente les cas d'utilisation suivants :

- Achat d'un produit.
- Vente d'un produit.

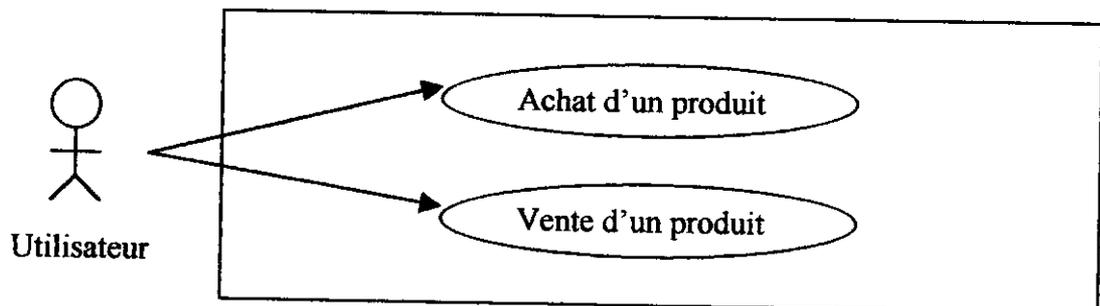


Figure V.1 : Diagramme de cas d'utilisation général de l'application.

2.1. Achat d'un produit

L'achat d'un produit se déroule comme suit :

- ✓ L'utilisateur saisit les informations nécessaires à la recherche d'un produit ;
- ✓ Les informations à saisir sont : le nom de produit, la marque de produit, prix Min et prix Max ;
- ✓ Un message doit être affiché dont le contenu est le nom et le prix de chaque produit trouvé avec les vendeurs correspondants;
- ✓ L'utilisateur choisit un produit parmi les produits disponibles;
- ✓ L'utilisateur doit entrer ses informations bancaires pour effectuer la transaction ;
- ✓ Un message doit être affiché pour indiquer si l'achat a été effectué ou non.

Dans ce qui suit, nous allons représenter tout les diagrammes de cas d'utilisation possible pour l'achat, puis un diagramme de cas d'utilisation général et détaillé.

L'achat d'un produit comprend les fonctions principales suivantes :

- Recherche d'un produit.
- Validation d'achat.

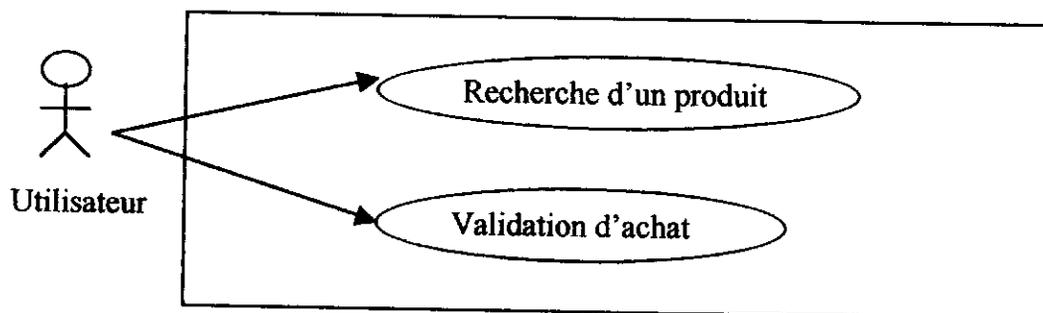


Figure V.2 : Diagramme de cas d'utilisation général de l'achat.

2.1.1. Recherche d'un produit

La recherche d'un produit est une fonction qui permet de trouver tous les produits disponibles dans le catalogue électronique avec les caractéristiques demandées, cette fonction inclus une autre fonction :

- Vérification de la disponibilité de produit

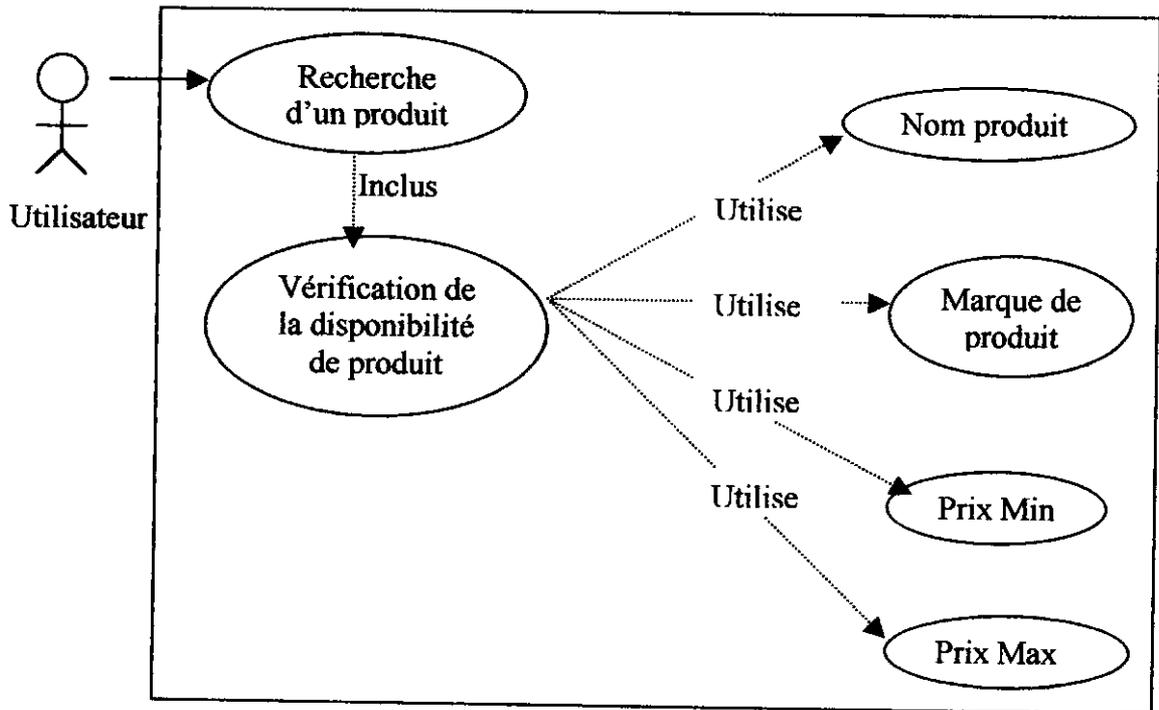


Figure V.3 : Diagramme de cas d'utilisation de la recherche d'un produit.

2.1.2. Validation d'achat

On peut décomposer cette fonction en deux autres fonctions :

- Effectuer achat.
- Paiement.

La fonction **Effectuer achat** inclus elle-même une autre fonction qui est le **choix de produit**, comme le montre le diagramme suivant :

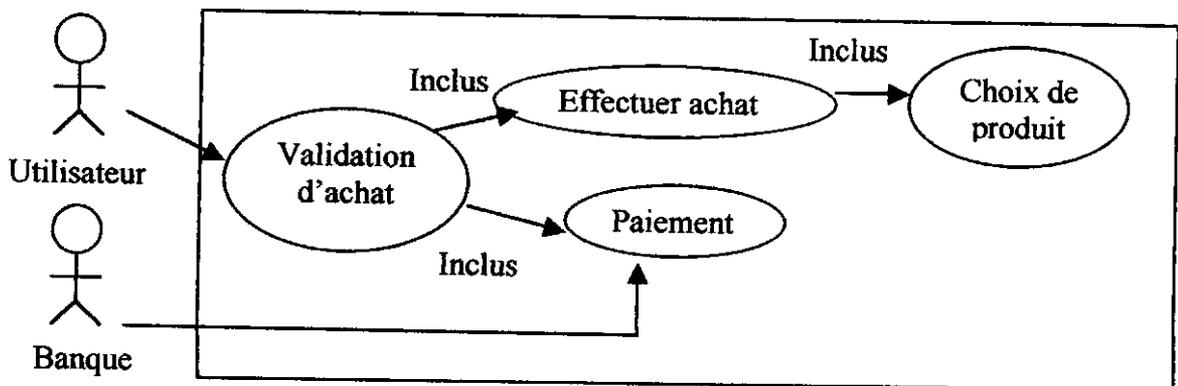


Figure V.4 : Diagramme de cas d'utilisation de la validation d'achat.

On va maintenant regrouper les deux diagrammes précédents dans un seul diagramme détaillé.

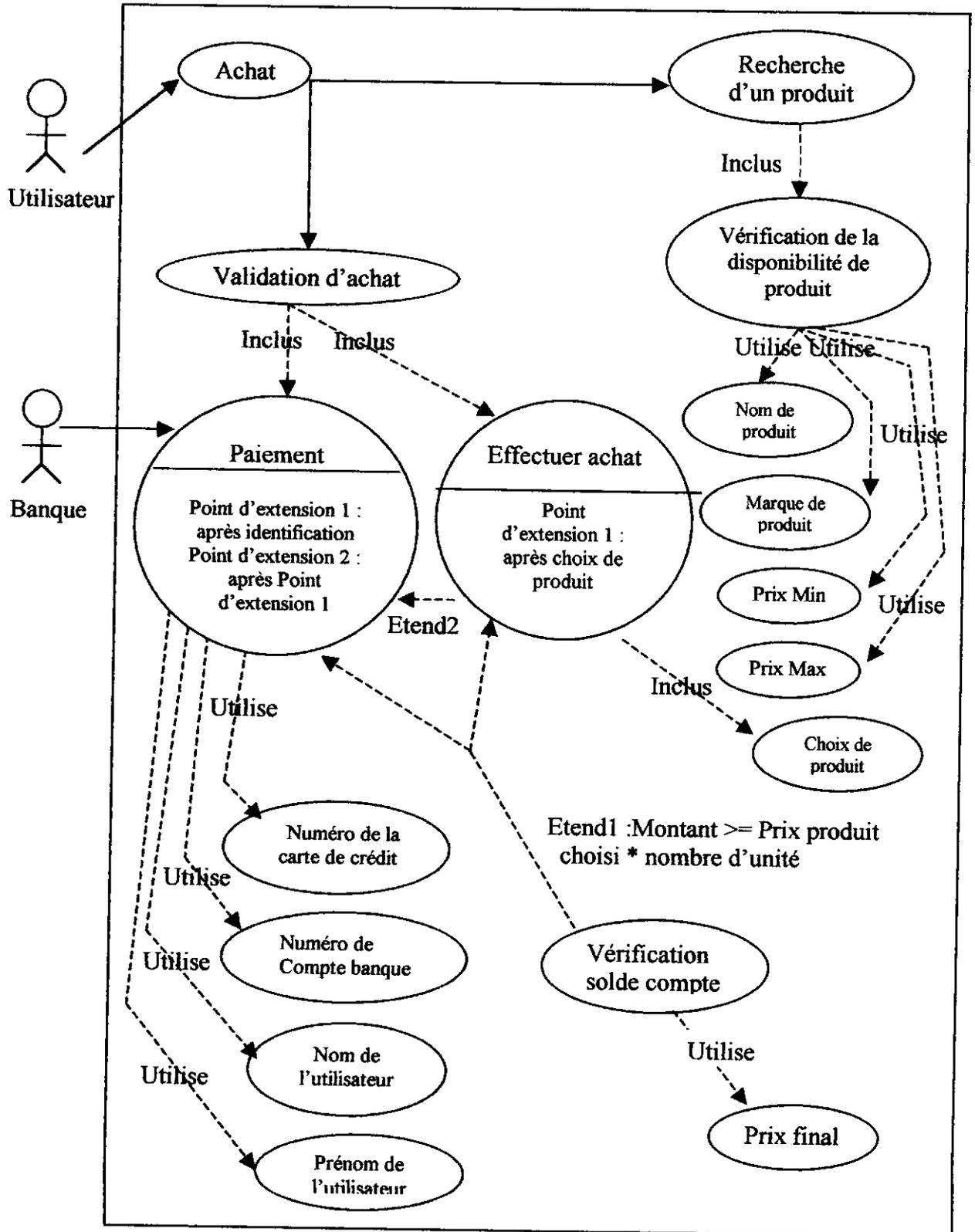


Figure V.5 : Diagramme de cas d'utilisation de l'achat d'un produit.

2.2. Vente d'un produit

La vente d'un produit est assurée par un fournisseur ou un vendeur des produits. Cette fonction englobe une autre fonction:

- Proposition d'un service ;

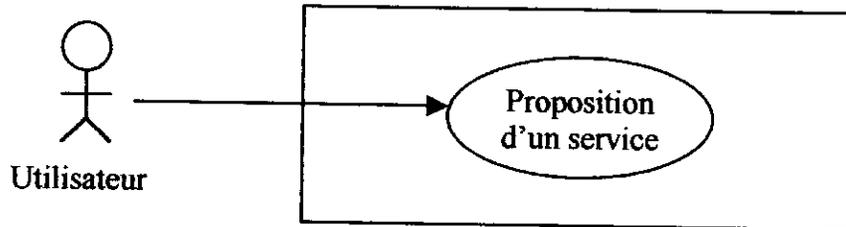


Figure V.6 : Diagramme de cas d'utilisation général de la vente

Dans ce qui suit, nous allons présenter le diagramme de cas d'utilisation général de la vente

2.2.1. Proposition d'un service

La proposition d'un service se déroule comme suit :

- ✓ L'utilisateur saisit les informations nécessaires à la vente du produit proposé;
- ✓ Les informations à saisir sont : l'identificateur de vendeur, le nom de produit, marque de produit, le nombre d'unité à vendre, prix Min et prix Max;
- ✓ Un message doit être affiché dont le contenu est : produit ajouté au catalogue électronique ;

On peut représenter le diagramme de cas d'utilisation de la vente comme suit :

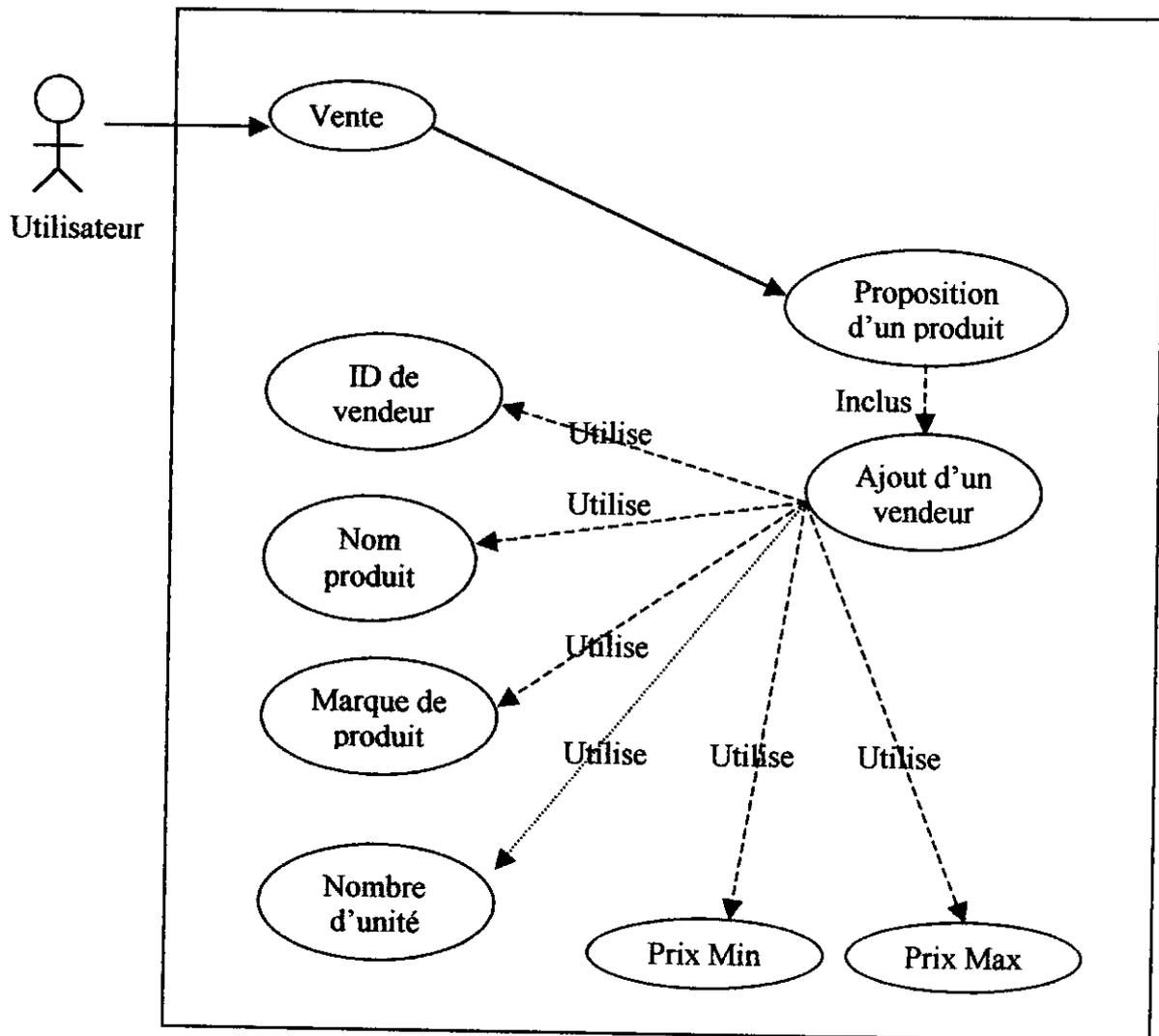


Figure V.7: Diagramme de cas d'utilisation de la vente d'un produit.

