

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE.

UNIVERSITE MENTOURI CONSTANTINE.
FACULTE DES SCIENCES DE L'INGENIEUR.
DEPARTEMENT D'INFORMATIQUE.

N° d'ordre :

Série :

THESE

Présentée pour obtenir le grade de

DOCTORAT EN SCIENCES EN INFORMATIQUE

CONTRIBUTION À LA COOPÉRATION DANS LES SYSTÈMES D'INFORMATION BASÉS WEB

Présentée par :

Mr. BRAHIMI Mahmoud

Dirigée par :

Pr. BOUFAIDA Mahmoud. Université Mentouri Constantine.

Devant le jury :

Président : Pr. BOUFAIDA Zizette. Professeur, Université Mentouri Constantine.

Rapporteur : Pr. BOUFAIDA Mahmoud. Professeur, Université Mentouri Constantine.

Examineurs : Dr. Kazar Okba. M.C, Université de Biskra.

Dr. Benchikha Fouzia. M.C, Université de Skikda.

Dr. Zarour Nacereddine. M.C, Université Mentouri Constantine.

Remerciements

Louange à Dieu le tout puissant, qui m'a donné la volonté, le pouvoir, la santé et la patience pour terminer ce travail et pour tout ce qu'il a effectué dans ma vie et pour ce qu'il continue à faire.

Ce travail a été effectué sous la direction de Monsieur Mahmoud BOUFAIDA, Professeur à l'université Mentouri de Constantine qui trouve ici l'expression de ma gratitude. Je lui exprime mes profonds remerciements pour l'aide compétente qu'il m'a apportée, pour son œil critique et pour sa patience et son encouragement à finir ce travail. Je le remercie de m'avoir accueilli dans son équipe et d'avoir été à mon entière disposition. Je le remercie très fort et je m'excuse auprès de sa noble personne de lui avoir pris beaucoup de son temps.

Ensuite je tiens à remercier Monsieur Lionel SEINTURIER, Professeur à l'université de Lille, France, pour son chaleureux accueil qu'il m'a réservé au sein du Laboratoire LIP6 à Paris au cours de mes stages, pour son assistance continue et pour ses conseils très judicieux.

Mes remerciements les plus vifs aux membres du jury qui ont accepté de valoriser ce travail pour me fournir des précieuses remarques pour l'amélioration et le raffinement de ce travail. Ainsi, je remercie :

Mme. Boufaida Zizette, Professeur à l'université Mentouri de Constantine

Mme. Benchikha Fouzia, Maître de Conférences à l'université de Skikda

Mr. Kazar Okba, Maître de Conférence à l'Université de Biskra

Mr. Zarour Nacereddine, Maître de conférence à l'université Mentouri Constantine.

Je tiens également à exprimer toute ma gratitude à tous les membres du Laboratoire LIRE, notamment notre équipe SI et BC et à tous les collègues de l'université de M'sila Pour leur Assistance, leur aide et leur encouragement.

SOMMAIRE

Introduction générale

| | |
|---|---|
| 1. Contexte du travail de recherche | 1 |
| 2. Problematique..... | 2 |
| 3. Contribution de la these..... | 3 |
| 4. Organisation de la these | 4 |

CHAPITRE I : E-business & coopération: Les nouvelles tendances de l'entreprise

| | |
|---|----|
| 1. Introduction | 6 |
| 2. Nouvelle economie & impacts des tic..... | 7 |
| 3. Nouvelle economie & défis de l'entreprise..... | 8 |
| 4. E-business..... | 12 |
| 4.1. Typologie des applications E-Business..... | 13 |
| 4.1.1. Le business-to-business (B2B)..... | 15 |
| 4.1.2. Le business-to-consumer (B2C)..... | 15 |
| 4.1.3. L'intermédiation..... | 15 |
| 4.2. Avantages et valeurs ajoutées du e-business..... | 16 |
| 4.3. Défis & problèmes du e-business..... | 18 |
| 4.3.1. Problèmes humains | 18 |
| 4.3.2. Problèmes de gestion..... | 18 |
| 4.3.3. Problèmes juridiques | 18 |
| 4.3.4. Problèmes techniques | 19 |
| 5. Coopération | 20 |
| 5.1. Typologie de cooperation..... | 21 |
| 5.2. Objectifs de la cooperation interentreprises | 22 |
| 5.3. E-business cooperatif | 23 |
| 6. Conclusion..... | 24 |

CHAPITRE II : Systèmes multi agents & business électronique

| | |
|---|----|
| 1. Introduction | 25 |
| 2. Du holon a l'agent: Historique de modélisation dans l'entreprise | 26 |
| 3. Systemes Multi-Agents : Concepts & principes de base..... | 29 |
| 3.1. Agents..... | 29 |
| 3.2. Differentes categories d'agents | 31 |
| 3.2.1. Les agents réactifs | 31 |
| 3.2.2. Les agents cognitifs..... | 31 |
| 3.3. Interaction entre Agents | 32 |
| 3.3.1. Collaboration simple | 34 |
| 3.3.2. Encombrement | 35 |
| 3.3.3. Collaboration coordonnee | 35 |
| 3.4. Communication entre agents..... | 35 |
| 3.4.1.1. Langage de communication KQML..... | 37 |
| 3.4.1.2. Langage de communication FIPA-ACL | 37 |

| | |
|--|----|
| 4. Systemes multi-agents & business electronique | 39 |
| 4.1. Typologie des agents E-business..... | 39 |
| 4.1.1. Agents d'application | 40 |
| 4.1.2. Agents personnels (ou d'interface) | 40 |
| 4.1.3. Agents courtiers..... | 40 |
| 4.1.4. Agents de négociation | 40 |
| 4.1.5. Agents mobiles | 43 |
| 5. Plate-formes & applications basées agents | 43 |
| 6. Conclusion..... | 46 |

CHAPITRE III : Web Services & E-business : Vers un middleware universel

| | |
|---|----|
| 1. Introduction | 47 |
| 2. Historique des Web Services..... | 48 |
| 3. Web Services: notions & concepts de base..... | 49 |
| 4. Opportunités des Web Services..... | 50 |
| 5. Entourage technologique des Web Services | 51 |
| 5.1. SOAP (Simple Object Access Protocol) | 52 |
| 5.2. UDDI (Universal Description, Discovery and Integration) | 53 |
| 5.3. WSDL (Web Service Description Language)..... | 53 |
| 5.4. Ebxml | 54 |
| 5.5. Rosettanet..... | 57 |
| 5.6. Biztalk | 58 |
| 6. Web Services & Web sémantique..... | 59 |
| 6.1. Semantic Web Services Language (SWSL)..... | 61 |
| 6.2. OWL-S (Ontology Web Language for Services)..... | 61 |
| 7. Développement des Web Services | 63 |
| 7.1. Critères techniques pour le développement | 64 |
| 7.2. Outils techniques pour le développement des Web Services | 66 |
| 8. Conclusion..... | 67 |

CHAPITRE IV : Un couplage agent et web service pour le développement des applications e-business coopératives

| | |
|---|----|
| 1. Introduction | 68 |
| 2. Couplage agent-Web Service | 69 |
| 3. Aperçu de l'approche | 71 |
| 4. Un modèle noyau à base d'agents | 72 |
| 4.1. Groupes d'applications..... | 74 |
| 4.1.1. Agent d'application | 75 |
| 4.1.1.1. Module de communication..... | 75 |
| 4.1.1.2. Boîte aux lettres..... | 76 |
| 4.1.1.3. Module de gestion des informations | 76 |
| 4.1.2. Agent coordinateur..... | 77 |
| 4.1.2.1. Module de communication..... | 77 |
| 4.1.2.2. Boîte aux lettres..... | 78 |
| 4.1.2.3. Répertoires de services..... | 78 |
| 4.1.2.4. Gestionnaire d'ontologie | 78 |
| 4.1.3. Protocole d'interaction & de coopération | 78 |
| 4.1.3.1. Interaction interne : | 80 |

| | |
|---|----|
| 4.1.3.2. Interaction externe : | 81 |
| 5. Intégration des Web Services | 83 |
| 5.1. Web Services agent finder | 85 |
| 5.2. Cas de correspondances des web services. | 86 |
| 6. Conclusion | 88 |