2.3. Diagramme de cas d'utilisation de l'application

Les différentes fonctionnalités offertes par notre outil forment ainsi un ensemble de cas d'utilisation (use case), exprimés dans les sections précédentes, afin de les formaliser, UML propose à travers des diagrammes de cas d'utilisation de répertoire et de structurer les fonctionnalités que le futur système offrira à ses utilisateurs.

La figure V.8 illustre le diagramme de cas d'utilisation de l'application.

II. DESCRIPTION DU SYSTÈME CESAM À TRAVERS LE MODÈLE D'APPLICATION PROPOSÉ

Dans cette partie, on va décrire formellement le système CESAM qui est basé sur l'architecture proposée dans le chapitre précédent, on va montrer comment les différents composants de l'architecture s'interrogent entre eux afin de réaliser le but voulu.

II.1. Achat d'un produit

Pour mieux comprendre le fonctionnement du système, on va décomposer la fonction d'achat en sous fonctions comme le montre le schéma suivant :

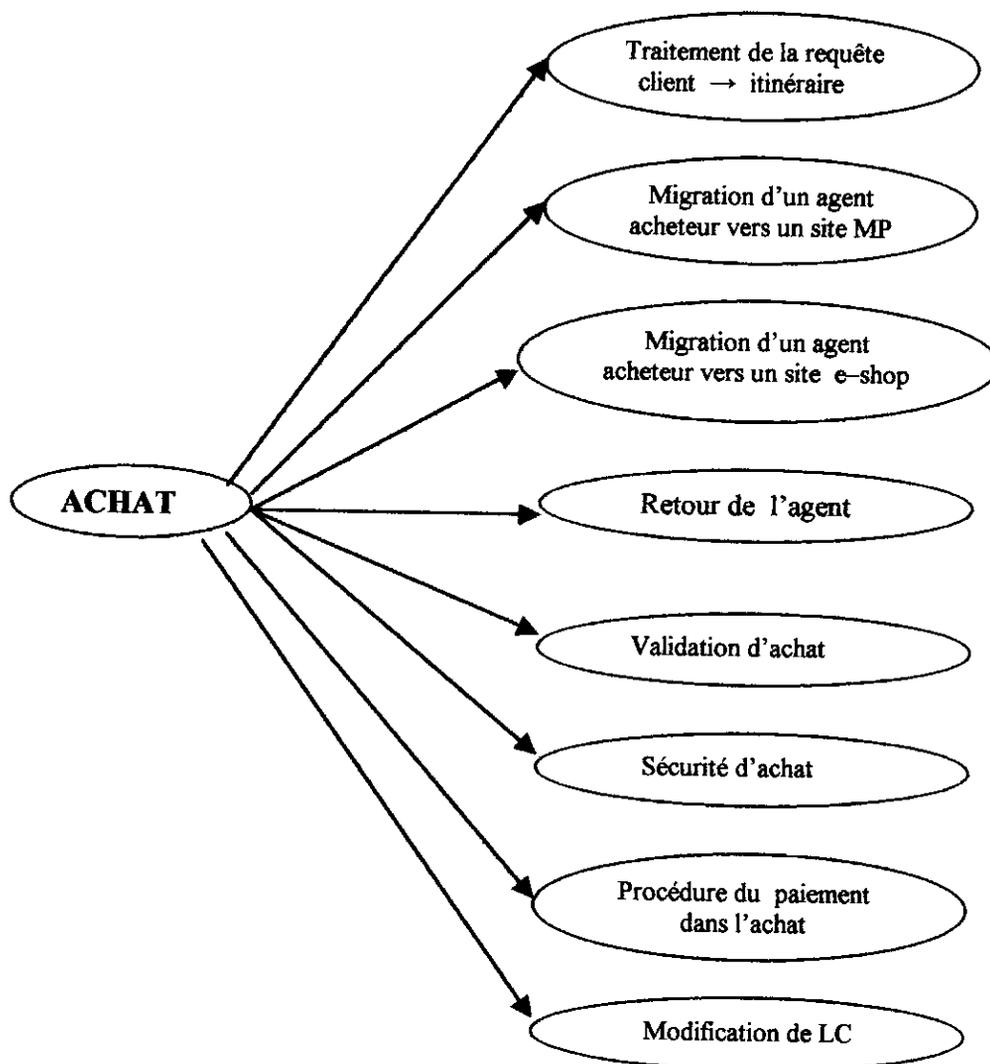


Figure V.9: Décomposition de la fonction d'achat.

Dans ce qui suit, on va détailler les sous fonctions montrées dans le schéma ci-dessus.

II.1.1. Diagramme de séquence

Scénario 1 : Traitement de la requête client jusqu'à l'affectation d'itinéraire

- Le client se connecte à l'ASP ;
- L'ASP envoie une fenêtre pour que le client puisse s'inscrire ;
- Le client spécifie ses besoins en remplissant le formulaire fourni par la fenêtre;
- Le client lance sa requête ;
- L'ASP crée un agent acheteur (BMA) pour exécuter la requête du client sur le réseau ;
- L'ASP demande au serveur MPNS d'envoyer un itinéraire initial ;
- L'MPNS propose un itinéraire à l'ASP ;
- L'ASP affecte l'itinéraire à l'agent acheteur déjà créé.

La figure V.10 illustre le diagramme de séquence du traitement de la requête client jusqu'à l'affectation d'itinéraire.

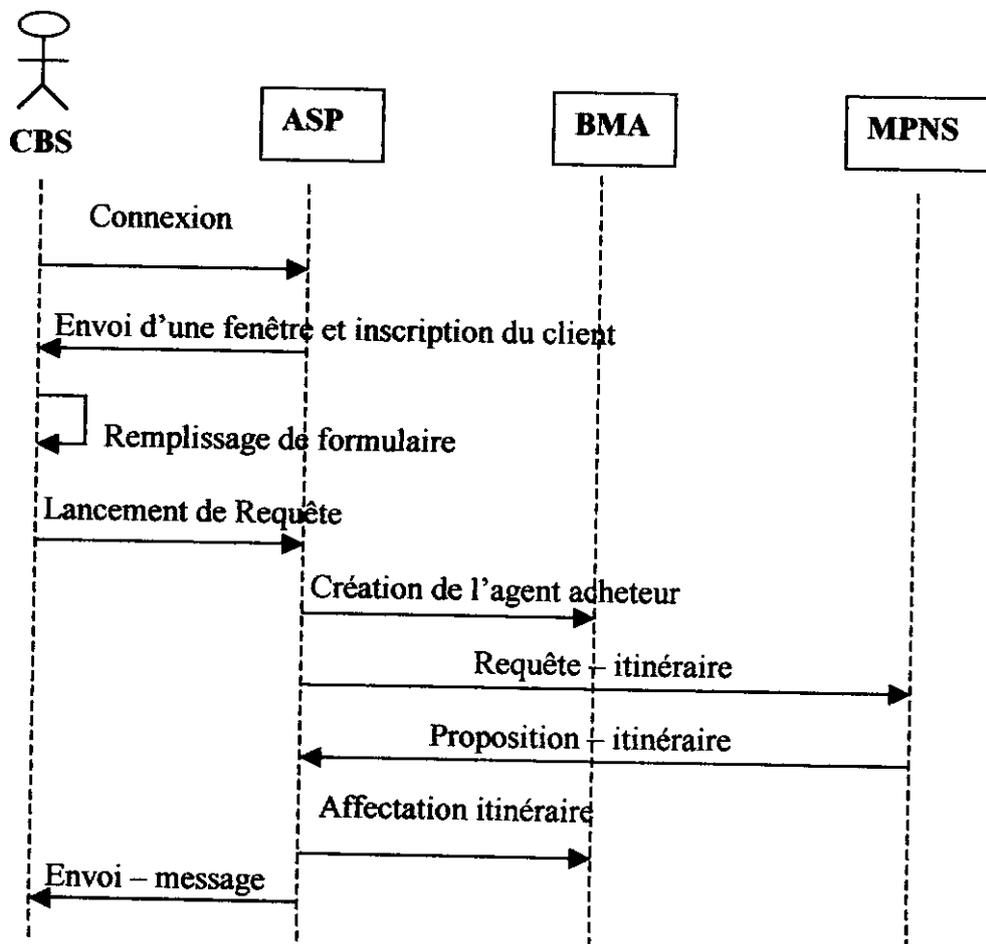


Figure V.10: Diagramme de séquence du traitement de la requête client.

Scénario 2 : Migration d'un agent acheteur vers un site MP

- L'agent acheteur BMA demande une autorisation de migration à l'agent facilitateur distant (MP-IFA) du premier site MP de l'itinéraire ;
- L'agent MP-IFA approuve la demande ;
- L'agent acheteur BMA notifie sa migration à l'agent facilitateur local de l'ASP(ASP-IFA) ;
- Au niveau du premier site MP, l'agent acheteur demande d'être authentifié par le MPSM (par- feu) de ce site ;
- MPSM demande un jeton au MPDS du premier site ;
- MPDS envoie un jeton au MPSM ;
- MPSM affecte le jeton à l'agent acheteur. L'agent est donc authentifié et contrôlé par le MPSM ;
- L'agent acheteur authentifié demande à l'agent facilitateur de site MP (MP-IFA) de le router vers un e-shop ;
- MP-IFA consulte un annuaire (MPDS) pour vérifier la disponibilité des ressources demandées ;
- Après la vérification, si la ressource a été trouvée, l'agent MP-IFA route l'agent acheteur (une sorte de sous itinéraire), sinon l'agent acheteur migre vers un autre site MP;
- Le processus est répété sur tous les sites MP présents dans l'itinéraire tant que les ressources ne sont pas disponibles dans le présent site.

La figure V.11 représente le diagramme de séquence de la migration d'un agent acheteur vers un site MP :

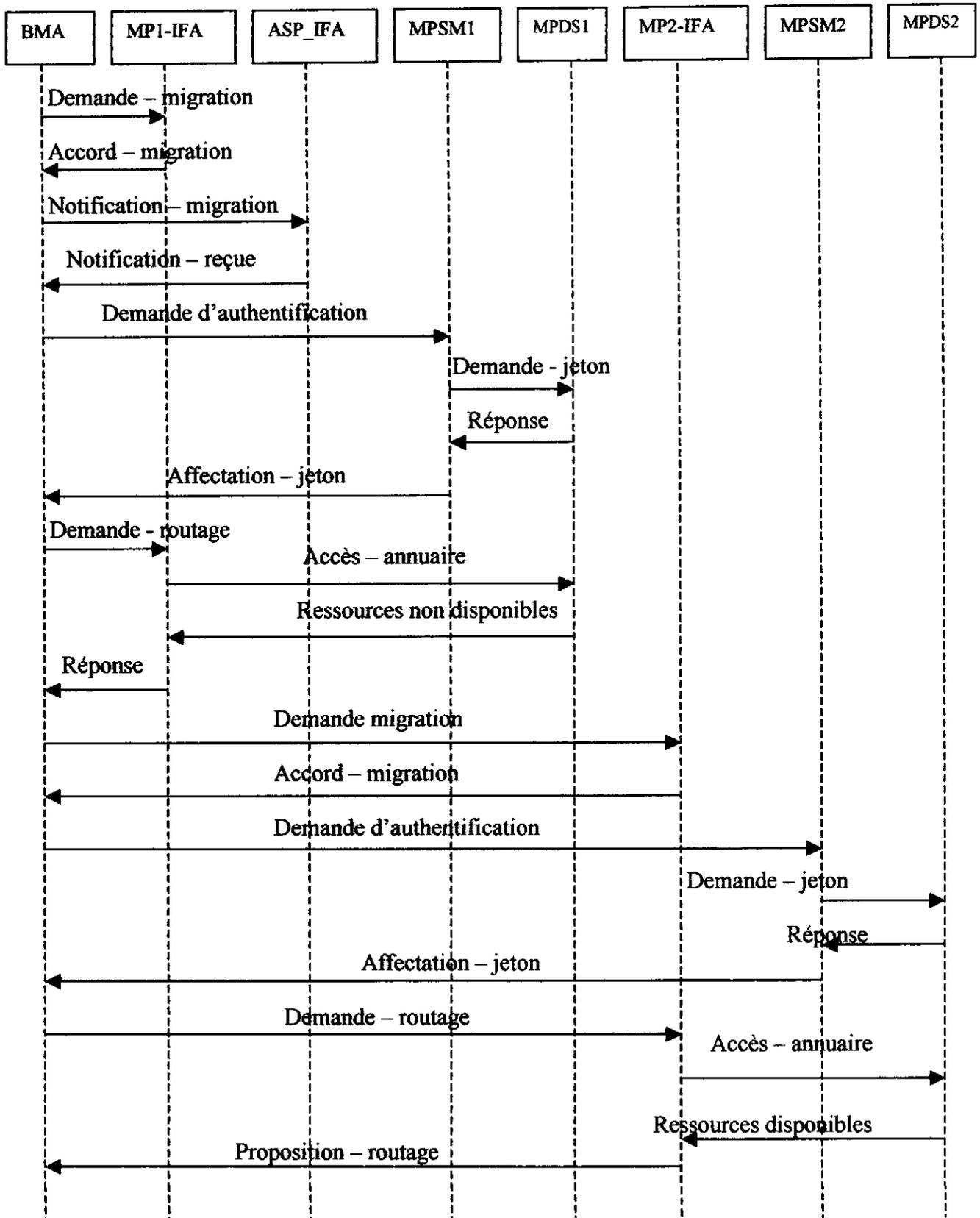


Figure V.11: Diagramme de séquence de la migration d'un agent acheteur vers un site MP.

Scénario 3 : Migration d'un agent acheteur vers un site e-shop

- L'agent acheteur BMA demande une autorisation de migration à l'agent facilitateur local e-shop-IFA du premier e-shop de sous itinéraire ;
- L'agent e-shop-IFA demande à l'agent acheteur d'envoyer son jeton ;
- L'agent e-shop-IFA approuve la demande après la vérification de jeton ;
- L'agent acheteur demande à l'agent facilitateur local de e-shop de le mettre en contact avec un ou plusieurs agents vendeurs ;
- L'agent facilitateur de e-shop consulte le catalogue local (LC), si un vendeur est présent sur le e-shop, il met l'acheteur en contact avec le vendeur trouvé, sinon l'agent acheteur migre vers un autre e-shop;
- Le processus est répété sur tous les e-shops présents dans le site MP tant qu'il n'existe pas un agent vendeur sur le présent e-shop.