CHAPITRE V : Implémentation & Etude de cas

| | |
|--|------------|
| 1. Introduction | 90 |
| 2. Outils et aspects techniques pour l'implantation | 91 |
| 2.1. Développement des agents | 92 |
| 2.2. Développement des web services | 92 |
| 2.3. Développement, parsing & interrogation des ontologies | 93 |
| 2.3.1. Protégé-2000 version 3.4 | 94 |
| 2.3.2. Semweb Version 1.062 | 94 |
| 2.3.3. Langage d'interrogation SPARQL | 94 |
| 2.3.4. IKVM Version 0.38 | 94 |
| 3. Etude de cas | 95 |
| 3.1. Problématique | 95 |
| 3.2. Développement de l'application | 95 |
| 3.3. Critères pour l'implantation | 99 |
| 3.3.1. Coté agents | 99 |
| 3.3.2. Coté Web Services | 99 |
| 3.3.3. Coté interaction | 102 |
| 4. Conclusion | 103 |
| Conclusion générale | 104 |
| Références bibliographiques | 107 |

INTRODUCTION GENERALE

1. CONTEXTE DU TRAVAIL DE RECHERCHE

Durant ces dernières années, et face à l'évolution rapide des aspects et besoins technologiques liés aux nouvelles technologies de l'information et de la communication (NTIC), les ambitions des entreprises et leurs systèmes informatiques sont devenus de plus en plus sophistiqués, et leurs environnements d'exécution de plus en plus complexes.

Dues à cette évolution, les applications des affaires électroniques* ou (E-business pour Electronic Business) ont pris une importante place dans la nouvelle économie. L'e-business est une notion neuve dont le contenu constitue non seulement l'ensemble des équipements informatiques et des logiciels reliant entre elles des entreprises ou une entreprise et un consommateur dans le but de faire des affaires mais aussi des processus d'affaires, de soutien, de formation, de financement, etc. Néanmoins, certains éléments de cette définition peuvent être dégagés et il est important de préciser dès à présent que le e-business représente bien plus que le simple fait de vendre des produits à des clients en ligne via un site web; le commerce électronique n'est, en fait, qu'un aspect de l'e-business. Étant un concept récent et complexe, le fonctionnement de l'e-business engendre des relations et situations nouvelles entraînant, par là même, de nombreuses questions et problèmes techniques, économiques, humains et même juridiques.

Une analyse approfondie sur les objectifs de ce domaine nous permet de conclure que les applications du e-business doivent regrouper des stratégies de fusions, d'alliances ou d'impartition visant à optimiser les échanges d'affaires par l'utilisation des nouvelles technologies d'information. Elles doivent aussi inclure des procédés d'affaires comme l'approvisionnement électronique, le paiement électronique, la vente en ligne, le suivi électronique de la livraison ou l'encaissement électronique. De ce fait, sont rares, les entreprises qui ont les capacités et les ressources pour atteindre toutes les opportunités de ce domaine notamment pour celles classées comme petites et moyennes entreprises (PME). L'entreprise doit donc faire appel à d'autres entreprises et de sous-traitants en formant une

* Dans le jargon de la nouvelle économie on utilise souvent le terme « E-business », même pour les francophones.

alliance pour réunir leurs compétences et travailler ensemble afin de réaliser leurs objectifs. La question de l'intégration d'une entreprise dans un milieu coopératif avec d'autres entreprises est devenue plus que vitale. Les accords interentreprises permettent, par ailleurs d'atteindre beaucoup plus rapidement une masse critique permettant de réaliser des économies d'échelle. Les structures organisationnelles deviennent plus légères, plus souples, et , grâce à la conclusion d'accords ou à la création de structures de coopération, les entreprises peuvent mettre en place des stratégies de qualité, développer leur capacité d'innovation ou augmenter leur productivité ainsi qu'accroître leur flexibilité. En bref, la mise en place de techniques de coopération permet d'accroître la compétitivité de l'entreprise (Thiberge, 2006). La coopération interentreprises peut se voir comme une organisation technique et économique permettant à des entreprises ayant des intérêts communs de travailler ensemble avec le souci de l'objectif général. L'organisation technique de ce mode de travail nous ramène d'invoquer dans ce stade l'utilisation des technologies de l'information et de la communication (TIC) et par conséquent l'e-business dans le sens large du mot.

De ce fait, nous constatons que le couplage entre ces deux grands besoins (coopération et business électronique) avec l'instauration des applications e-business coopératives est plus que nécessaire pour pouvoir parler d'une nouvelle économie avec des hautes mesures de performance.

2. PROBLEMATIQUE

La mise en œuvre des applications e-business coopératives est fortement compliquée. Plusieurs obstacles humains, juridiques, de gestion et techniques peuvent influencer le processus de développement de telles applications (Petras & Tadas, 2004), (Alloui & al, 2002). Les processus d'organisation ne sont pas réorientés quand la numérisation arrive, afin d'adapter les entreprises correctement. Ces dernières ont des modèles de travail basés beaucoup plus sur des relations et des liens personnels, et la peur du changement est considérablement élevée. Cette peur s'accroît avec l'absence de méthodologies pour le développement des projets e-business et le manque de professionnels qui ont une vue d'ensemble générale sur les projets e-business, et sur la gestion de point de vue technologique. De ce fait, les solutions e-business restent toujours coûteuses notamment sur le niveau applicatif (logiciel). Par conséquent, le marché

informatique connaît un manque d'outils techniques permettant de donner des réponses complètes sur les préoccupations du e-business. Ce peu d'outils qui existe ne donne que de réponses partielles orientées beaucoup plus vers les problèmes de communication et d'échanges d'informations. La technologie de l'information n'est qu'une technologie pour le partage efficace, le traitement et la gestion de l'information, et actuellement elle ne peut devenir ni un canal de vente ni un vrais support pour les principes de gestion. En bref, Le marché de la nouvelle économie réclame toujours l'existence des approches complètes qui donnent des réponses exhaustives non seulement sur le plan technique mais aussi sur le plan de gestion.

La grande utilité du domaine du e-business avec les nombreux obstacles qu'il connaît, constituent un champ de développement très séduisant dans lequel plusieurs solutions intelligentes ont été proposées (Khubaib & Mansoor-uz-Zafar, 2008). Nous constatons ici, que la plupart des réponses contribuées dans ce domaine restent partielles et limitées sur quelques besoins élémentaires comme les problèmes de communication et les problèmes d'interaction (Wang & al, 2006) ou encore les problèmes de médiation (Sheng-Uei & Fangming, 2002).

3. CONTRIBUTION DE LA THESE

Notre objectif est de proposer un modèle qui joue le rôle d'un support technique et de gestion pour le développement des applications e-business coopératives. Nous constatons que le domaine de travail exige des techniques avec lesquelles les systèmes multi agent (SMA) (Ferber, 1997) apparaissent comme le paradigme d'implantation le plus adéquat. La faculté de l'agent qui lui permette de supporter les ambitions des acteurs e-business vienne à l'origine de ses aptitudes de coopération, de communication et d'autonomie. Mais dans un contexte tel que l'e-business, ce paradigme n'est pas suffisant avec l'ambition des gestionnaires et l'apparition du nouveau concept « Web Services» (Gillbert & al, 2003). Ce dernier semble être comme la meilleure réponse aux problèmes liés aux mécanismes et les standards du Web. Les Web services apparaissent comme des modules qui peuvent être publiés, localisés, et invoqués à travers le Web avec les moindres exigences techniques. La notion d'agent avec toutes ses potentialités et celle du Web service avec sa souplesse constituent un soutien énorme pour la résolution des problèmes du domaine du e-business coopératif.