La figure V.12 illustre le diagramme de séquence de la migration d'un agent acheteur vers un site e-shop :

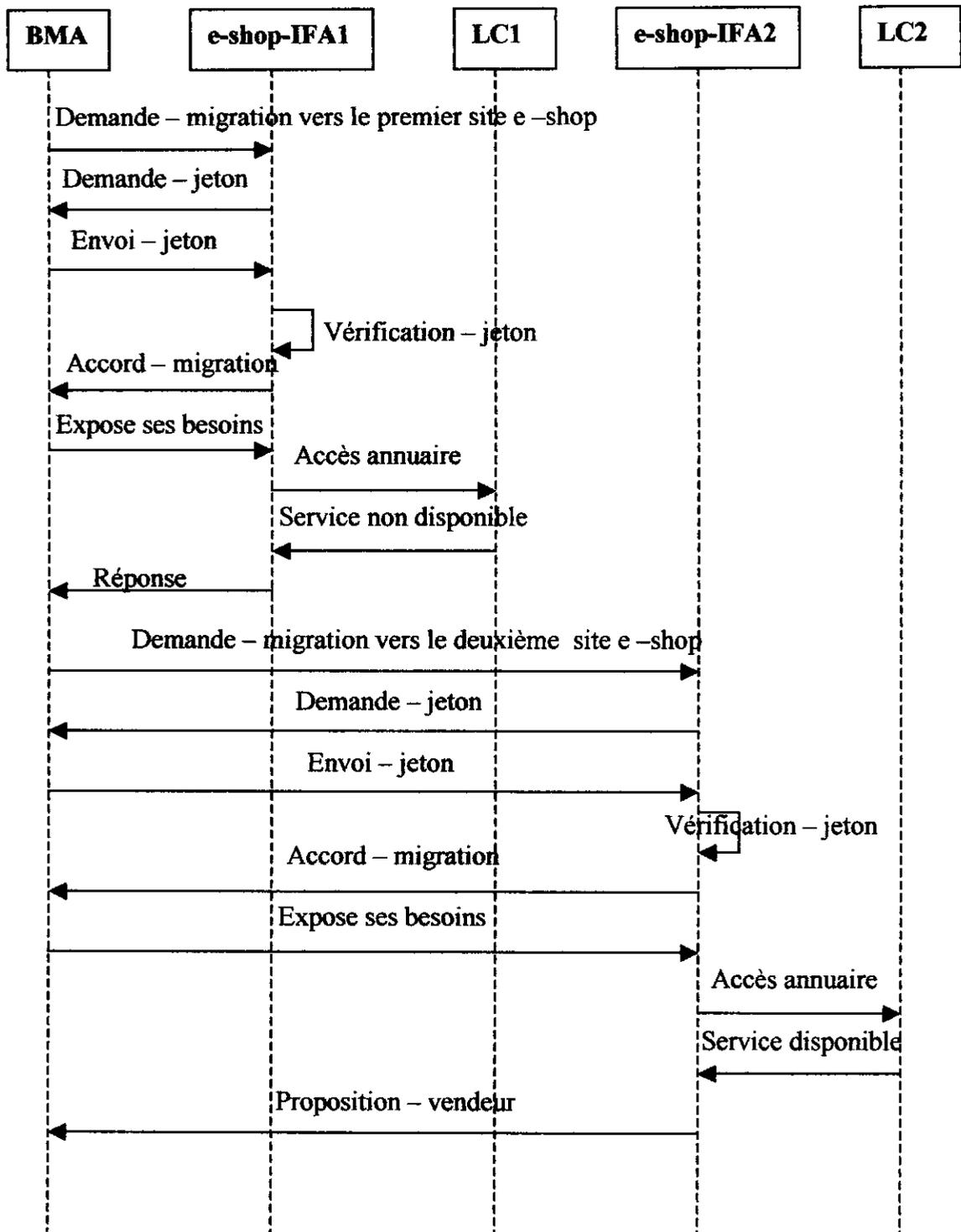


Figure V.12 : Diagramme de séquence de la migration d'un agent acheteur vers un site e-shop.

Scénario 4 : Retour de l'agent acheteur

- L'agent acheteur BMA demande une autorisation de migration à l'agent facilitateur du site MP ;
- L'agent MP-IFA approuve la demande ;
- L'agent acheteur migre vers le site MP et demande à l'agent facilitateur de l'ASP « ASP-IFA » l'autorisation de migration ;
- L'agent ASP-IFA approuve la demande ;
- L'agent acheteur retourne à l'ASP avec les résultats obtenus ;
- L'ASP transmet les résultats finaux à l'utilisateur ;
- L'ASP désactive l'agent acheteur.

La figure V.13 représente le diagramme de séquence de retour de l'agent acheteur :

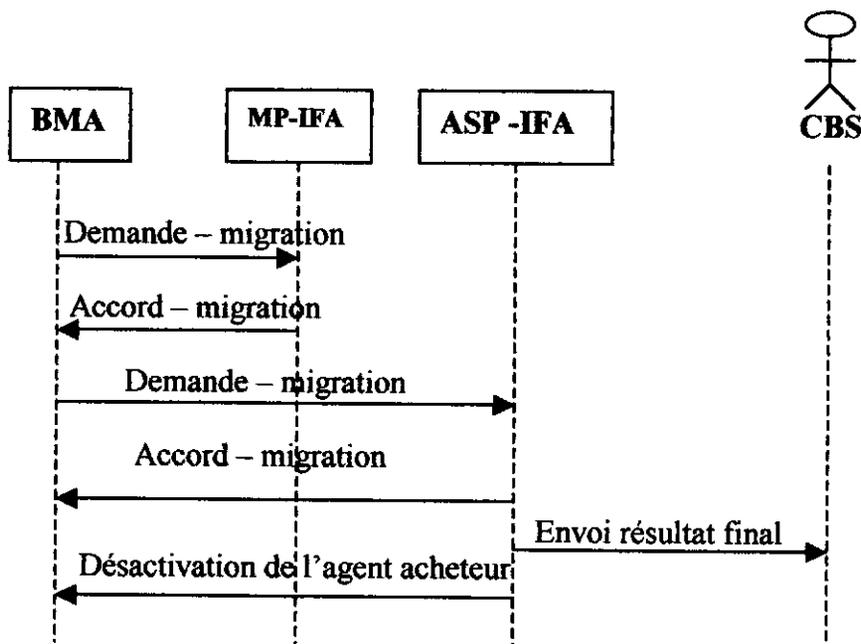


Figure V.13 : Diagramme de séquence de retour de l'agent acheteur.

Scénario 5 : Validation d'achat

- Le client se connecte à l'ASP ;
- L'ASP envoie une fenêtre;
- Le client demande la validation d'achat du produit choisi ;
- L'ASP lui demande de spécifier le nombre d'unité à acheter ;
- Le client spécifie sa demande;
- L'ASP vérifie la disponibilité de la quantité demandée chez le vendeur ;
- Le client effectue l'achat ;

La figure V.14 illustre le diagramme de séquence de la validation d'achat :

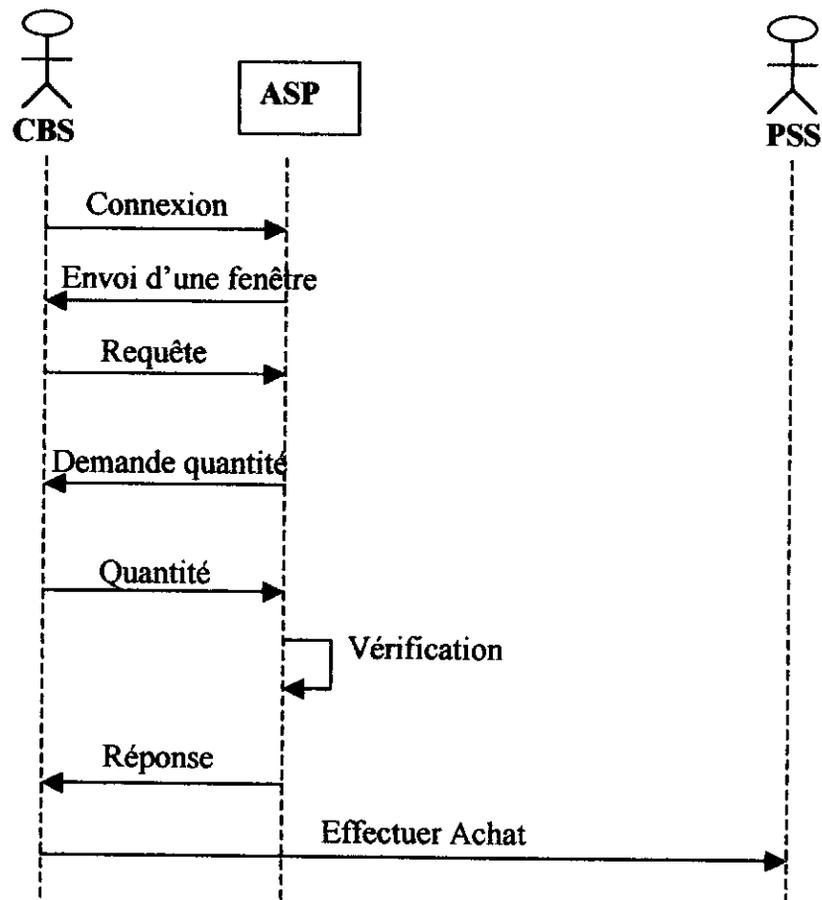


Figure V.14: Diagramme de séquence de la validation d'achat.

Scénario 6 : Sécurité d'achat

- L'ASP-IFA demande au TSA d'envoyer un certificat pour l'agent acheteur ;
- TSA transmet un certificat à l'ASP ;
- TSA consigne le certificat dans sa base de connaissance ;
- L'ASP-IFA affecte le certificat à l'agent acheteur déjà créé ;
- L'ASP-IFA signe l'agent acheteur par la clé privée de ce dernier ;
- MP-IFA vérifie si le certificat est valide auprès de la TSA ;
- TSA envoie la réponse ;
- MP-IFA demande la clé publique de l'agent ;
- TSA envoie la clé publique ;
- MP-IFA décrypte les signatures de l'agent par la clé publique de ce dernier ;
- Le e-shop-IFA crypte les résultats obtenus par la clé publique de l'agent ;
- L'ASP-IFA décrypte les résultats cryptés par la clé privée de l'agent acheteur ;
- L'ASP-IFA notifie au TSA la destruction de l'agent acheteur ;
- TSA supprime le certificat de l'agent.

La figure V.15 illustre le diagramme de séquence de la sécurité d'achat.

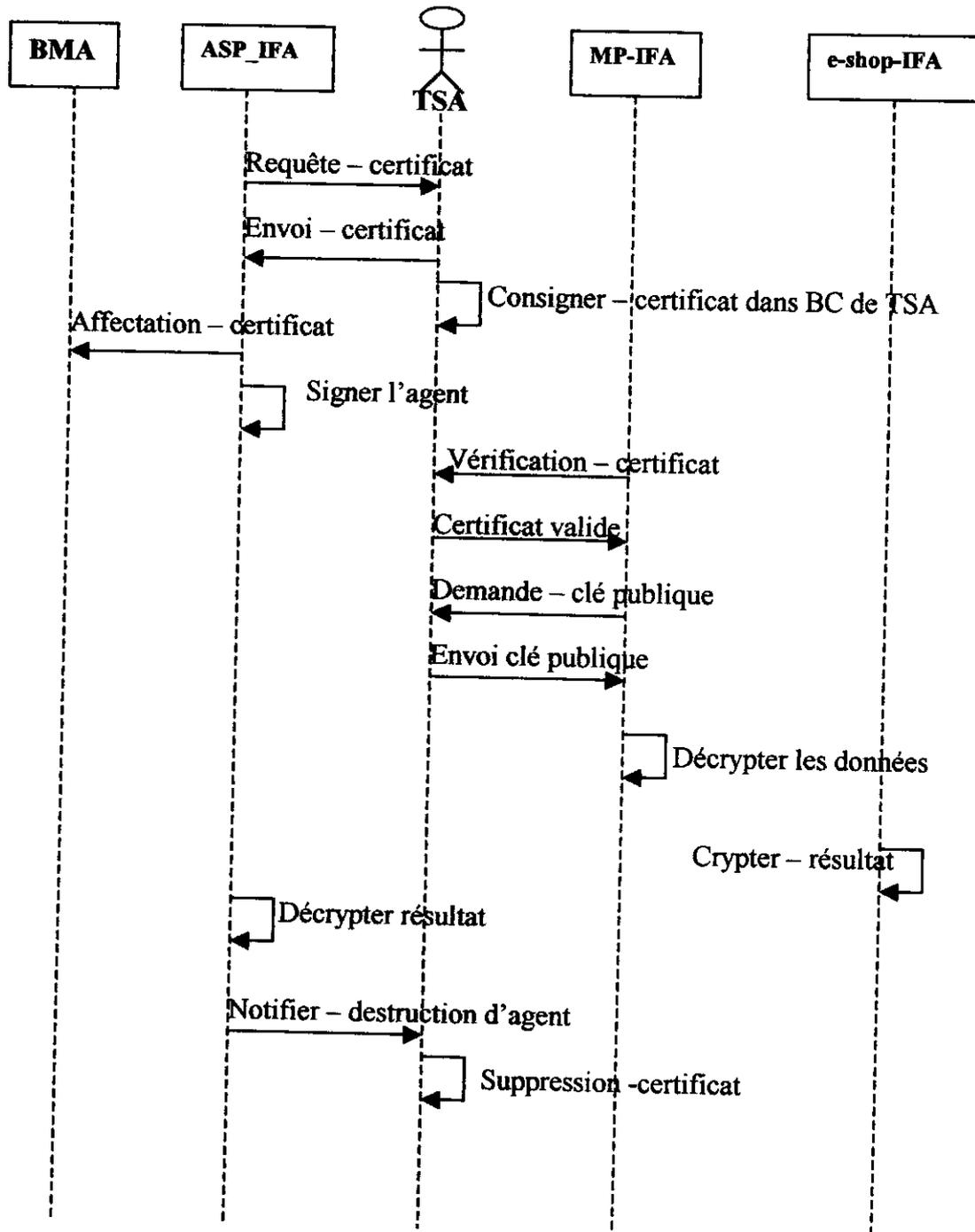


Figure V.15: Diagramme de séquence de la sécurité d'achat.

Scénario 7 : Procédure du paiement dans l'achat

- Le vendeur demande le numéro de compte virtuel (VAN) de l'acheteur ;
- L'acheteur envoie son VAN au vendeur ;
- Le vendeur vérifie le compte VAN auprès de l'autorité bancaire virtuelle (VBA) ;
- Le vendeur valide la transaction, l'acheteur revient avec les résultats des transactions monétaires effectuées.

La figure V.16 illustre le diagramme de séquence du paiement :

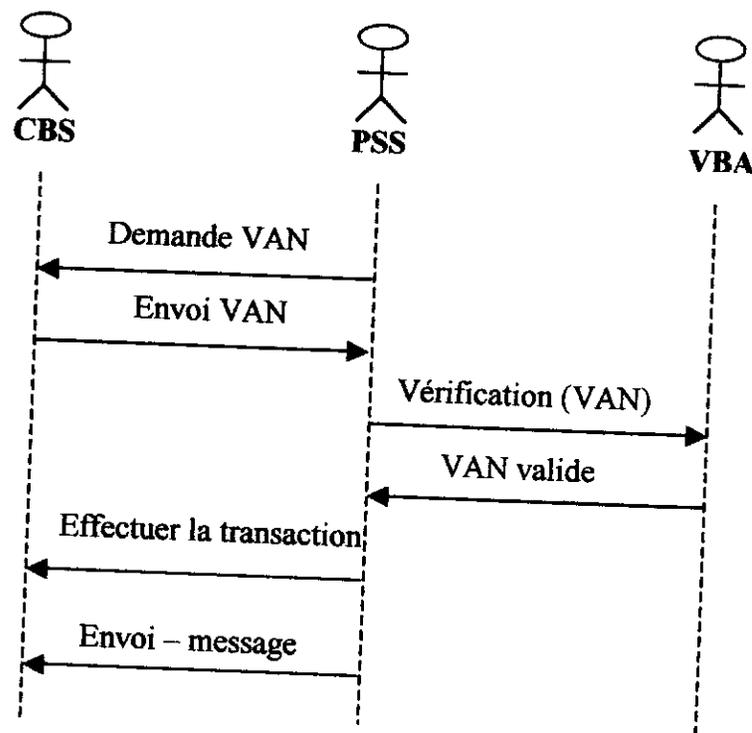


Figure V.16: Diagramme de séquence du paiement.