Dans cette perspective, nous proposons un modèle à base d'agents avec l'intégration des Web services pour le développement des applications e-business coopératives. Pour répondre aux besoins de nos objectifs, ce modèle est basé essentiellement sur deux concepts clés : l'agent et le web service. La partie agent est le noyau de ce modèle, elle est représentée par un ensemble de groupes identiques. Chaque groupe possède un ensemble d'agents d'application dirigés par un agent coordinateur. Ce noyau permet de réduire le manque de sécurité et la vulnérabilité des applications e-business avec l'optimisation de l'utilisation des ressources (en hommes et/ou en machines). Il permet également la gestion de plusieurs activités concourantes et concurrentes en même temps avec la supervision des activités pour pouvoir réagir aux événements imprévus et résoudre dynamiquement les éventuels problèmes.

De plus, nous avons proposé les fonctionnalités publiques ou celle destinées au monde externe comme Web services afin de libérer le noyau agent. Ces Web services sont manipulés à l'aide d'un agent médiateur appelé Agent Finder. Ces Web services permettent la capture et la mise en disposition de toutes les informations nécessaires aux partenaires de la coopération.

Une dernière contribution est proposée dans ce travail pour montrer techniquement l'utilité du couplage agent/Web services. Cette contribution consiste à fixer l'ensemble des outils de développement permettant l'implantation des applications e-business coopératif. Cet ensemble s'articule généralement autour de la plate-forme DotNet. Finalement, une étude de cas concerne une petite société de vente de livre est proposée. Cette étude combine les activités du e-commerce et celles de l'e-administration et elle permet de mieux comprendre et cerner les éléments nécessaires de notre solution.

4. ORGANISATION DE LA THESE

Le travail présenté dans ce manuscrit est divisé en cinq chapitres. Les trois premiers chapitres constituent un état de l'art et par conséquent ils dévoilent le nécessaire de concept pour la bonne compréhension du contexte de notre travail. Les deux derniers chapitres sont consacrés à la présentation de notre contribution.

Le premier chapitre décrit le contexte de la nouvelle économie et les problèmes actuels des entreprises. Il présente le domaine du e-business avec ses critères en montrant par la suite la nécessité d'avoir des applications coopératives du e-business. Nous y justifions notre choix technique pour l'implantation de telles applications en parlant de la nécessité de coupler le paradigme agent avec celui des Web services.

Dans le deuxième chapitre, nous présentons les systèmes multi agents en mentionnant leurs critères et forces permettant de contribuer dans le domaine de l'e-business coopératif. Ce chapitre définit aussi les problèmes et les faiblesses de ce paradigme et qui peuvent être résolus par les Web services.

Le troisième chapitre est consacré à la présentation de la technologie des Web services. Il décrit les caractéristiques, les techniques et les moyens pour manipuler un tel concept. De façon réciproque, nous définissons aussi les faiblesses de ce domaine qui peuvent être résolues par le paradigme agent.

Le chapitre quatre décrit conceptuellement notre modèle proposé. Nous parlerons au premier lieu de quelques techniques concernant le couplage et l'exploitation des agents avec les Web services. Puis nous passerons à la présentation de notre solution avec ses deux phases principales : la définition du noyau agent et celle de l'intégration des Web services.

Le dernier chapitre est consacré pour la présentation des aspects techniques intervenant dans la mise en œuvre de telles applications avec une étude de cas avec laquelle nous apportons plus d'explication sur notre modèle. L'implantation est bien présente dans ce chapitre avec l'exploitation de l'environnement DotNet avec toute sa panoplie (C #, ASP.Net, etc.).

Enfin, nous terminerons cette thèse par un bilan de ce travail en insistant sur quelques points à développer dans une proche perspective.

CHAPITRE



E-business & Coopération: Les Nouvelles Tendances de l'Entreprise

1. INTRODUCTION

Les nouvelles technologies de l'information et de la communication (NTIC), dont Internet est la plus éclatante illustration, semblent avoir un tel impact non seulement sur le rythme de la croissance économique mais aussi sur le contenu des activités économiques, que ce soit les activités de production, celles de consommation ou encore celles de financement.

Cette progression a bien bouleversé les notions classiques de l'économie et celles du management traditionnel. Les nouvelles technologies de l'information et de la communication ont fait émerger un nouveau territoire immatériel, celui de l'échange et du partage de l'information et de la connaissance.

Comme l'entreprise est l'élément le plus touché par cette évolution, elle doit proposer, faire émerger, modifier et adopter des nouvelles stratégies de management et notamment elle doit informatiser son ancien système d'information. L'adoption du e-business tout en engageant dans des alliances de coopération avec les autres éléments de l'environnement économique, semble plus que nécessaire pour que l'entreprise puisse assurer sa continuité dans un monde de plus en plus concurrentiel. Afin de bien clarifier cette nouvelle situation économique, les nouveaux modes de travail ainsi que les défis soulevés par l'entreprise, nous présentons dans ce chapitre les notions de la nouvelle économie ainsi que les changements, les problèmes et les besoins que l'entreprise doit satisfaire.

2. NOUVELLE ECONOMIE & IMPACTS DES TIC

La plupart des définitions données au terme « Nouvelle économie » se convergent dans l'idée qui désigne que cet aspect est le résultat des changements induits par l'utilisation des nouvelles technologies, et en particulier de l'Internet dans la sphère économique. Quand on évoque la nouvelle économie, on pense évidemment globalement aux TIC mais peut-être plus particulièrement à Internet, c'est-à-dire à l'apparition (récente) et à la généralisation (rapide) d'un support universel de communication de données, qu'il s'agisse de textes, de sons ou d'images. Autrement dit, c'est l'application et l'adaptation des règles de l'économie classique aux échanges autour des technologies de l'information. De ce fait, nous pouvons constater que la nouvelle économie regroupe les secteurs des TIC qui sont classés en trois groupes (Christian, 2007) :

- *l'informatique* (entreprises produisant les matériels et les services connexes : Exploitation, maintenance et commerce) ;
- *l'électronique* (fabrication des composants et de certains appareils) ;
- et *les télécommunications* (activités de services et fabrication d'appareils)

Cet éclatement technologique a bien poussé les gestionnaires et les décideurs économiques de s'orienter de plus en plus vers les outils et les disciplines informatiques proposées dans ce contexte. Cet orientation peut être témoignée par l'adoption des Travaux Coopératifs Assistés par Ordinateur (TCAO), des "Groupwares"* , des "Systèmes de Décision Distribuées", des "Collecticiels" et pour la pointe la plus avancée des "Mondes Virtuels", des "Cyberspaces"** , des "Distributed cyberspaces", des "Collaborative Engines for Multiparticipant Cyberspaces", etc. (Saul, 1991).

Par conséquent, de nouvelles notions se manifestent au niveau structurel de l'économie moderne, telles que (Nouvelle, 2008).

- *Economie de l'immatériel*. Les diverses informations : données, textes, sons, images, programmes peuvent être distribués par le réseau, réduisant le coût de production à un coût de recherche.

* Techniques et méthodes d'informatique et de télécommunications destinées au travail de groupe à distance.

** Désigne le monde virtuel, dans lequel on se plonge lorsqu'on touche à un réseau informatique et surtout, Internet, le réseau des réseaux.

- *Dynamicité des prix.* L'achat aux enchères se développe sur le net dans tous les domaines : billets d'avion, chambres d'hôtels, matériel, laissant au consommateur le soin de fixer le prix du marché.
- *Une réduction des stocks.* Internet permet de relier les intermédiaires à leurs distributeurs, la production est lancée en temps réel à la demande du client supprimant les stocks.