Scénario 8 : Modification de catalogue local

Notons que le catalogue local (LC) est unique pour chaque e-shop et contient toutes les informations sur tous les services disponibles dans ce dernier.

- ASP-IFA réactive l'agent acheteur ;
- L'agent acheteur demande une autorisation de migration à MP ;
- MP-IFA approuve la demande ;
- L'agent acheteur demande une autorisation de migration à l'e-shop ;
- e-shop-IFA approuve la demande ;
- L'agent acheteur demande la mise à jour du catalogue local ;
- e-shop-IFA met à jour le catalogue local ;
- e-shop-IFA notifie l'agent vendeur ;
- L'agent acheteur demande une autorisation de migration à MP ;
- MP-IFA approuve la demande ;
- L'agent acheteur demande une autorisation de migration à l'ASP ;
- ASP-IFA approuve la demande ;
- ASP-IFA détruit l'agent acheteur.

La figure V.17 illustre le diagramme de séquence de la modification de LC:

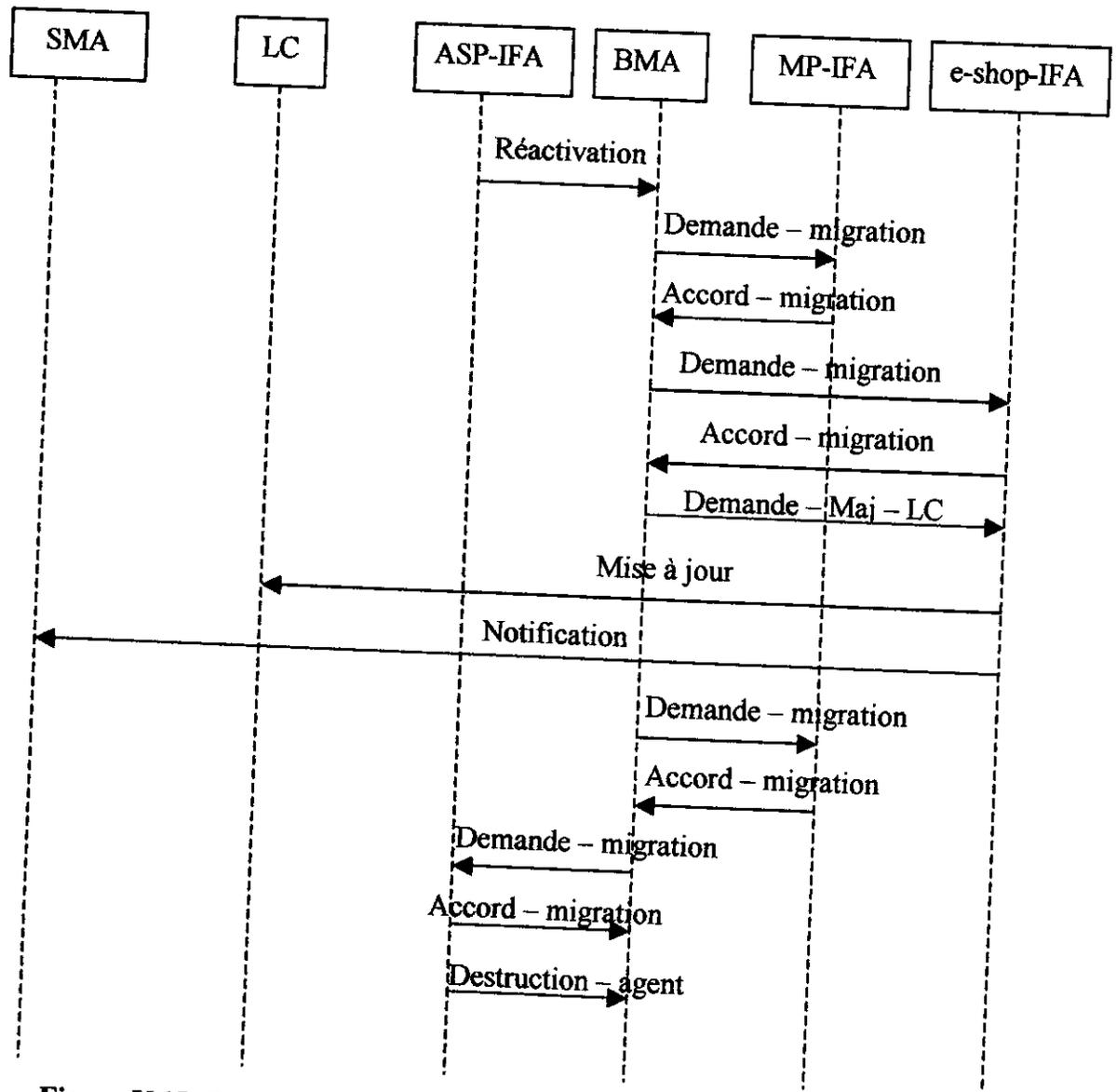


Figure V.17: Diagramme de séquence de la modification de catalogue local.

II.2. Vente d'un produit

On va décomposer aussi la fonction de vente en sous fonctions comme le montre le schéma suivant :

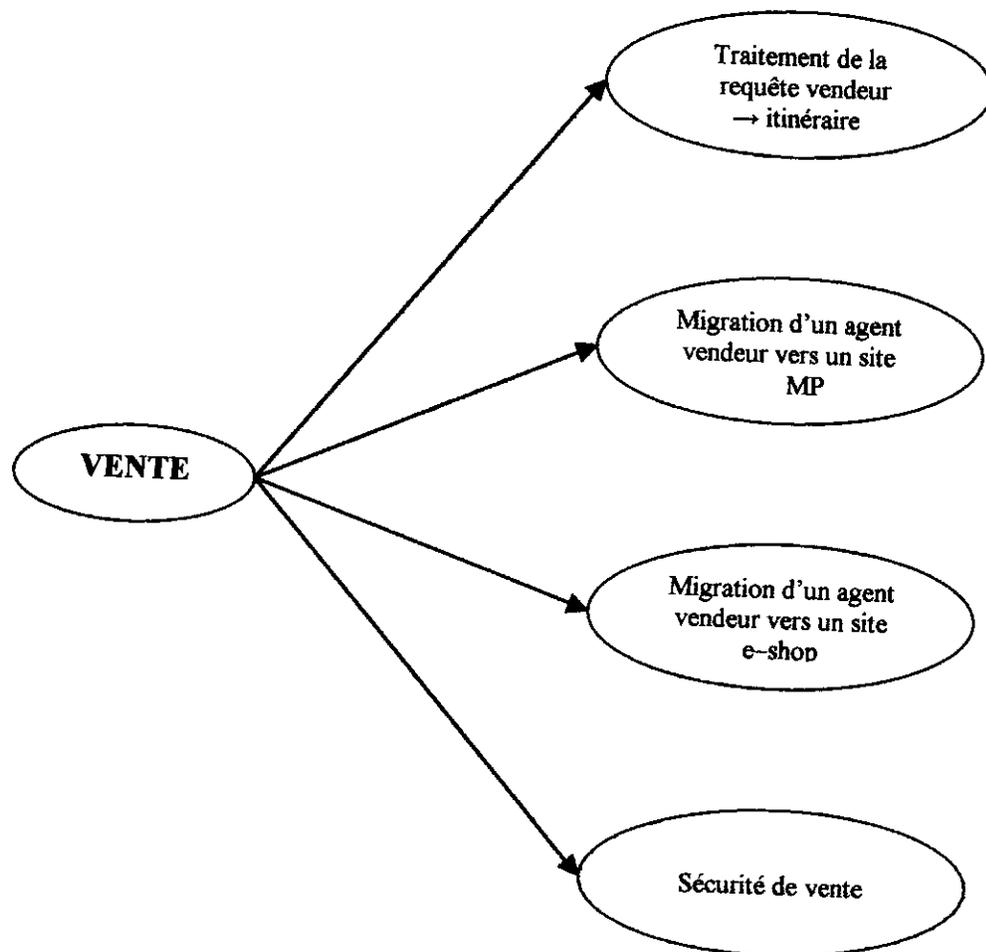


Figure V.18 : Décomposition de la fonction de vente.

Dans ce qui suit, on va détailler les sous fonctions montrées dans le schéma ci-dessus.

II.2.1. Diagramme de séquence

Scénario 1 : Traitement de la requête vendeur jusqu'à l'affectation d'itinéraire

- Le vendeur se connecte à l'ASP ;
- L'ASP envoie une fenêtre pour que le client puisse s'inscrire ;
- Le vendeur propose ses services en remplissant le formulaire fourni par la fenêtre;
- Le vendeur lance sa requête ;
- L'ASP crée un agent vendeur (SMA) pour exécuter la requête sur le réseau ;
- L'ASP demande au serveur MPNS un itinéraire initial ;
- L'MPNS propose un itinéraire à l'ASP ;
- L'ASP affecte l'itinéraire à l'agent vendeur déjà créé.

La figure V.19 illustre le diagramme de séquence du traitement de la requête vendeur jusqu'à l'affectation d'itinéraire :

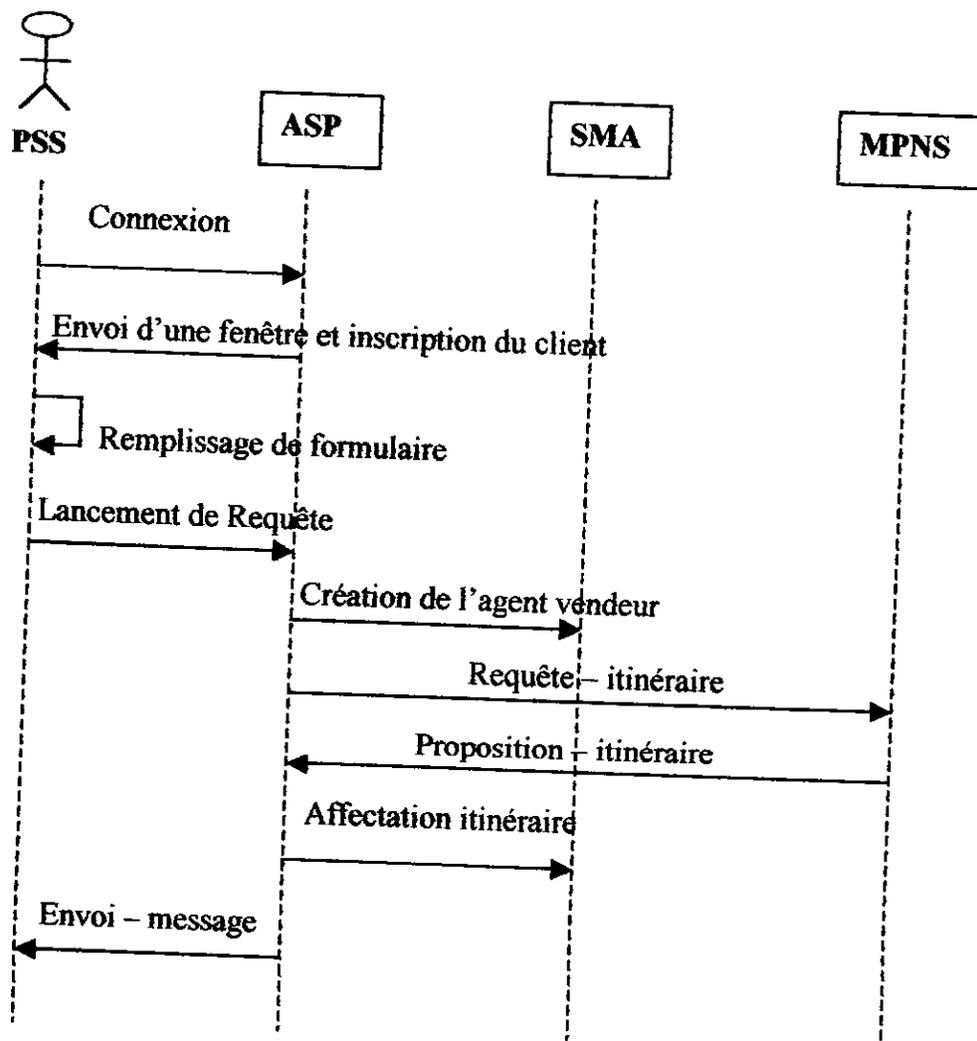


Figure V.19: Diagramme de séquence du traitement de la requête vendeur.

Scénario 2 : Migration d'un agent vendeur vers un site MP

- L'agent vendeur SMA demande une autorisation de migration à l'agent facilitateur distant (MP-IFA) du premier site MP de l'itinéraire ;
- L'agent MP-IFA approuve la demande ;
- L'agent vendeur SMA notifie sa migration à l'agent facilitateur local de l'ASP (ASP-IFA) ;
- Au niveau du premier site MP, l'agent vendeur demande d'être authentifié par le MPSM (pare feu) de ce site ;
- MPSM demande un jeton au MPDS du premier site ;
- MPDS envoie un jeton au MPSM ;
- MPSM affecte le jeton à l'agent vendeur, l'agent est donc authentifié et contrôlé par le MPSM ;
- L'agent vendeur authentifié demande à l'agent facilitateur (MP-IFA) de le router vers un e-shop ;
- MP-IFA consulte un annuaire (MPDS) ;
- Si la ressource a été trouvée, l'agent MP-IFA route l'agent vendeur (une sorte de sous itinéraire), sinon l'agent vendeur migre vers un autre site MP ;
- Le processus est répété sur tous les sites MP présents dans l'itinéraire tant que les ressources ne sont pas disponibles dans le présent site.

La figure V.20 représente le diagramme de séquence de la migration d'un agent vendeur vers un site MP :

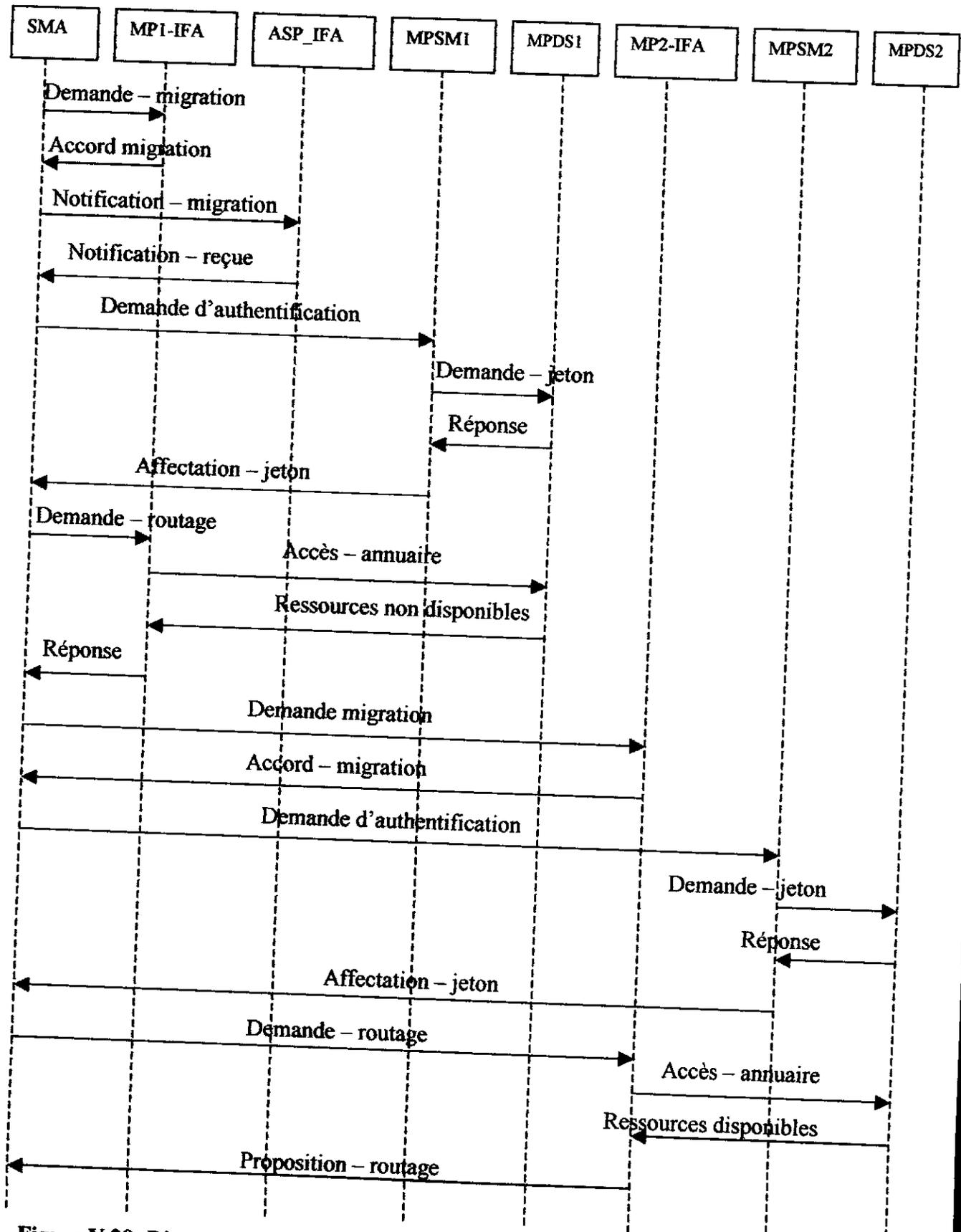


Figure V.20: Diagramme de séquence de la migration d'un agent vendeur vers un site MP.

Scénario 3 : Migration d'un agent vendeur vers un site e-shop

- L'agent vendeur SMA demande une autorisation de migration à l'agent facilitateur local e-shop-IFA du premier e-shop de sous itinéraire ;
- L'agent e-shop-IFA demande à l'agent vendeur d'envoyer son jeton ;
- L'agent e-shop-IFA approuve la demande après la vérification de jeton ;
- L'agent vendeur expose ses services à l'agent e-shop-IFA ;
- L'agent e-shop-IFA ajoute le vendeur au catalogue local (LC) ;
- L'agent vendeur reste pendant un temps T dans le site e-shop ;
- Si un acheteur est présent sur le e-shop, le facilitateur met le vendeur en contact avec l'acheteur trouvé, sinon l'agent vendeur migre vers un autre e-shop après la fin de T ;
- Le processus est répété sur tous les e-shops présents dans le site MP tant qu'il n'existe pas un agent acheteur sur le présent e-shop.

La figure V.21 illustre le diagramme de séquence de la migration d'un agent vendeur vers un site e-shop :

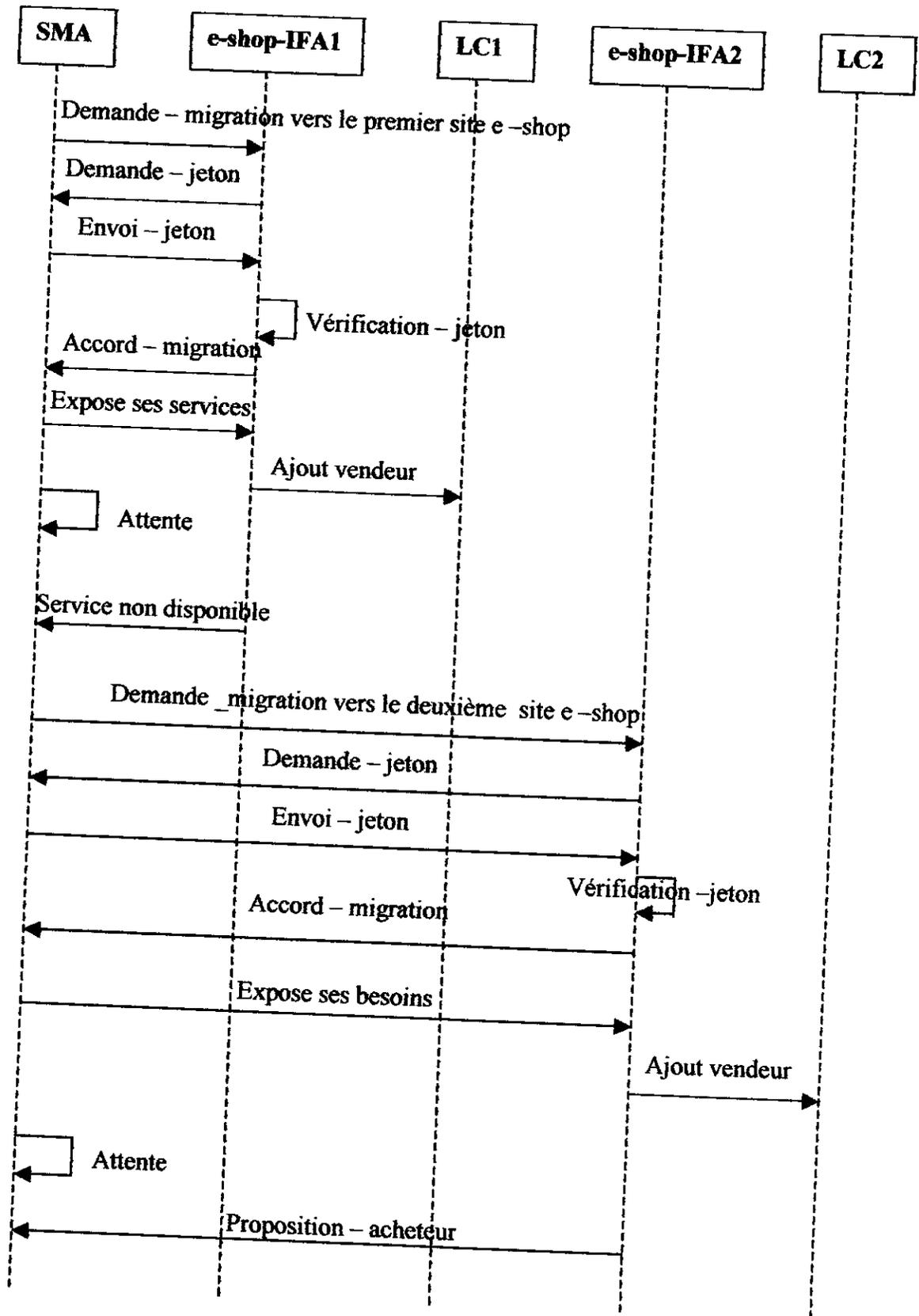


Figure V.21: Diagramme de séquence de la migration d'un agent vendeur vers un site e-shop.

Scénario 4 : Sécurité de vente

- L'ASP-IFA demande au TSA d'envoyer un certificat pour l'agent vendeur ;
- TSA transmet un certificat à l'ASP ;
- TSA consigne le certificat dans sa base de connaissance ;
- L'ASP-IFA affecte le certificat à l'agent vendeur déjà crée ;
- L'ASP-IFA signe l'agent vendeur par la clé privée de ce dernier ;
- MP-IFA vérifie si le certificat est valide auprès de la TSA;
- TSA envoie la réponse ;
- MP-IFA demande la clé publique de l'agent ;
- TSA envoie la clé publique ;
- MP-IFA décrypte les signatures de l'agent par la clé publique de ce dernier.

La figure V.22 illustre le diagramme de séquence de la sécurité de vente :

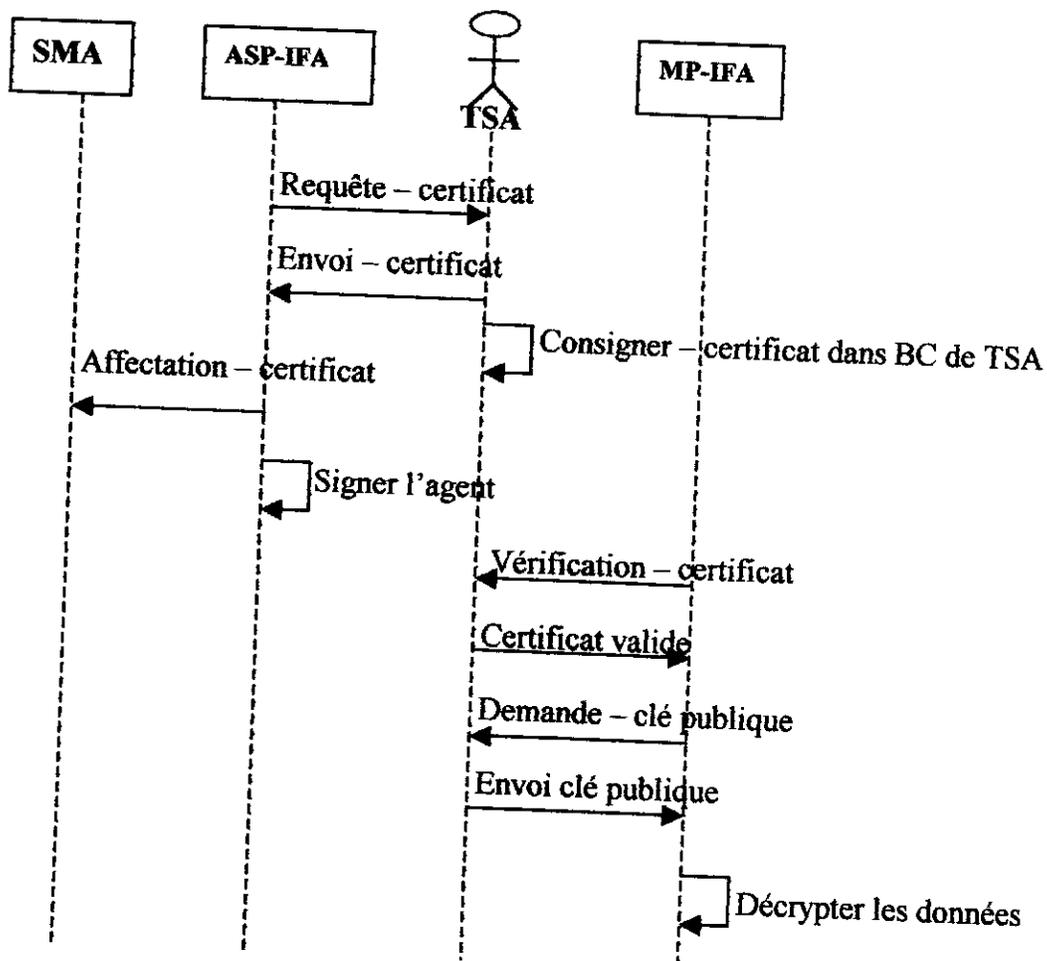


Figure V.22: Diagramme de séquence de la sécurité de vente.

II.3. Diagramme de classe général

La figure V.23 illustre notre système globalement à travers un diagramme de classe détaillé :

III. CONCLUSION

Dans ce chapitre et par l'intermédiaire de plusieurs diagrammes, nous avons essayé de bien expliquer la solution proposée tout en illustrant l'ensemble de ses composants et leur manière d'agir.

Le chapitre suivant portera sur la réalisation de la solution proposée.

Chapitre VI

Implémentation et test

INTRODUCTION

Dans ce chapitre, nous allons mettre en œuvre le système CESAM présenté dans le chapitre précédent. Ce système comporte deux types d'agents :

1. Les agents statiques : ce sont des agents de gestion (les agents facilitateurs).
2. Les agents mobiles : ce sont les agents délégués par l'utilisateur afin d'effectuer une tâche (les agents acheteurs et les agents vendeurs).

I. CHOIX DU LANGAGE DE PROGRAMMATION

Plusieurs langages destinés à la conception d'agent sont en train d'être développés, mais beaucoup de ces langages sont surtout destinés à mettre en évidence certains principes et contraintes, et ne sont pas encore destinés à devenir des standards. Java est considéré comme étant le langage le plus approprié pour la création des agents parce qu'il dispose d'une machine virtuelle qui procure l'indépendance du système d'exploitation et disponible sur la plupart des systèmes actuels.

Java est un langage de programmation Orienté-Objets qui permet de développer des programmes que l'on peut exécuter sur la machine virtuelle Java (JVM). Ce langage offre des mécanismes qui permet la mobilité des agents, les principaux mécanismes sont les suivants :

L1. Mobilité et portabilité du code

Une des propriétés essentielle de Java est que le code généré par le compilateur Java est mobile. Ceci signifie que le code peut être déplacé entre différentes machines. La mobilité du code nécessite que le code soit portable. La portabilité du code produit par Java provient du fait que le code (appelé byte code) est interprété par la JVM.

L2. Liaison dynamique

Un aspect crucial pour l'utilisation de code mobile est la liaison dynamique. Cette dernière signifie ici la possibilité de ne pas déterminer que lors de l'exécution le code qui sera exécuté pour un appel de méthode. Etant donné que Java permet le chargement dynamique des classes, une variable d'un type donné (une interface en Java) peut contenir une référence Java pointant sur une instance dont la classe à été chargée

dynamiquement. Java retarde la liaison du code à partir de cette variable jusqu'à l'appel effectif, permettant ainsi l'exécution de code chargé dynamiquement.

I.3. Sérialisation

La sérialisation permet de stocker l'état des objets, ainsi que la manière de recréer cet objet. Ces objets stockés sont appelés des *données série*, c'est à dire envoyée une par une (à la manière du port série d'un ordinateur) et non simultanément (comme le port parallèle).

II. RÉALISATION DU SYSTÈME CESAM

Dans ce qui suit, nous allons présenter l'implémentation du modèle de sécurité présenté dans la partie conception.

Le système que nous allons réaliser contient :

- Une place de marché qui est elle-même le site e-shop où l'on trouve les produits à acheter ou services à offrir pour les clients potentiels. Chaque e-shop dispose d'une base de données ou bien d'un catalogue local (LC) qui contient toutes les informations sur tous les services disponibles dans le e-shop.
- Un environnement d'exécution(ASP).
- Une autorité de sécurité (TSA).
- Une autorité bancaire virtuelle (VBA).

II.1. Réalisation d'un environnement d'exécution (ASP)

Un agent mobile nécessite un environnement d'exécution spécifique sur tous les sites de son itinéraire, un tel environnement doit permettre au minimum :

1. La création des agents.
2. Leur exécution.
3. Leurs migrations.
4. La communication entre agents.
5. La sécurité des agents.

Nous avons réalisé un environnement spécifique aux applications de commerce électronique manipulant les agents acheteurs et vendeurs et les places de marché.

II.1.1. Implémentation de l'agent mobile acheteur

Un agent acheteur est une instance d'une classe Java qu'on a définie, cette classe hérite de la classe SaBoAgent (Sali_Bouzidi_Agent), et elle assure :

- La communication avec d'autre agent.
- La négociation avec les agents vendeurs.

La figure ci-dessous illustre la forme de la classe SaBoAgent :

```

public class SaBoAgent implements SaBoAgentEventListener , Runnable {
.....
public void process() {};
public void reset() {};
public void initialize() {};

// requis par l'interface Runnable
public void stop() {};
public void run() {};
}

```

Figure VI.1 : La forme de la classe SaBoAgent.

La figure VI.2 illustre la forme de la classe agent acheteur :

```

public class AgentAcheteur extends SaBoAgent implements Serializable {
.....
SaBoAgentMessage msg ;
.....
public void process() {
initialize();
.....
msg = new SaBoAgentMessage("register:",name , null , null ,null , null) ;
.....
FacilitatorAgent.register(e) ; // l'agent acheteur s'enregistre auprès de facilitateur
.....
}
public void stop() {.....}
public void run() {.....}
.....
public void processMessage(SaBoAgentMessage msg) {.....}
.....
// la fonction de négociation
void negotiate(Offre Offre, SaBoAgentMessage msg) {.....}
.....
}

```

Figure VI.2 : La forme de la classe agent acheteur.

II.1.2. Implémentation de l'agent mobile vendeur

Un agent vendeur est une instance d'une classe Java qu'on a définie, cette classe hérite de la classe SaBoAgent, et elle assure :

- La communication avec d'autre agent.
- La négociation avec les agents acheteurs.

La figure VI.3 illustre la forme de la classe agent vendeur :

```

public class AgentVendeur extends SaBoAgent implements Serializable{
.....
SaBoAgentMessage msg ;
.....
public void process() {
initialize() ;
.....
msg = new SaBoAgentMessage("register:",name , null , null ,null , null) ;
.....
FacilitatorAgent.register(e) ; // l'agent vendeur s'enregistre auprès de facilitateur
.....
}
public void stop() {.....}
public void run() {.....}
.....
public void processMessage(SaBoAgentMessage msg) {.....}
void rejectOffre(Offre Offre) {.....}
.....
// la fonction de négociation
void negotiate(Offre Offre, SaBoAgentMessage msg) {.....}
.....
}

```

Figure VI.3 : La forme de la classe agent vendeur.