Ces nouvelles notions citées ci-dessus, et grâce aux NTIC donnent lieu aux nouveaux modes de travail entre les éléments constituant l'entreprise et même entre l'entreprise et son environnement externe.

3. NOUVELLE ECONOMIE & DÉFIS DE L'ENTREPRISE

D'après ce que nous avons cité précédemment, la technologie vient alors bouleverser le processus économique : acheter, vendre, produire et distribuer. A tout moment, un nouveau concurrent peut bouleverser la chaîne de valeur en se positionnant en tant que nouvel intermédiaire. On passe d'un marketing de masse à un marketing personnalisé. La concurrence devient féroce, à la fois globale et locale. Chaque entreprise peut s'adresser au monde mais en s'adaptant localement à chaque marché. On peut maintenant partager sa passion avec des passionnés du monde entier et intégrer une communauté. Le rapprochement avec des partenaires et clients passe par une communication active dans les news-groups. Grâce à la messagerie électronique et aux nouvelles pratiques professionnelles du net, les accords, partenariats et contrats peuvent se conclure à travers le monde en une seule nuit. Pour résumer ces principaux changements, nous représentons ce tableau synthétique :

| | Ancienne économie | Nouvelle économie |
|---------------------------------------|-----------------------------|----------------------------|
| Caractéristiques économiques : | | |
| Marchés | Stables | Dynamiques |
| Compétition | National | Global et local |
| Organisation | Hiérarchisé, bureaucratique | En réseau |
| Industrie : | | |
| Organisation de la production | Production de masse | Production flexible |
| Facteurs de production | Capital/Travail | Innovation/Connaissance |
| Facteurs de compétitivité | Mécanisation | Numérisation |
| Sources d'avantages | Baisses des coûts et | Innovation, Qualité, Coût, |

| | | |
|-----------------------------|------------------------------|--|
| comparatifs | économies d'échelle | |
| Relations entre entreprises | Solitaire | Alliances et collaboration |
| Masse salariale : | | |
| Buts politiques | Plein-emploi | Adaptativité et plus hauts revenus |
| Compétences | Compétences spécifiques | Compétences variées et transdisciplinaires |
| Education requise | Une compétence ou un diplôme | Un apprentissage à vie |
| Management du personnel | Adversité | Collaboration |
| Nature de l'emploi | Stable | Risque et opportunité |
| Gouvernement : | | |
| Relations aux affaires | Régulariser | Encourager la croissance |
| Régulation | Commande et contrôle | Outils de mesure du marché, flexibilité |

Tableau I.1. Nouvelle économie & changements stratégiques (Nouvelle, 2008)

Ainsi, la nouvelle économie peut être une rupture et une menace, mais elle peut être aussi une continuité et une opportunité. Cela dépend principalement de la volonté de l'entreprise d'accompagner le changement. De ce fait l'entreprise doit soulever un minimum de défis afin de s'imposer et d'assurer sa continuité dans ce nouveau contexte économique. Les petites et moyennes entreprises sont notamment confrontées à de nombreux changements qui requièrent des réponses innovatrices (Pigneur, 1997) :

- la globalisation des marchés,
- l'émergence d'une économie dite de l'information, et
- l'évolution d'une économie de production vers une économie davantage orientée client.

Pour survivre et prospérer dans cet environnement, les entreprises doivent être efficaces, innovatrices et concurrentielles. Elles doivent être capables de répondre au bon moment, de se concentrer sur la qualité, et de réaliser une "personnalisation de masse" (Pine & al, 1995). Mais pour arriver à cela aujourd'hui, ils doivent aussi être capables de tenir compte des NTIC afin de mieux s'intégrer dans leur environnement, établir des relations plus coopératives avec les autres organisations et être davantage concurrentiels sur les marchés internationaux.

Il est bien clair, que le changement immédiat que l'entreprise doit l'effectuer afin d'affronter ces nouveaux défis est d'adapter et d'aligner sa stratégie, sa structure

organisationnelle et son utilisation des NTIC de façon à rester concurrentiel, survivre ou croître (voir figure I.1).

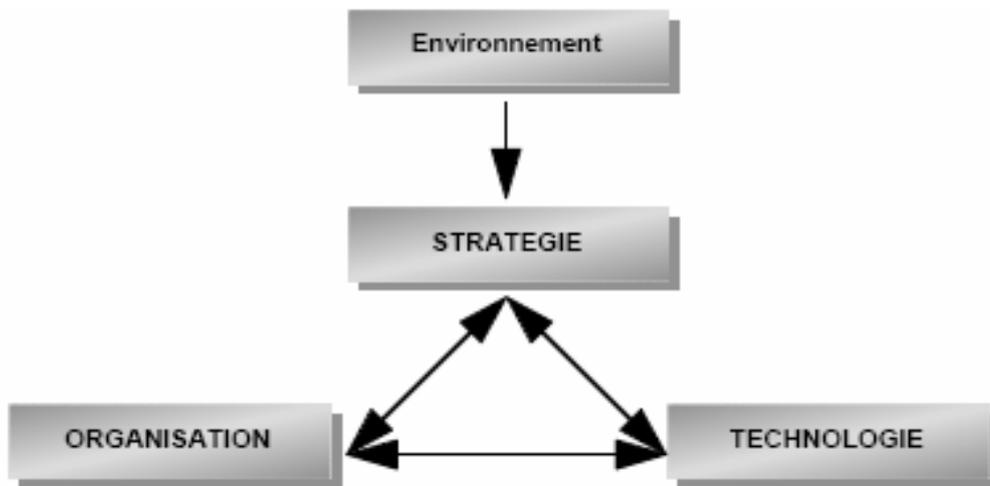


Figure I.1. Aligner stratégie, organisation et technologie (Pigneur, 1997)

Pour assurer cette adaptation avec succès, l'entreprise peut avoir besoin d'être ré-construit autour des concepts « stratégie, organisation et technologie ». La reconstruction de l'entreprise est définie selon Vernadat (Vernadat, 1996) comme « l'art de comprendre, définir, spécifier, analyser et mettre en oeuvre les processus business pour tout le cycle de vie de l'entreprise, afin que cette dernière peut accomplir ses objectifs, être plus efficace et plus compétitive dans son marché ». Selon Chandler, économiste et historien, la stratégie consiste à « déterminer les objectifs et les buts à long terme d'une organisation puis à choisir les modes d'action et d'allocations des ressources qui permettront d'atteindre ces mêmes buts et objectifs » (Chandler, 1988). La stratégie d'entreprise va donc, dans un premier temps, se référer à l'ensemble des décisions qui ont une incidence sur le devenir de l'entreprise à long terme. Dans un deuxième temps, elle se rapportera à la mobilisation et à l'emploi des ressources nécessaires. D'après cette définition, nous pouvons constater que la stratégie est le point de départ de toutes les autres activités de l'entreprise. Nous constatons que beaucoup d'entreprises ont commencé à consacrer moins de temps et d'effort dans la préparation des plans stratégiques à long terme. La conclusion était qu'une stratégie à long terme rend l'entreprise moins flexible, et comme le monde changeait rapidement, une entreprise ne devrait pas avoir vraiment une stratégie (Hammonds, 2001). En parallèle, il y avait aussi l'idée que l'évolution de la technologie devrait être adoptée dans l'entreprise sans examiner les raisons derrière son utilisation. Autrement dit, la technologie doit

conduire la stratégie de l'entreprise au lieu d'agir sur elle. Parler d'une stratégie d'entreprise et des changements prévisibles, ne doit pas dégénérer cette stratégie de ces objectifs classiques. Selon Porter, les entreprises adoptent essentiellement trois sortes d'objectifs (Porter & al, 1985) :

- réduction des coûts,
- diversification de leurs canaux de distribution et prospection de nouveaux marchés, et
- amélioration du service à leurs clients et de la qualité de leur offre.