II.2. Simulation d'une autorité de sécurité (TSA)

Cette autorité permet de certifier les agents créés à l'ASP ainsi que les places de marché à visiter par les agents mobiles. Un certificat contient les informations suivantes :

- Un identificateur de propriétaire du certificat.
- La clé publique.
- La clé privée.

La clé publique sert pour le cryptage des données de l'agent en utilisant l'algorithme RSA et la clé privée pour le décryptage.

II.3. Réalisation d'une place de marché

La place de marché est composée d'un agent facilitateur, un ou plusieurs agents acheteurs, et un ou plusieurs agents vendeurs.

II.3.1. Implémentation de l'agent statique facilitateur

II.3.1.1. L'agent facilitateur de l'ASP

Cet agent permet de créer et de détruire les agents, de les activer et de les désactiver, d'affecter l'itinéraire aux agents mobiles créés, certifier ces agents en leur attribuant les certificats donnés par la TSA, les signer avant chaque migration, les transférer et de décrypter les résultats rapportés par ces derniers.

L'agent facilitateur (ASP_IFA :Intelligent Facilitator Agent) est un agent statique qui s'exécute sur l'ASP, sa classe hérite de la classe SaBoAgent.

II.3.1.2. L'agent facilitateur de MP

L'agent facilitateur (MP_IFA) est l'intermédiaire entre les acheteurs et les vendeurs. Tous les agents doivent se présenter à l'agent facilitateur avant qu'ils puissent interagir avec d'autres agents dans le marché.

Les vendeurs lui annoncent leurs désirs de vendre des produits ou des services, alors que les acheteurs lui demandent de recommander un vendeur éventuel.

L'agent facilitateur est un agent statique qui s'exécute sur la place de marché (chaque MP possède son propre agent facilitateur), sa classe hérite de la classe SaBoAgent. La figure VI.4 illustre la forme de la classe de facilitateur de MP :

```

public class MPFacilitator extends SaBoAgent {
.....
static Rsa rsa;
.....
public void process() {.....}
public void stop() {.....}
// requis par l'interface Runnable
public void run() {.....}
.....
public static synchronized void reception() {.....}
public synchronized void migration(SaBoAgentMessage msg) {.....}
.....
}

```

Figure VI.4 : La forme de la classe de l'agent facilitateur de MP.

Cet agent joue un grand rôle dans la sécurité, il permet de décrypter les données de l'agent mobile visiteur, et de crypter les résultats recueillis par ce dernier.

Toutes les communications entre les acheteurs et les vendeurs passent par l'agent facilitateur en utilisant des objets appelés SaBoAgentMessage, ces messages incluent des performative de langage KQML comme : register, advertise, unadvertise, recommend-one, ask, et tell. La figure VI.5 illustre la forme de la classe SaBoAgentMessage :

```

public class SaBoAgentMessage implements Serializable{
.....
SaBoAgentMessage(.....){.....}
.....
}

```

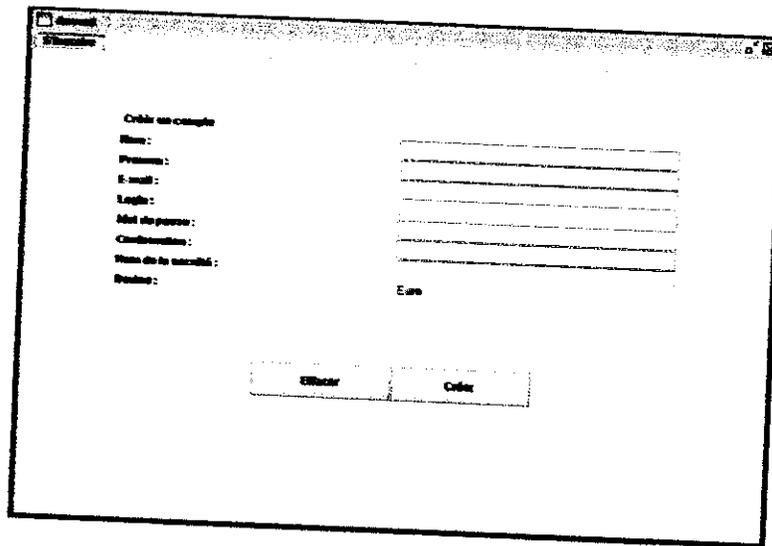
Figure VI.5 : La forme de la classe SaBoAgentMessage.

III. RÉALISATION DE L'APPLICATION

L'application se devise en deux sous applications, l'une c'est l'interface utilisateur et l'autre c'est le place de marché, elle manipule un système d'agent et assure 2 services :

- 1) L'achat d'un produit.
- 2) La vente d'un produit.

Lorsqu'un utilisateur démarre l'application, il se connecte directement à l'ASP, ce dernier lui envoie une fenêtre pour qu'il puisse s'inscrire. Chaque utilisateur inscrit est sauvegardé dans une base de donnée. La figure VI.6 montre l'interface à travers laquelle l'utilisateur peut s'inscrire.



The image shows a screenshot of a web browser window displaying a registration form. The form is titled "Créer un compte" and contains several input fields for user information. The fields are labeled as follows: "Nom:", "Prénom:", "E-mail:", "Login:", "Mot de passe:", "Confirmation:", "Date de naissance:", and "Sexe:". The "Mot de passe:" field has a small "E" icon next to it. Below the form, there are two buttons: "Effacer" and "Créer".

Figure VI.6 : Interface de l'inscription.

III.1. La spécification des besoins

Lorsqu'un utilisateur veut acheter un produit (respectivement vendre un produit), il spécifie ses besoins en remplissant un formulaire et délègue un agent mobile acheteur(respectivement vendeur) pour effectuer l'achat (respectivement la vente) à sa place sans qu'il le sache. Les données de l'agent crée sont sauvegardées dans une base de donnée avec l'identificateur de l'utilisateur correspondant. La figure VI.7 montre l'interface de spécification des besoins.

Figure VI.7 : Interface de spécification des besoins pour les acheteurs.

Si l'utilisateur est un vendeur il y aura l'ajout d'un autre champ qui est le nombre d'unité à vendre.

Figure VI.8 : Interface de spécification des besoins pour les vendeurs.

III.2. L'affectation des certificats

Lorsqu'un agent acheteur (respectivement vendeur) est créé, il doit être certifié par une autorité de sécurité (TSA) qui assure la sécurité de l'agent, l'intégrité et la confidentialité des données manipulées par l'agent pendant sa migration.

Notons que la place de marché à visiter par l'agent a été certifiée par la TSA pour assurer la protection de l'hôte et de l'agent. La figure VI.9 illustre le format d'un certificat.

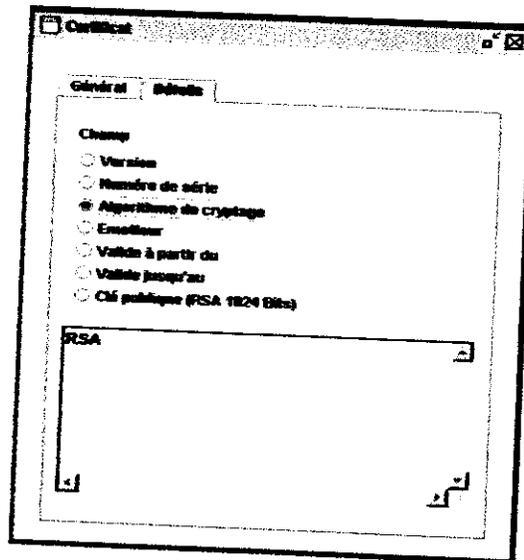


Figure VI.9 : Le format d'un certificat.

III.3. La place de marché

La place de marché est une autre application qui manipule un système d'agent en arrière plan.

Lorsque l'agent acheteur (respectivement l'agent vendeur) arrive à la place de marché et après l'authentification, il se présente à l'agent facilitateur pour le mettre en contact avec un vendeur intéressant (respectivement un acheteur intéressant). Les interactions entre agents seront visualisées sur une console.

III.4. L'affichage des résultats

Au retour de l'agent acheteur, les résultats trouvés seront affichés à l'utilisateur pour qu'il puisse choisir le produit à acheter. Pour l'opération de vente, les résultats à afficher sont des messages indiquant aux utilisateurs si leurs produits ont été ajoutés au catalogue local ou non.

III.5. Le paiement

L'utilisateur inscrit à l'ASP s'enregistre automatiquement dans une banque virtuelle, et cette dernière lui ouvre un compte pour qu'il puisse conclure les différentes transactions. La figure VI.10 montre la fenêtre à travers laquelle s'effectue le paiement.

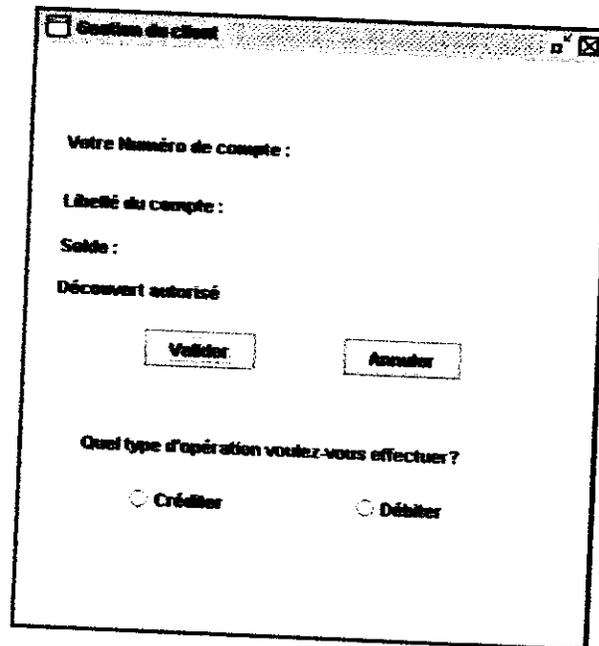


Figure VI.10 : L'interface du paiement.

Le commerce électronique permet l'utilisation de plusieurs formes de paiement soit le crédit ou le débit.

IV. TEST

Cette partie permet de tester l'efficacité du modèle de sécurité, ce test est réalisé sur une machine reliée à un réseau où sur l'une des machines il y a la place de marché et la base de donnée de la banque virtuelle, et sur l'autre il y a le serveur ASP et l'autorité de sécurité TSA.

Les produits en questions sont des équipements informatiques : disque dur, imprimante, carte graphique et mémoire.

Tableau VI.1 : Exemple.

	Vendeur	Acheteur
Nom	Alice	BOB
E-mail	Alice@yahoo.fr	BOB@yahoo.fr
Identificateur	Vendeur1	Acheteur1
Nom de produit	Imprimante jet d'encre	Imprimante jet d'encre
Marque de produit	Epson Stylus Photo R245	/
Prix min	129 £	100 £
Prix max	145 £	140 £
Nombre d'unité	50	/

Il y a d'autre information à saisir qui sont nécessaire à la connexion. Lorsque l'utilisateur démarre l'application, il se connecte directement à l'ASP où il s'inscrit et spécifie ses besoins.

Après que l'utilisateur (Acheteur ou vendeur) clique sur le bouton suivant, un agent mobile sera crée, et certifié.

La TSA crée une paire de clé (clé publique(1024 bits), clé privée) et affecte un certificat contenant la paire de clé à l'agent.

Avant la migration de l'agent à la place de marché pour exécuter la requête, l'ASP signe l'agent par la clé privée de ce dernier.

A l'arrivé de l'agent, l'agent facilitateur vérifie si l'agent visiteur a été certifié par la vraie TSA et non pas par un ennemi, si oui, les données de l'agent seront décryptées par sa clé publique qui sera envoyé à la demande, sinon, l'agent revient à son site initiateur.

Après la vérification de la validité de certificat, L'agent mobile expose ses besoins à l'agent facilitateur. Voici les différentes étapes parcourues par l'agent vendeur et acheteur :

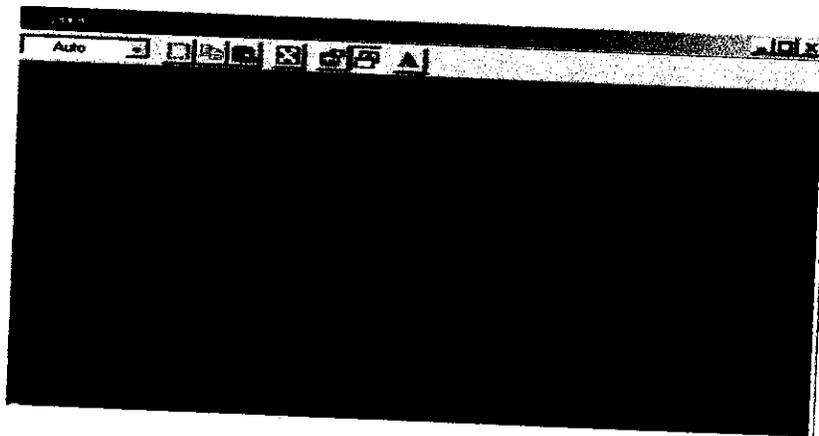


Figure VI.11: l'agent vendeur dans l'ASP.

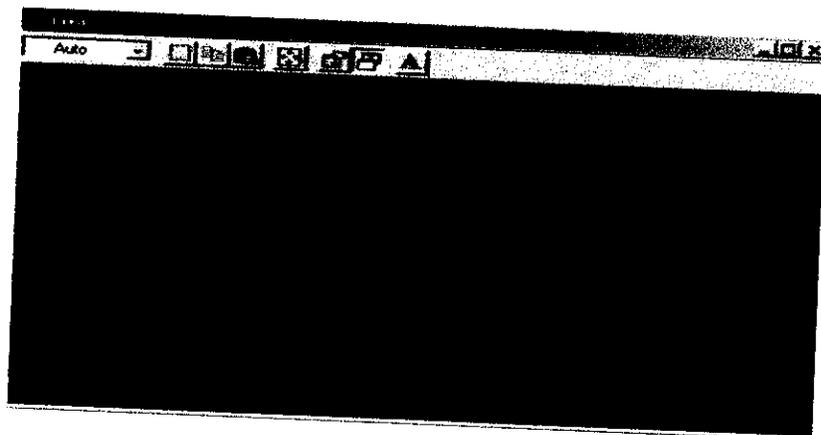


Figure VI.12: l'agent vendeur dans la place de marché

L'agent acheteur suit les mêmes étapes. Lors de son arrivé à la place de marché, Si un vendeur a été trouvé, il négocie avec lui sinon il revient au site initial. La figure VI.13 montre les interactions entre les agents dans la place de marché.

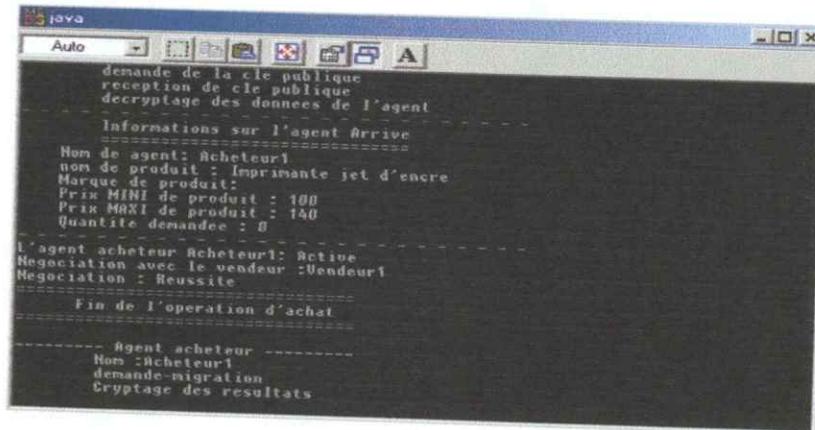


Figure VI.13 : La négociation dans la place de marché

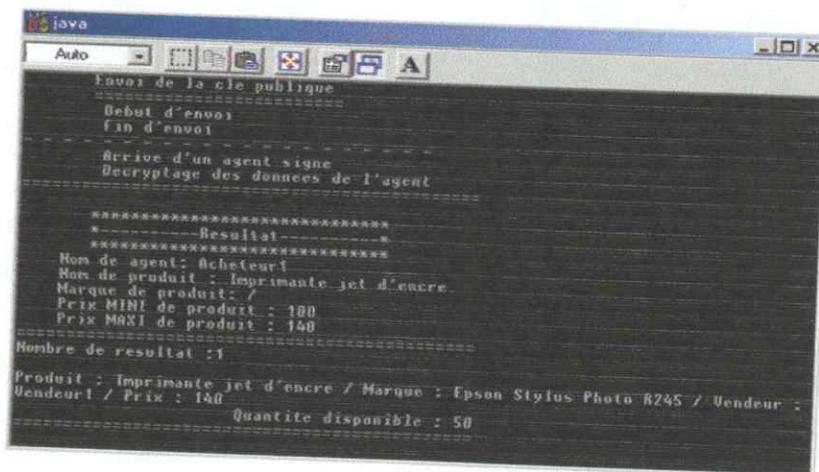


Figure VI.14: Retour de l'agent acheteur à l'ASP.