Afin de réaliser et contrôler ces objectifs, au-dessous de la conduite des NTIC. Nous distinguons trois formes de changements que les entreprises essaient d'adopter pour:

- adopter un mode de travail en équipe, en aplatissant leurs hiérarchies et en améliorant leurs processus de décision et de gestion (Lipnack & al, 1993),
- construire ou s'intégrer dans des réseaux d'entreprises en adoptant de nouveaux modèles en liaison avec leurs partenaires commerciaux et en se transformant en entreprises étendues (Konsynski, 1993),
- établir de nouvelles formes de relations avec leurs clients en adoptant de nouveaux canaux de distribution, des processus de "personnalisation de masse" et des structures d'intermédiation renouvelées (Pine & al, 1995).

D'après cette analyse, nous pouvons déduire que la technologie occupe un champ de plus en plus large dans les considérations des gestionnaires. Dans le futur proche, le Web sera partout dans le monde. D'ici à quelques années les chiffres du business électronique seront en valeur plus qu'un trillion de dollars américains et les utilisateurs du Web seront plus de 600 millions. Ceci donne des occasions précédemment inconnues aux différentes organisations et entreprises pour réussir et survivre. L'orientation vers le business électronique est plus que nécessaire dans ce contexte économique, les leaders de l'industrie ont changé leurs stratégies vers l'e-business, et d'autres, tôt ou tard, plus ou moins, devront rejoindre la grande famille du e-business. De plus, et pour répondre à un environnement économique et concurrentiel de plus en plus complexe, à l'internationalisation et la globalisation des marchés, la technologie n'est pas suffisante dans ce stade, les entreprises doivent opter de plus en plus pour une multiplication des alliances et des partenariats en développant de nouvelles formes de fonctionnement. En bref, la nouvelle stratégie qui

assure le succès et la continuité de l'entreprise est celle qui couple entre la technologie ou le business électronique dans le sens large et la coopération ou la création des alliances avec d'autres entreprises et organisations. Nous présentons dans ce qui suit les deux domaines concernant le « E-business » et « la « coopération » afin de démontrer la force de l'existence de ces deux concepts à la fois dans la même entreprise.

4. E-BUSINESS

La notion d'e-business recouvre les différentes applications possibles de l'informatique faisant appel aux NTIC pour traiter de façon performante les relations de communication d'information d'une entreprise telle qu'une PME (Petites et Moyennes Entreprises) avec des organisations externes ou des particuliers.

L'e-business constitue non seulement l'ensemble des équipements informatiques et des logiciels reliant entre elles des entreprises ou une entreprise et un consommateur dans le but de faire des affaires mais aussi les processus d'affaires, de soutien, de formation, de financement, etc. Néanmoins, certains éléments de cette définition peuvent être dégagés et il est important de préciser dès à présent que le e-business représente bien plus que le simple fait de vendre des produits à des clients en ligne via un site web; le commerce électronique n'est, en fait, qu'un aspect de l'e-business. Les technologies utilisées sont principalement celles de l'Internet.

L'e-business est une transformation numérique des processus des affaires classiques qui aboutissent à un effet profond sur des procédures de gestion existantes (Patel & al, 2000). Le schéma ci-dessous illustre ces multiples relations de communication d'information d'une entreprise avec l'extérieur dans le cadre de l'e-business:

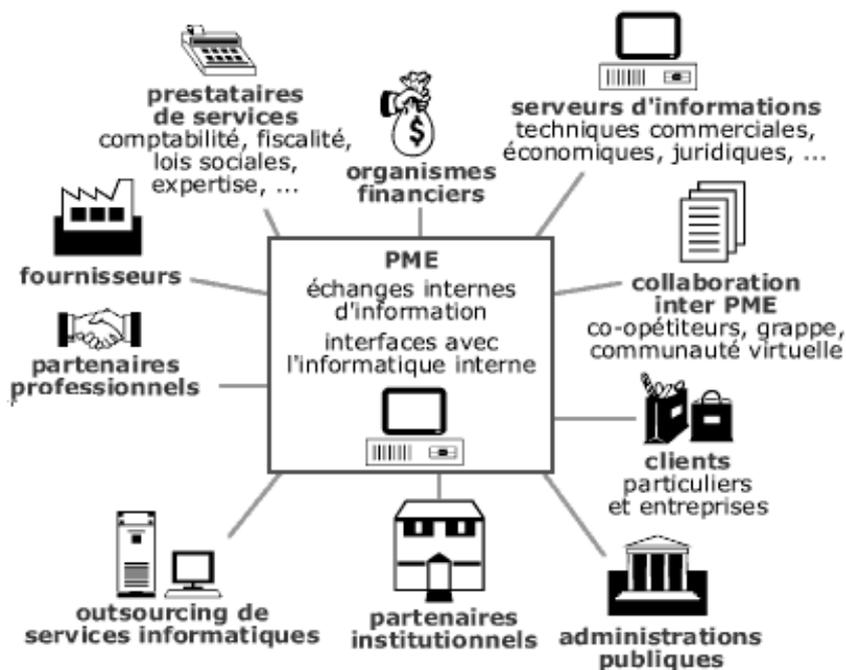


Figure I.2. Les multiples relations d'une entreprise avec l'extérieur (AWT, 2007)

Les applications de l'e-business ne se limitent pas à l'échange externe d'informations. Elles doivent permettre une articulation étroite (intégration):

- avec les processus internes à l'entreprise (production, stocks, vente, comptabilité, etc.);
- avec ceux des clients, fournisseurs ou partenaires.

4.1. TYPOLOGIE DES APPLICATIONS E-BUSINESS

D'après le schéma ci-dessus nous remarquons la diversité de relations entre l'entreprise et son environnement externe, ce qui implique l'existence de multiples applications basées sur le business électronique. Cette multiplicité d'applications est exprimée par la matrice de Weilhhammer (Weilhhammer, 2000) sous la forme suivante :

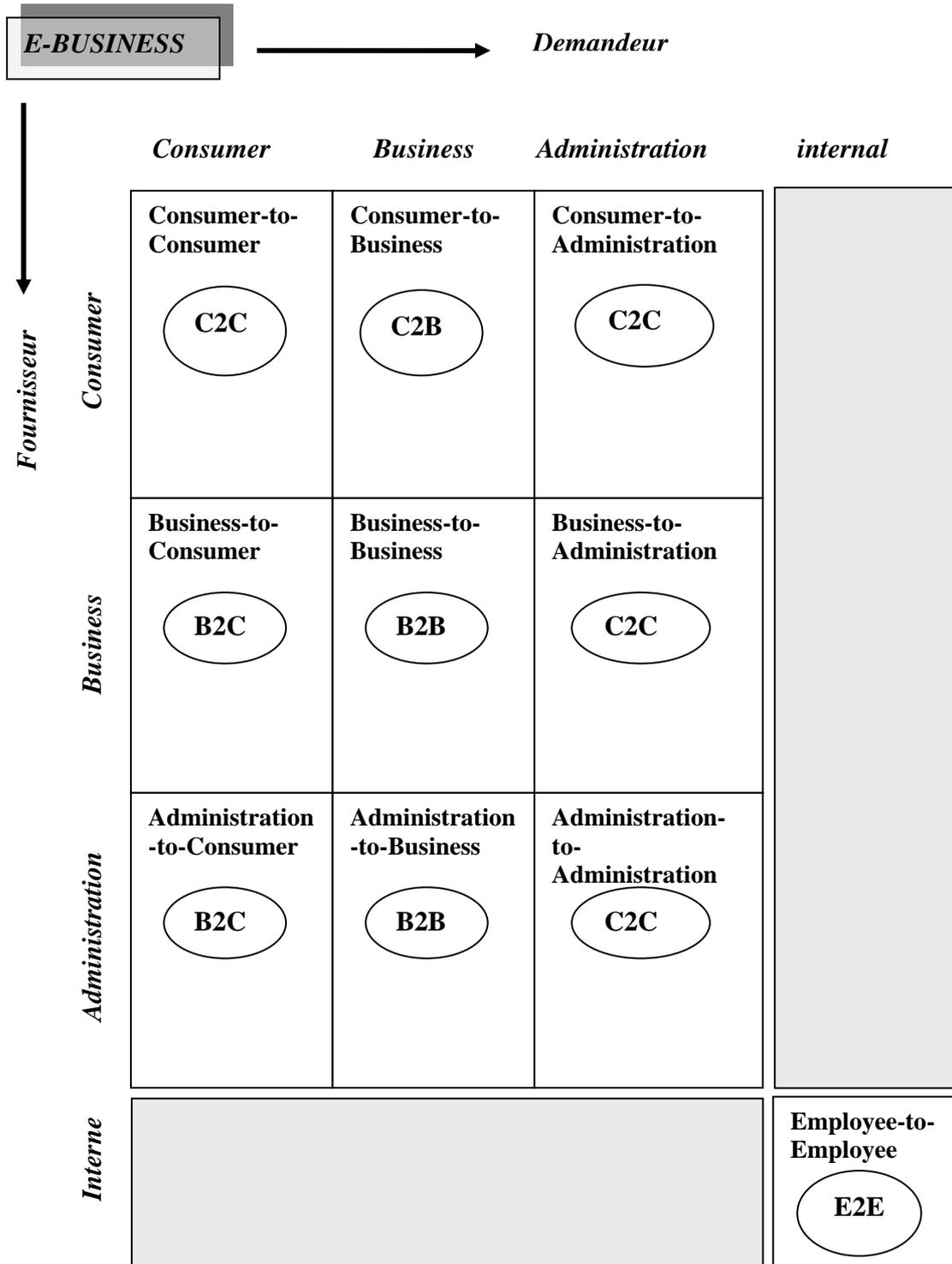


Figure I.3. Matrice du business électronique (Weilnhammer, 2000)

Nous remarquons que cette multiplicité vient de la diversité des partenaires qui peuvent s'engager dans l'e-business est qui sont essentiellement trois : le consommateur, l'entreprise est l'administration, sans oublier les relations internes qui peuvent être instaurer entre les employés de la même organisation (Employee-to-Employee). Dans la

vie économie pratique, et d'après Paul Jérrouville (Jérrouville, 2001), les valeurs ajoutées de l'e-business prennent leur importance dans trois secteurs essentiels qui représentent les secteurs e-business les plus diffusés :

4.1.1. Le business-to-business (B2B)

Il s'agit, d'une manière générale, des relations entre entreprises. C'est une forme d'e-business qui se rapporte aux technologies de chaîne d'approvisionnement (supply chain). C'est la forme de technologie e-business la plus répandue et la plus réussie de nos jours. Il se dit d'une activité professionnelle (échanges ou transactions commerciales) effectuée d'entreprise à entreprise. Le commerce B to B désigne par exemple les échanges entre un fournisseur de produits intermédiaires et un producteur distribuant au consommateur final.

4.1.2. Le business-to-consumer (B2C)

Il recouvre les activités économiques (business) à destination du grand public et du consommateur final (consumer). C'est une forme d'e-business qui met en jeu des détaillants dans la vente ou l'achat de services. La différence principale entre cette forme et le B2B consiste dans le fait que dans le B2B les parties sont des associées connues et de confiance quant au B2C les parties sont généralement inconnues.

4.1.3. L'intermédiation

C'est un cas particulier de politique e-business qui consiste pour une entreprise à devenir un prestataire de services à valeur ajoutée sur internet. L'entreprise qui se positionne de la sorte devient un «intermédiaire de marché électronique». Ce type de politique a sans conteste de l'avenir puisque les entreprises comme les particuliers éprouvent de plus en plus de difficultés à trouver de l'information pertinente au sein d'un réseau de plus en plus foisonnant.

Dans plusieurs situations, la confusion entre l'e-business et le e-commerce est fortement diffusée. Ces deux concepts sont similaires. Mais, en réalité, l'e-business est un terme plus générique que l'e-commerce parce qu'il attribue non seulement à l'achat et la vente, mais aussi l'entretien de clients et la collaboration avec des associés. Le commerce électronique ne représente qu'un secteur particulier de l'e-business qui concerne notamment le B2C (Business to consumer). Le e-commerce consiste à vendre un produit ou un service à travers un catalogue électronique. Il concerne l'utilisation de supports

électroniques pour tout ou partie des relations commerciales entre une entreprise et les particuliers:

- publicité,
- présentation de catalogue,
- commande en ligne,
- paiement électronique, et
- distribution.

4.2. AVANTAGES ET VALEURS AJOUTÉES DU E-BUSINESS

L'e-business a pour objectif l'amélioration de la rentabilité des entreprises en augmentant le chiffre d'affaires de celles-ci. D'où la définition de l'e-business qui consiste en une intégration au sein de l'entreprise d'outils basés sur les technologies de l'information et de la communication pour en améliorer le fonctionnement afin d'aboutir aux valeurs ajoutées pour elle-même, pour ses clients et pour ses partenaires. Grâce à ce concept, nous pouvons dégager plusieurs points bénéfiques de l'e-business dont l'essentiel :

- Les opérations de l'entreprise deviennent plus concurrentielles en augmentant la productivité tout en décroissant les coûts (Mohini & al, 2004).
- L'entreprise peut instaurer une diversité de canaux de communication avec ses clients et ses partenaires 24 heures sur 24 et sept jours sur sept en négligeant toute les distances géographiques (Singh, 2000)
- L'e-business assure une meilleure image pour l'entreprise, exige moins de surface au sol, mène à une grande exactitude des données et apporte plus d'occasions aux responsables (Whinston & al, 1998).
- L'entreprise peut réduire les frais de la publicité et de la commercialisation. Le mode publicitaire -en ligne- est non seulement efficace, mais il est souvent moins cher que la publicité traditionnelle.
- La chaîne logistique de l'entreprise s'améliore. Grâce aux modes de travail et de communication assurés par l'e-business, le cycle (commande, livraison, production et commercialisation) peut se dérouler dans les meilleures conditions en assurant les meilleurs temps de réponse.

Généralement, nous pouvons résumer et classer les avantages du e-business selon les principaux bénéficiaires de ce mode de travail et qui sont, l'entreprise elle-même, ses clients, ses partenaires sans oublier son personnel (AWT, 2007)