L'agent acheteur retourne au site initial avec les résultats trouvés qui sont cryptés par sa clé publique. A son arrivé, ces résultats seront décryptés et affichés à l'utilisateur. La figure VI.15 montre les résultats rapportés par l'agent acheteur :

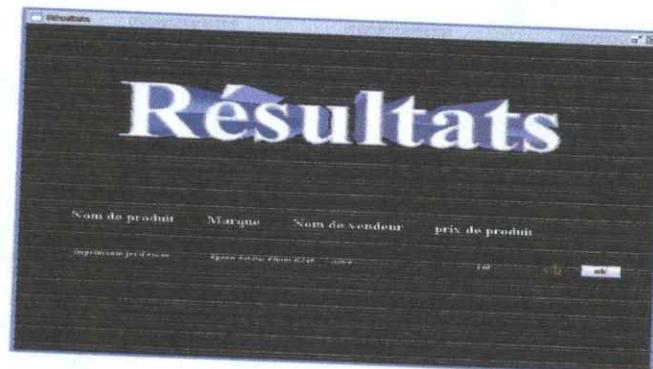


Figure VI.15 : L'interface d'affichage des résultats d'achat.

Lorsque l'utilisateur choisit un produit, le système lui demande de spécifier le nombre d'unité à acheter.

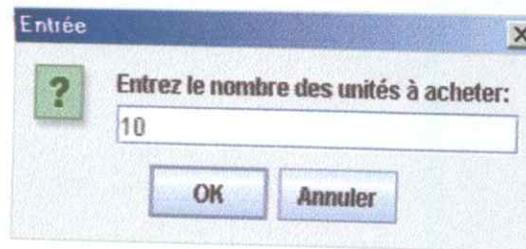


Figure VI.16 : La détermination de nombre d'unité.

Le Système calcule le prix total, désactive l'agent acheteur et demande à l'utilisateur de valider son achat. La figure VI.17 montre l'interface de validation d'achat :



Figure VI.17 : La validation d'achat.

Pour conclure l'achat, le système demande à l'utilisateur d'entrer le numéro de la carte de crédit qui sera crypté par la clé publique de l'agent acheteur.

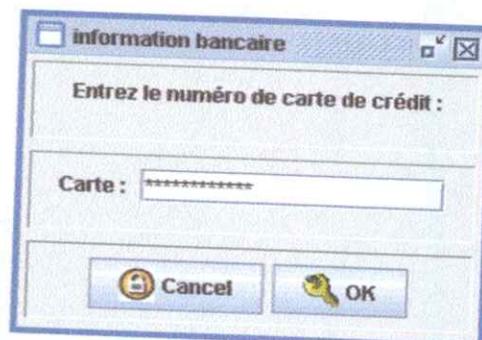


Figure VI.18 : Information bancaire.

Les étapes du paiement sont :

1. Créditer le compte de l'acheteur à partir de la carte de crédit.

Gestion du client

Votre Numéro de compte : 7002

Libellé du compte : Internet

Solde : 1400.0

Déouvert autorisé : 0.0

Quel type d'opération voulez-vous effectuer ?

Créditer Débiter

Montant : 1400

Effectuer

Quitter



Figure VI.19 : La fenêtre de crédit du compte acheteur.

2. Débiter le compte de l'acheteur vers le compte de vendeur.

Gestion du client

Numéro de compte de vendeur : 7001

Libellé du compte : Internet

Solde : 1400.0

Déouvert autorisé : 0.0

Quel type d'opération voulez-vous effectuer ?

Créditer Débiter

Montant : 1400

Effectuer

Quitter

Figure VI.20 : La fenêtre de débit du compte acheteur.

Après le paiement, l'ASP réactive l'agent acheteur, ce dernier migre vers MP, demande à l'agent facilitateur de mettre à jour la quantité de produit de Vendeur1(Alice) dans le catalogue local, notifie l'agent vendeur, retourne à l'ASP où il sera détruit automatiquement.

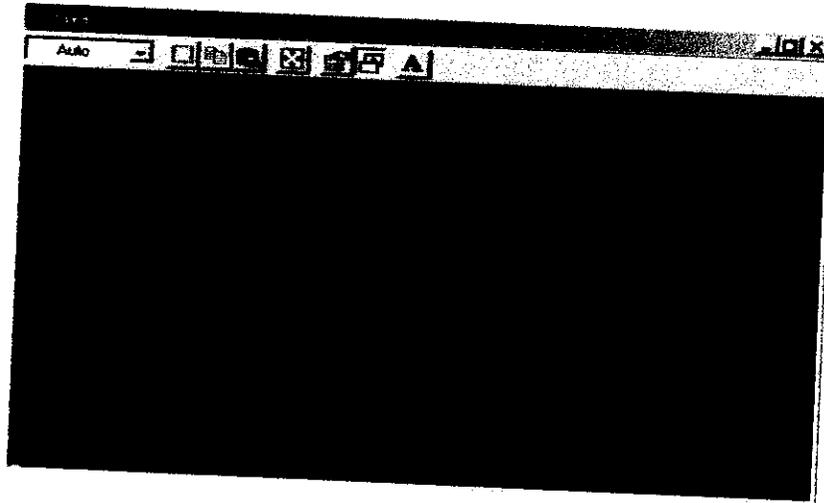


Figure VL.21 : Modification de catalogue local.

L'agent vendeur sera détruit automatiquement lorsque sa quantité de produit à vendre sera vide.

Conclusion générale

Le développement de l'ingénierie logicielle des vingt dernières années a montré un mouvement vers des paradigmes et des langages qui utilisent des concepts abstraits provenant du domaine du problème lui-même, la technologie des agents est l'une de ces paradigmes.

Aujourd'hui, la technologie "agent mobile" est de plus en plus attractive vu les avantages qu'elle offre pour la réalisation des tâches complexes et répétitives dans des systèmes ouverts et dynamiques. Néanmoins, l'utilisation de cette technologie dans des applications réelles est encore loin des attentes des chercheurs, et reste limitée, à cause des problèmes de sécurité, notamment ce qui concerne la protection de l'agent vis-à-vis de l'hôte, et la protection de l'hôte contre les agents.

Dans ce travail, nous avons d'abord présenté le domaine des agents, plus particulièrement les agents mobiles, ensuite nous avons défini les applications réparties notamment le commerce électronique et les problèmes rencontrés lors de l'utilisation des agents mobiles dans ce type d'application. A la fin, nous avons proposé un modèle de sécurité qui répondra aux préoccupations déjà citées.

Les objectifs :

Nous allons maintenant revoir l'essentiel de ce que nous avons accompli comme travail à la lumière des objectifs que nous nous sommes fixés.

1) **Étudier le concept d'agent et d'agent mobile**

Nous avons présenté les différentes définitions, les caractéristiques, l'environnement et les typologies des agents, Ajoutant les interactions et les divers langages de communication entre ces derniers. Ensuite nous avons présenté les agents mobiles comme un nouveau paradigme pour les applications réparties.

2) **Présenter les applications réparties et le domaine de commerce électronique**

Nous avons présenté les applications réparties et nous avons étudié en détail l'une de ces applications qui est : **le commerce électronique**. A la fin, nous avons abordé le problème de sécurité posé par les agents mobiles.

3) Proposer un modèle de sécurité pour les agents mobiles

Nous avons présenté plusieurs solutions en montrant l'incapacité de chacune, ensuite nous avons proposé un modèle de sécurité qui est basé sur le modèle acheteur / vendeur.

4) Décrire la solution proposée

Nous avons décrit la solution en détail en utilisant le langage UML.

5) Une mise en œuvre possible de l'architecture proposée

Pour illustrer le modèle de sécurité proposé, nous avons implémenté une application de type e-commerce basée sur les agents mobiles (agents acheteurs et agents vendeurs). Cette application est réalisée par le langage de programmation JAVA.

Perspectives :

Inévitablement notre application n'est pas parfaite, il reste encore des choses à ajouter :

- La sécurité est un domaine vaste, car même si on fait des efforts, on reste loin de pouvoir traiter tous les problèmes, pour l'instant le système fonctionne d'une manière acceptable sur un réseau local, mais sur Internet, c'est un autre problème.
- Réalisation complète de l'architecture proposée, car nous avons considéré dans notre application un seul site MP qui est lui-même le site e-shop. Donc la migration de l'agent d'un site MP vers un autre site MP ou d'un site e-shop vers un autre site e-shop est absente.
- Une étude ergonomique plus avancée pour l'interface utilisateur.

Enfin, ce modeste travail nous a permis d'acquérir une vraie expérience sur le terrain et de voir les difficultés qui peuvent être rencontrées ainsi que les solutions qui peuvent leur être apportées.

Bibliographie

- [1]: Jean-Pierre Biot . & Yves Demazeau . « principe et architecture des systèmes multi-agent » Lavoisier 2001.
- [2]: Philippe SELLEM. « Accès à des bases de données réparties à travers le Web : Methode de recherche (agent intelligents)»2000.
- [3]: AKLOUF Y. & DRIAS H., 'VEMMA :Une architecture de place de marché à base d'agent mobiles et intelligents',in proc of ISPS'03,Alger, Mai 2003.
- [4]: Adam E. Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise :application aux systèmes administratifs complexes. Thèse de Doctorat, Université de Valenciennes et du Hainaut-Cambésis, septembre, 2000.
- [5]: Grislin-Le Strugeon E., Adam E., Kolski C. (2001).Agents intelligents en interaction Homme-Machine dans les Systèmes d'information. Chapitre 7, dans *Environnements évolué et évaluation de l'I.H.M., Interaction Homme-Machine pour les S.I. 2*, C. Kolski (Ed.), Editions HERMES, Paris, pp. 209-248.
- [6]: Chaib-draa B. Interaction between agents in routine, familiar and unfamiliar situations. *International Journal of Intelligent and Cooperative Information Systems*, vol. 1, n° 5, p. 7-20, 1996.
- [7]: Kuflik T., Shoval P. User Profile Generation for Intelligent Information Agents – Research in Progress. *Proceedings of the Conference on Advanced Information Systems Engineering*, CaiSE'00, Stockholm, Suède, 2000.
- [8]: Nwana Hyacint S. Software Agent : an overview. *Knowledge Engineering Review*, vol. 11, n° 3, p. 205-244,1996.
- [9]: Lant T. K. Computer Simulations of Organizations as Experiential Learning Systems:Implications for Organization Theory. In *Computational Organization Theory*,Carley K. M. & Prietula M.J. (Eds.), London : LEA publishers,p. 195-215, 1994.
- [10]: Koda T. *Agents with Faces: A Study on the Effects of Personification of Software Agents*. M.S. Thesis, Massachusetts Institute of Technology, 1996.
- [11]: Lashkari Y., Metral M. et Maes P. Collaborative Interface Agents. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, AAAI '94 (1994), Seattle, USA, AAAI Press, 1994.
- [12]: Lieberman H. Autonomous Interface Agents. *Proceedings of the Conference on Human Factors in Computing Systems*, CHI'97, Atlanta,USA, Pemberton S. (Ed.), New-York, USA: ACM Press, p. 67-74, 1997.

Bibliographie

- [13]: Laurel B. Interface Agents : Metaphors with Character. In *Software Agent*, J. Bradshaw (Ed.), AAAI Press, p. 67-77, 1997.
- [14]: Rhodes J. *Just-In-Time Information Retrieval*. Ph.D. Thesis, MIT Media Lab, 2000.
- [15]: Rus D., Gray R. et Kotz D. Transportable Information Agent, *Journal of Intelligent Information systems*, vol. 9, n° 3, p. 215-238, 1997.
- [16]: L. Gulyás, L. Kovács, A. Micsik, B. Pataki, I. zsámboki, << An Overview of Mobile Software Systems >>, Departement of Distributed System, MTA SZTAKI, Computer and Automation Research Institute of the Hungarian Academy of Sciences MTA SZTAKI Technical Report, Contains work prepared under the KNIXMAS (Knowledge shared XPS-based Research Network Using Multi-Agent Systems) EU-Inco N°:977113 Project, 2000.
- [17]: Sylvie Pesty, Carine Webber et Nicolas Balacheff, Baghera: une architecture multi-agents pour l'apprentissage humain, Grenoble, Laboratoire Leibniz IMAG.
- [18]: http://cours2.fsa.ulaval.ca/cours/sio65293/agents/introduction/KMCenter_Agent.pdf
- [19]: Yan Wang, Jian Ren: « Building Internet Marketplaces on the Basis of Mobile Agents for parallel Processing », Department of Computer Science, National University of Singapore, Proc. Of 3rd International Conference on Mobile Data Management (MDM2002), IEEE Computer Society Press, Jan. 8-11 2002, Singapore, pp.61-68.
- [20]: <http://www.decisionnel.net/agentintelligent/index.htm>.
- [21]: Odyssey (General Magic).
http://www.genmagic.com/technology/mobile_agent.html.
- [22]: Aglets (IBM Corporation).
<http://www.trl.ibm.co.jp/aglets/>.
- [23]: Danny B. Lange and Yariv Aridor. *Agent Transfer Protocol – ATP/0.1*. IBM Corporation, 19 mars 1997.
- [24]: Concordia (Mitsubishi Electric).
<http://www.meitca.com/hsl/projects/concordia/>.
- [25]: Voyager (ObjectSpace). <http://www.objectspace.com/voyager/>.
- [26]: Russell, Stuart J. and Peter Norvig (1995), **Artificial Intelligence: A Modern Approach**, Englewood Cliffs, NJ: Prentice Hall.
- [27]: Maes, Pattie (1995), "Artificial Life Meets Entertainment: Life like Autonomous Agents," *Communications of the ACM*, 38, 11, 108-114.

Bibliographie

- [28]: Smith, D. C., A. Cypher and J. Spohrer (1994), "KidSim: Programming Agents Without a Programming Language," *Communications of the ACM*, 37, 7, 55-67.
- [29]: Hayes-Roth, B. (1995). "An Architecture for Adaptive Intelligent Systems," *Artificial Intelligence: Special Issue on Agents and Interactivity*, 72, 329-365, .
- [30]: Jim White. Ibm tokyo labs-general magic meeting. Sunnyvale CA, 1 novembre 1996..
- [31]: <http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm>
- [32]: Wooldridge, Michael and Nicholas R. Jennings (1995), "Agent Theories, Architectures, and Languages: a Survey," in Wooldridge and Jennings Eds., *Intelligent Agents*, Berlin: Springer-Verlag, 1-22.
- [33]: J-P.Arcangeli & V.Hennebert & S.Leriche & F.Migeon & M.pantel : « JAVACT 0.5.0: principes, installation, utilisation et développemnt d'aplications ». Thèse de doctorat, Institut de recherche en informatique de Toulouse. Rapport IRIT/2004-5-R.
- [34]: Hohl(Fritz).- Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts. *Mobile Agents and Security*, 1998, pp.92-113.
- [35]: White (James E.). _ *Telescript Technology : The Fondation for the Electronic Marketplace.* _ rapport technique (*white paper*), 2465 Latham Street , Mountain View, CA,USA, General Magic, Inc.,1994.
- [36]: Hohl (Fritz). _ *The Mobile Agent List : Names only Table of Known Mobile Agent Systems.* In : mole.informatik.uni-stuttgart.de/mal/mal.html.
- [37]: James E.White : <<*Telescript technology : an introduction to the language*>>, General Magic White paper GM-M-TSWP-0495-V1, General Magic Inc.,1995.
- [38]: NOIE, « Business to Business E-Commerce Case Study: Transport and Logistics-Australia », présentation au Comité de travail du Groupe sur l'économie de l'information de l'OCDE, document reprographié, 17 juin 1999., p. 2.
- [39]: Thorn (Tommy). _ *programming languages for Mobile Code.* ACM computing surveys, Vol.29,n3,1997,pp.213-239.
- [40]: Yee (Bennet S.),- A Sanctuary for Mobile Agents. In : *secure Internet Programming; Security Issues for Mobile and Distributed Objects*, éd, par Vitek (Jan) et Jensen (Christian D.), pp.261-273. - *Lecture Notes in Computer Science*, n 1603, Springer, 1999.