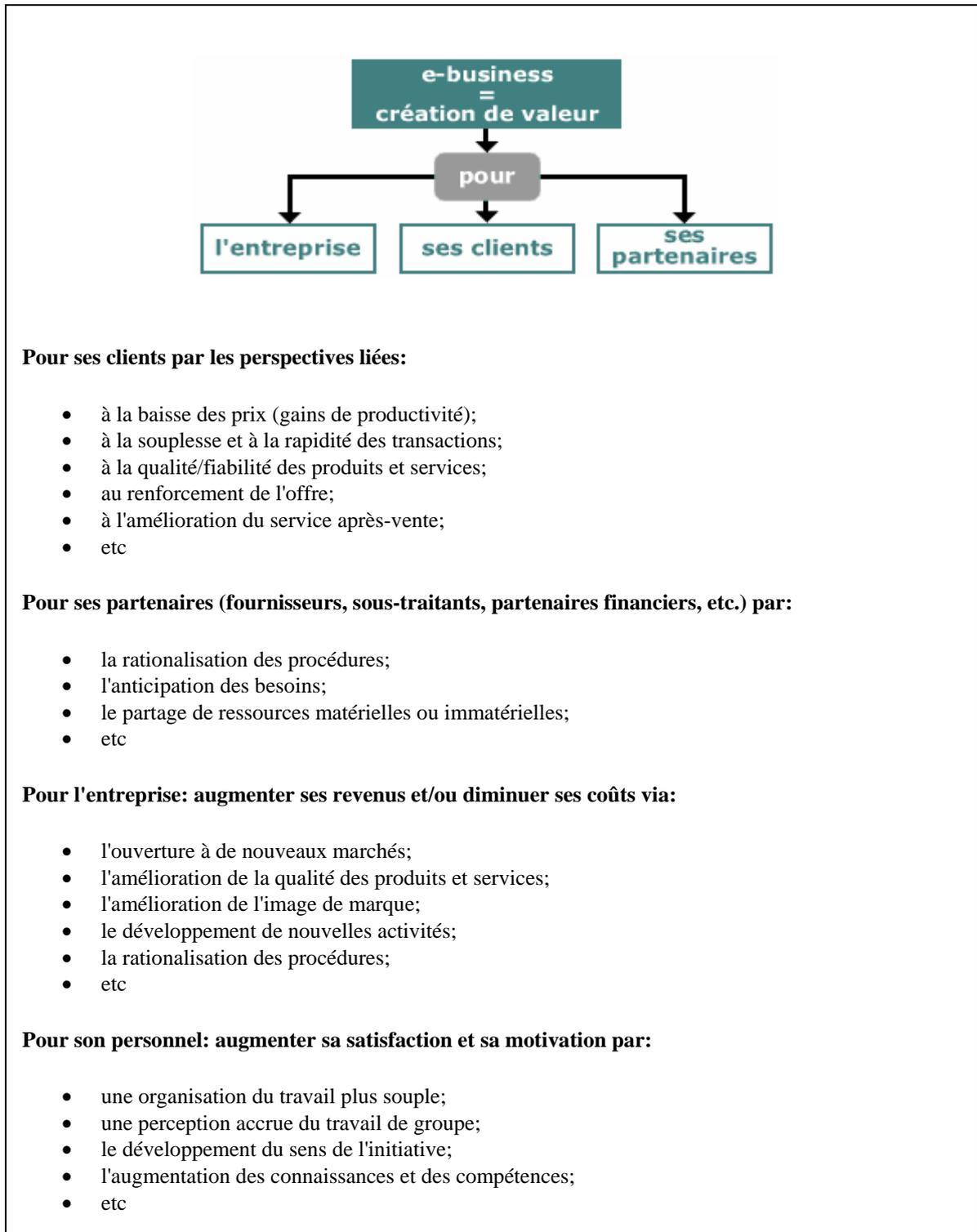


Figure I.4. Avantages du e-business (AWT, 2007)

4.3. DÉFIS & PROBLÈMES DU E-BUSINESS

Bien que l'e-business rapporte des solutions miraculeuses pour les diverses parties du monde économique, il reste toujours comme une notion neuve dont le contenu n'est pas encore bien maîtrisée. Néanmoins, le fonctionnement de l'e-business engendre des nouvelles relations et situations entraînant, par là même, de nombreuses questions de différentes natures. Petras (Petras & Tadas, 2004) et Kadlec (Kadlec & al 2002) ont cité de nombreux problèmes que nous pouvons les classifier selon leurs aspects humains, manageriel, juridique et technique et dont l'essentiel sont :

4.3.1. Problèmes humains

- Les managers ignorent, totalement ou en partie, les opportunités et l'importance offerte par les nouvelles technologies de l'information et de la communication. Par conséquent les gestionnaires des entreprises ne soutiennent pas ces types de projets.
- L'e-business est souvent confondu avec l'idée d'avoir un site Web.
- Il y a un manque de professionnels qui ont une vue d'ensemble générale sur les projets e-business, et sur la gestion de point de vue technologique.

4.3.2. Problèmes de gestion

- les processus d'organisation ne sont pas réorientés quand la numérisation arrive, afin d'adapter les entreprises correctement.
- Aucune méthodologie n'existe pour le développement des projets e-business.
- Les entreprises ont des modèles de travail basés beaucoup plus sur des relations et des liens personnels, et la peur du changement est considérablement élevée. Pour eux : un échec pourrait être le dernier.

4.3.3. Problèmes juridiques

- les problèmes relatifs à la conclusion et à la validité des contrats conclus sous forme électronique
- les difficultés liées à la taxation du commerce électronique
- les problèmes de signature électronique ou de fourniture de services en ligne
- le marketing,
- la publicité,

- le paiement,
- les autorisations ou contraintes régissant le commerce transfrontalier (notamment l'obligation d'établissement ou d'autorisation préalable à la fourniture locale de biens ou de services).

4.3.4. Problèmes techniques

- La technologie de l'information n'est qu'une technologie pour le partage efficace, le traitement et la gestion de l'information, et actuellement elle ne peut devenir ni un vrai canal de vente ni un vrai support pour les principes de gestion.
- Le manque d'outils techniques permettant de donner des réponses complètes sur les préoccupations du e-business. Le peu d'outils qui existe ne donne que de réponses partielles orientées beaucoup plus vers les problèmes de communication et d'échanges d'informations.
- les applications e-business souffrent d'un manque de sécurité et restent toujours vulnérables.
- Les solutions e-business restent toujours coûteuses notamment sur le niveau applicatif (logiciels)

Un petit survol d'analyse sur les défis et les problèmes cités précédemment nous permet de conclure que sont rares les entreprises ayant les capacités et les ressources pour surmonter tous ces problèmes. L'entreprise doit donc faire appel à d'autres entreprises et sous-traitants en formant une alliance pour réunir leurs compétences et travailler ensemble afin de réaliser leurs objectifs. Comme nous nous sommes limités sur les problèmes de l'aspect technique, nous évoquons dans ce contexte le concept d'e-business coopératif. En effet, une entreprise peut faire appel à une autre entreprise pour n'importe quelle tâche, la production, la conception, le marketing ou même pour l'échange des expertises. Pour cela, elle doit prendre en compte aussi l'organisation et la coordination entre les différentes parties de sa propre structure. On ne peut pas parler de coopération avec l'environnement extérieur sans parler de coopération avec les constituants de l'entreprise elle-même. Les projets qui sont réalisés par plusieurs entreprises sont des projets qui font appel à des domaines complètement différents et qui sont donc gérés par plusieurs entreprises à niveau

d'investissement équivalent. Nous allons fournir dans ce qui suit les caractéristiques de ce nouveau mode de travail basé sur la coopération intra et interentreprises.

5. COOPÉRATION

Il n'existe pas réellement de consensus sur une définition précise de la coopération. Plusieurs définitions souffrent d'une certaine proximité avec d'autres notions, comme celle de « collaboration » par exemple, ce qui explique une certaine confusion qui règne dans les recherches concernant les formes de coopération dans l'entreprise (Boutigny, 2004). Pour Smith (Smith & al, 1995), la coopération est un processus par lequel des individus, des groupes ou des organisations entrent en relations et travaillent ensemble dans un but commun. Cette définition reste générale et loin de la distinction entre le concept de coopération et celui de la collaboration. Alors pour Schmidt (Schmidt, 1994), le travail coopératif « émerge quand plusieurs acteurs engagés dans la réalisation d'une tâche commune, sont mutuellement dépendants dans leurs productions individuelles pour être en mesure d'effectuer la tâche qui leur a été confiée ». Cette définition est encore restrictive et elle ne considère exclusivement que le travail porté sur un objectif commun qui correspond au but principal du collectif.

Pour nous, nous estimons que la définition la plus adéquate et loin de toute confusion et celle de Cerisier (Cerisier 1999). Il définit la coopération comme une organisation collective du travail dans laquelle la tâche à satisfaire est fragmentée en sous tâches, chacune de ces sous tâches est ensuite affectée à un acteur, soit selon une distribution parfaitement horizontale dans laquelle tâches et acteurs sont équivalents, soit selon une logique d'attribution en fonction des compétences particulières de chacun. La collaboration s'entend en revanche par « une situation de travail collectif dans laquelle tâche et but sont communs. Tous les acteurs travaillent sur les mêmes points. ».