Bibliographie

- [41]: David Wong, Noemi Paciorek, Tom Walsh, Joe DiCelie, Mike Young, Bill Peet, <<Concordia : An Infrastructure for Collaborating Mobile Agents >>, Proceedings of the First International Workshop on Mobile Agents, MA'97, Berlin, Allemagne, 7-8 avril, 1997.
- [42]: James E. White, C. S. Helgeson, and D. A. Steedman, <<System and method for distributed computation based upon the movement, execution, and interaction of processes in a network>>, US Patent 5,603,031, filed 8 filed 1993, issued 11 February 1997.
- [43]: Leila Ismail, Daniel Hagimont « Spécialisation de serveurs par des agents mobiles » Projet SIRAC (IMAG-INRIA), INRIA, 655 av. de l'Europe, 38330 Montbonnot Saint-Martin, France.
- [44]: OCDE, The Economic and Societal Impacts of Electronic Commerce : Preliminary Findings and Research Agenda, février 1999.
- [45]: HAYES Frank, " The history so far ", *Computerworld*, 6 June 2002, Vol 36, p 24.
- [46]: M. Anthony Rutkowski (Vice-président, General Magic Incorporation) Genève, 10-14 novembre 1997 « MISE EN OEUVRE DU PROGRAMME D'ACTION POUR LA TROISIEME DECENNIE DE LA LUTTE CONTRE LE RACISME ET LA DICRIMINATION RACIALE ».
- [47]: S. Krakowiak. *Principes des systèmes d'exploitation des ordinateurs*. Dunod, 1985.
- [48]: Sahra Bouchenak-Khelladi : « Mécanismes pour la migration de processus. Extension de la machine virtuelle Java », I.M.A.G. DEA D'INFORMATIQUE : SYSTEMES ET COMMUNICATIONS, Institut National Polytechnique de Grenoble, Juin 1998.
- [49]: Stéphane PERRET : « Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure ». Thèse de doctorat, Université Joseph Fourier - Grenoble I, Novembre 1997.
- [50]: Siegfried ROUVRAIS : « Utilisation d'agents mobiles pour la construction de services distribués ». Thèse de doctorat, Université Rennes 1, Juillet 2002.
- [51]: Foundation for Intelligent Physical Agents (www.fipa.org). FIPA Abstract Architecture Specification, December 2002. statut : « Standard, version L ».
- [52]: D. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran, and J. White. MASIF: The OMG Mobile Agent System Interoperability Facility. In K. Rothermel And F. Hohl, editors, Proceedings of the 2nd International Workshop on Mobile Agents, Volume 1477 of Lecture Notes in Computer Science, pages 50-67. Springer-Verlag : Heidelberg, Germany, 1998.

Bibliographie

- [53]: <http://www.grasshopper.de/>.
- [54]: Stamos (James W.) et Gifford (David K.).- Remote Evaluation. ACM Transactions on programming languages and systems, Vol.12,n4,Octobre 1990,pp.537-565.
- [55]: Shoch (John F.) et Hupp (Jon A.).- The « Worm » Programs- Early Experience with Distributed Computing. Communications of the ACM, Vol.25,n3, Mars 1982, pp.172-180.
- [56]: Object Serialization. - documentation, Sun Microsystems, Inc. Java.sun.com/J2se/1.3/docs/guide/serialization/.
- [57]: Chess (David M.), Harrison (Colin G.) et Kershebaum (Aaron).- Mobile Agents: Are They a good Idea?- rapport technique nRC 19887, T.J- Watson Research Center, NY,USA,IBM Research Division, Février 1994.
- [58]: Cardelli (Luca).- Abstractions for mobile computations. Secure Internet programming, Security Issues for Mobile and Distributed Objects,1999, pp.51-94.
- [59]: OMG. The Common Object Request Broker: Architecture and Specification, OMG Document Number 91.12.1, Revision 1.1, Décembre 1991.
- [60]: <http://strategis.ic.gc.ca/epic/internet/inecom-come.nsf/fr/>.
- [61]: GERARD Ribière " Architecture en systèmes d'information IBM ".
- [62]: Guillaume Desgeorge "La sécurité des réseaux",Mars 2000.
- [63]: http://www.cvm.qc.ca/mlaflamme/comm_e/intro/avantag.html.
- [64]: http://www.cvm.qc.ca/mlaflamme/comm_e/intro/avenir.html.
- [65]: Wooldridge M., *Multi-Agent Systems: a modern approach to DAI*, Weiss G. editor, June 1999.
- [66]: www.dicisionnel.net/agentintelligent.
- [67]: <http://www.enligne.fr/comparer-les-prix/agents/agoff.html>.
- [68]: <http://cours2.fsa.ulaval.ca/cours/sio-65293/agents/e-commerce/agent-vendeurs.html>.
- [69]: www.uml.free.fr.
- [70]: <http://cours2.fsa.ulaval.ca/cours/sio-65293/agents/négociation>.

Bibliographie

- [71]: Pr. P.Y. Schobbens – FUNDP. P. Massonet, C. Ponsard – CETIC "agents intelligents au service de la gestion électronique de l'entreprise? ",24/09/2003.
- [72]: U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating Word of Mouth" proceedings of the CHI-95 Conference, Denver, CO, ACM Press, May 1995.
- [73]: Articles/ Définition du cahier des charges et fixation des objectifs www.supralogic.com.
- [74]: George C. Necula and Peter Lee. Safe, untrusted agents using proof-carrying code. In Giovanni Vigna, editor, Mobile Agent Security, Lecture Notes in Computer Science No. 1419, pages 61–91. Springer-Verlag : Heidelberg, Germany, 1998.
- [75]: S. Loureiro. Mobile Code Protection. Thèse de doctorat, ENST Paris / Institut Eurecom,2001.
- [76]: John Zachary. Protecting mobile code in the wild. IEEE Internet Computing, 7(2) :78–82, 2003.
- [77]: Djamel Eddine MENACER. « Un modèle d'architecture à base d'agents mobiles pour les applications réparties ». Thèse de Magister, Institut National d'informatique - Alger.
- [78]: D.Russel & G.T. Gangemi : «Computer Security Basics », O'Reilly & Associates, 1991.
- [79]: D. Kotz, R. Gray, and D. Rus : "Transportable Agents Support Worldwide Applications", Department of Computer Science, Dartmouth College, Hanover, NH, Mars 1996.
- [80]: D. Chess, C. Harrison et A. Kershenbaum, "Mobile Agents: are they a good idea ?", IBM Research Report, RC 19887, Décembre 1994.
- [81]: McIcan (John).- Security Models. Encyclopedia of Software Engineering (Ed . John Marciniak).Wiley and Sons. Inc.1994.
- [82]: Hohl (Fritz).- A Model of Attacks of Malicious Hosts against Mobile Agents, In :4 th ECOOP Workshop on Mobile object systems (MOS 98): Secure Internet Mobile Computations.- 1998.
- [83]: Young (Adam) et Yung(Moti).- Sliding Encryption: A Cryptographic Tool for Mobile Agents. In : Proceedings of Fast Software Encryption Workshop. pp. 230- 241.- lecture Notes in Computer Science, n 1267, Springer, 1997.
- [84]: Vigna (Giovanni).-Protecting Mobile Agents through Tracing. In : 3rd ECOOP Workshop on Mobile object Systems.- juin 1997.

Bibliographie

[85]: Sander (Thomas) et Tschudin (Christian F.)- Protecting Mobile Agents Against Malicious Hosts. In : Mobile Agents and security, éd, par Vigna (G.). pp.44-60.- lecture Notes in Computer Science, n 1419, Springer, 1998.

[86]: pierre-Alain Muller,modélisation objet avec UML,EYROLLES,2001.

[87]: Hartmut Vogler, Marie-Luise Moschgath, Thomas KunKelman : «Enhancing Mobile Agents with Electronic Commerce capabilities», Darmstadt University of technology Information Technology Transfer Office, Wilhelminenstr. 7, D-64283 Darmstadt, Germany, 1999.

Annexe

L'algorithme RSA

INTRODUCTION

I. Cryptographie à clé publique

Le but de ce genre de cryptographie est de résoudre le problème de la distribution des clés secrètes employée dans la cryptographie symétrique. Introduite par Diffie et Hellman en 1975 avec l'algorithme connu sous le nom de Diffie-Hellman, ce procédé utilise deux clés :

Une publique à partager pour le cryptage et l'autre privée à garder secrète pour le décryptage.

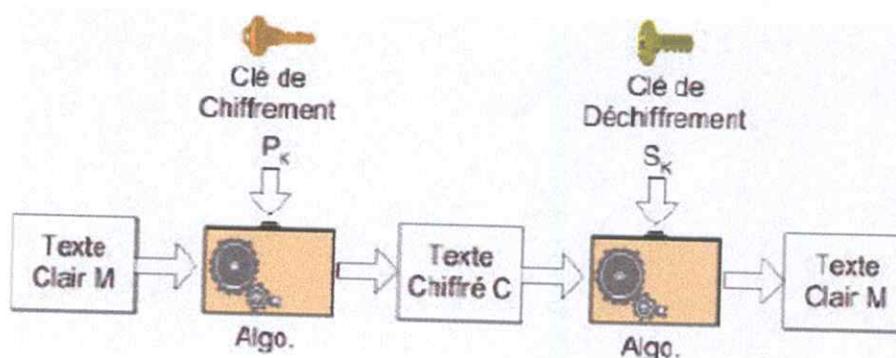
Les deux clés sont établies soit par leur propriétaire, soit par une autorité à laquelle il est rattaché. Elles sont, généralement, liées par des relations mathématiques, mais il reste difficile de deviner la clé privée à partir de la clé publique. Cela est dû au choix de clés de grande taille.

Ce type de cryptage comporte les avantages suivants :

- Permettre l'échange des données d'une façon sécurisée sans une contrainte sur le partage des clés : pas de transmission des clés,
- Pas besoin de partager la même clé : les clés sont distinctes,
- Permettre la signature numérique des documents : une empreinte identifiant le propriétaire du document et vérifiant l'intégrité des messages,
- et aussi l'authentification : pour s'assurer de l'identité des parties communicantes.

Parmi les algorithmes utilisant la cryptographie asymétrique, notons : RSA (Rivest Shamir Adelman), Elgamel, Diffie-Hellman, DSA (Digital Signature Algorithm).

La figure 1 illustre le principe de la cryptographie à clé publique :



Sk (pour Secret key ou clé secrète).
Pk (pour Public key ou clé publique).

Figure 1 : Cryptage à clé publique.

II. L'ALGORITHME RSA

II.1. Définition

R.S.A. signifie Rivest-Shamir-Adleman, en l'honneur de ses inventeurs : Ron Rivest, Adi Shamir et Leonard Adleman qui l'ont inventé en 1977. Le brevet de cet algorithme appartenait jusqu'au 6 septembre 2000 à la société américaine RSA Data Security, qui fait maintenant partie de Security Dynamics et aux Public Key Partners, (PKP à Sunnyvale, Californie, États-Unis). RSA est un algorithme à clé publique qui sert aussi bien au chiffrement de documents, qu'à l'authentification. Comme il est à clé publique et qu'il a résisté à toutes les attaques jusqu'à présent, l'algorithme RSA est devenu un standard *de facto* dans le monde.

Tout le principe de RSA repose sur le fait qu'il est très difficile et long de décomposer un très grand nombre en deux facteurs premiers.

II.2. Principe de fonctionnement

Sa sécurité repose sur la condition que :

- Le calcul du produit de deux nombres premiers est aisé et,
- La factorisation d'un nombre en ses deux facteurs premiers est beaucoup plus complexe.

Soit M un message en clair, C le message crypté, et e, d, n des nombres entiers tels que (e, n) constitue la clé publique, (d, n) constitue la clé privée et n est le produit de deux nombres premiers très grands.

Pour crypter un message M on utilise la formule : $C = M^e \text{ mod } n$.
Et pour décrypter : $M = C^d \text{ mod } n$.

Pour rendre l'algorithme incassable, les nombres e , d et n doivent être très grands. Le calcul de ce triplet reste un problème délicat, car ils doivent être, en même temps, grands pour ne pas être cassés facilement et petits pour une rapidité des algorithmes de cryptage et décryptage.

Leur calcul s'effectue de la sorte :

- Trouver deux nombres premiers p et q (de taille égale en nombre de bits) et calculer n tel que $n = pq$.
- Une fois cette étape finie, il faut trouver un nombre e premier avec : $(p-1)(q-1)$.
- La troisième et dernière étape consiste à calculer d tel que $ed \bmod (p-1)(q-1) = 1$.

Parmi les inconvénients de RSA nous citons :

- La taille des clés (de l'ordre de 1024) influant sur sa rapidité,
- et la factorisation du nombre n en deux nombres premiers, constituant la clé de décryptage, en un temps fini mais très grand. La factorisation est donc coûteuse en temps machine et en ressources, mais reste possible.

Annexe

UML

I. UML EN BREF

UML (Unified Modeling Language, que l'on peut traduire par " langage de modélisation unifié ") est une notation permettant de modéliser un problème de façon standard. Ce langage est né en 1994 de la fusion des trois méthodes de modélisation d'objet :

- OMT : Object Modeling Technique (créé par Jim Rumbaugh).
- BOOCH : Nom tiré de son inventeur (Grady Booch).
- OOSE : Object Oriented Software Engineering.

UML est devenu la référence en terme de modélisation objet. [69]

II. LES DIAGRAMMES UML

II.1. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation représente les cas d'utilisations, les acteurs et les relations entre les cas d'utilisation et les acteurs.

Un cas d'utilisation est un classificateur qui modélise une fonctionnalité d'un système ou d'une classe. L'instanciation d'un cas d'utilisation se traduit par l'échange de messages entre le système et ses acteurs. [86]

II.2. Diagramme de séquence

Les diagrammes de séquences nous permettent de montrer les interactions entre objets, ils nous permettent de bien schématiser les scénarios des cas d'utilisation.

La représentation se concentre sur la séquence des interactions selon un point de vue temporel. Ils sont en général plus aptes à modéliser les aspects dynamiques des systèmes temps réel et des scénarios complexes mettant en œuvre peu d'objets.

Un diagramme de séquence représente une interaction entre objets, en insistant sur la chronologie des envois de message.

Un objet est matérialisé par un rectangle et une barre verticale, appelée ligne de vie des objets, cette barre verticale représente l'écoulement de temps (de haut en bas). Les objets communiquent en échangeant des messages représentés au moyen de flèches horizontales, orientées de l'émetteur du message vers le destinataire. L'ordre d'envoi des

messages est ainsi donné par la position de ces messages sur les lignes de vie des objets. [86]

II.3. Diagramme de classe

Le diagramme de classe exprime de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. [86]

III. AVANTAGES D'UML

Le principal avantage d'UML c'est qu'il est devenu le standard en terme de modélisation objet, universellement reconnu celui-ci est polyvalent et performant.

Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus est un autre avantage de langage de modélisation UML.

IV. INCONVÉNIENTS D'UML

Son point faible est sans contestation possible, la lourdeur (relative) de sa mise en place au sein de n'importe quel processus. Son apprentissage assez long et rigoureux peut également être un frein à son utilisation.