La coopération se caractérise par son aspect pluridisciplinaire puisqu'elle implique non seulement des disciplines techniques telles que l'étude des interactions homme/machine, les réseaux de télécommunication, les théories de l'intelligence artificielle, l'étude des processus industriels, mais aussi des disciplines des sciences humaines telles que la sociologie des organisations, la sociologie scientifique, la psychologie cognitive, la psychologie sociale, l'ergonomie, l'anthropologie, les théories de

la décision, les sciences cognitives, l'ethnologie, la linguistique, et l'éducation (Karsenty 1994).

5.1. TYPOLOGIE DE COOPERATION

Plusieurs études de recherche témoignent que la coopération entre les entreprises ou en terme général les alliances stratégiques entre les entreprises concernent les grandes entreprises, peuvent prendre plusieurs formes : additives ou complémentaires, entre entreprises concurrentes ou non (Hennart, 1988; Dussauge & Garrette, 1995). Les travaux sur les alliances d'entreprises montrent en revanche que ce type de firme évite, voire refuse les alliances entre concurrents, et favorise les alliances complémentaires plutôt qu'additives (Puthod, 1996 ; Bacus, 1997). De ce fait, plusieurs dimensions sont utilisées dans la littérature pour caractériser les différentes situations de coopération rencontrées. Celles-ci sont soit fondées sur les acteurs qui coopèrent, soit sur la coopération elle-même (Saubesty, 2003). Le tableau ci-dessous montre ces situations avec leurs dimensions appropriées.

| Dimensions | Travaux |
|--|---|
| Les compétences des agents coopérants | (Soubie & al, 1994) font la distinction entre les situations dans lesquelles les compétences des agents sont comparables et celles dans lesquelles les compétences des agents sont complémentaires. |
| La forme formelle ou informelle de la coopération | La coopération informelle implique des arrangements dans lesquels des normes de comportements (Ouchi, 1980) plutôt que des obligations contractuelles déterminent les contributions des parties. |
| La dépendance entre les parties (hiérarchique / non hiérarchique) | Les groupes ou les individus qui coopèrent peuvent être liés verticalement (un supérieur et un subordonné) ou horizontalement (des individus engagés dans une tâche commune). Nous nous intéressons à la coopération horizontale. Cependant, cette dichotomie semble abusive, les acteurs étant intégrés dans des systèmes hiérarchiques. |
| L'objet de la coopération communautaire versus complémentaire | (Dameron, 2000) distingue la coopération complémentaire, où les acteurs sont prêts à travailler ensembles pour bénéficier de ressources complémentaires, de la coopération communautaire où les acteurs sont prêts à travailler ensembles pour appartenir au groupe. |

Tableau 1.2 : Situations du travail coopérative (Saubesty, 2003)

Pour que notre travail soit bénéfique, nous considérons toutes les situations où la coopération peut apporter un plus. Nous nous intéressons au statut technique permettant à

l'entreprise de nouer des alliances avec d'autres entreprises et partenaires sans nécessairement en augmenter la taille et engager des coûts additionnels.

5.2. OBJECTIFS DE LA COOPERATION INTERENTREPRISES

L'objectif premier de la coopération est de répondre à un besoin des entreprises. Les projets doivent correspondre à une perspective de bénéfice (financier, humain, matériel...), de limitation de risques ou de baisse des coûts. Le porteur du projet de coopération doit constamment mettre en évidence l'intérêt commun des entrepreneurs à coopérer, pour s'assurer du succès de la démarche. Les possibilités d'actions de coopération sont multiples et dépendent des besoins des entreprises, d'après les rapports de l'organisme d'Entreprises Territoires et Développement (ETD) (ETD, 2008) les bénéfices de la coopération au dessous des alliances sont multiples. Nous citons certains d'entre eux :

- Partage de l'information : étude de marché commune, recherche de perspectives et d'axes de développement pour une filière, veille économique,
- Partage de compétence et de moyens : partage de compétences, de savoir faire, de personnel (groupements d'employeurs), investissement en commun, groupement d'achat, recours groupée à un prestataire, gestion collective de domaines spécifiques...
- Actions de formation : définition en commun des besoins de qualification, partenariats avec les systèmes éducatifs, actions sur l'adéquation formation / métier...
- Actions de recherche et d'innovation : intelligence économique et territoriale, recherche et développement sur thématiques communes, partenariats avec acteurs locaux (centres de recherche, universités, laboratoires), actions de filière (ANVAR, DRIRE, DRRT, ...).
- Actions commerciales : promotion organisée en commun (site Internet, salons, publicité, labels), marketing du territoire, travail en commun à l'export, création d'une structure commerciale, partage d'un commercial...
- Collaboration avec les institutionnels : élaboration d'une stratégie globale de développement local, partagée par les acteurs publics et privés du territoire, élaboration d'un véritable projet commun entreprises / territoire...
- Partage des risques et des coûts.

- Concentration de l'entreprise sur ce qu'elle sait faire au mieux...

Généralement, et d'un point de vue fonctionnel, une alliance se traduit par une plus grande collaboration et une meilleure gestion des connaissances. Elle permet également de répondre aux problématiques de gestion de cycle produit, en agrégeant de manière satisfaisante les données de conception, de gestion financière et logistique, les données relatives aux fournisseurs et aux clients. Elle vise également à réduire les cycles d'approvisionnement - places de marché privées par exemple - et de décision (JND, 2002).

5.3. E-BUSINESS COOPERATIF

Il est clair, que les alliances au dessous du e-business coopératif sont caractérisées par un mélange de coopération et de concurrence. Les membres d'une alliance restent autonomes et distribués, entièrement responsables de leurs budgets, de leur planification et de l'établissement de leurs programmes et leurs ressources. Cette situation a besoin d'un minimum d'exigences (Alloui & al, 2002):

- Optimisation de l'utilisation des ressources (en hommes et/ou en machines)
- Gestion de plusieurs activités concourantes et concurrentes en même temps.
- Supervision des activités pour pouvoir réagir aux événements imprévus et résoudre dynamiquement les éventuels problèmes.
- Capture et mise en disposition de toutes les informations nécessaires aux partenaires de la coopération.

De plus, les travaux de coordination et de communications sont multiples. Les partenaires doivent assurer l'échange des informations et des demandes en surveillant leurs exécutions. Par conséquent, l'infrastructure pour un tel mode de travail devrait permettre : (i) une autonomie dans les décisions et les tâches propres à chaque partenaire ; (ii) une coordination des activités concourantes et concurrentes avec les ressources consommées en collaboration entre eux (iii) la capacité de s'adapter dans un environnement dynamique et en cours d'évolution afin de manipuler des événements inattendus et à faire face à de nouvelles conditions telles que la participation d'un nouveau partenaire dans l'alliance ou le contrôle des conséquences du départ d'un membre.

Face à ces exigences techniques compliquées, les systèmes multi agent (SMA) (Weiss, 1999) apparaissent comme un paradigme apte à répondre sur les préoccupations

citées précédemment. La faculté d'un agent qui lui permette de supporter les ambitions des acteurs e-business vient à l'origine de ses aptitudes de coopération, de communication et d'autonomie. Mais dans un contexte tel que l'e-business, ce paradigme n'est pas suffisant avec l'ambition des gestionnaires et l'apparition du fameux concept « Web Services ». Ce dernier semble comme la meilleure réponse aux problèmes liés aux mécanismes et les standards du Web. Les Web services (Gillbert & al, 2003) apparaissent comme des modules qui peuvent être publiés, localisés, et invoqués à travers le Web avec les moindres exigences techniques. Notre objectif est de mêler la puissance de ces deux paradigmes pour créer un couplet Agent-Web Service pour le développement des applications e-business coopératives. Nous présentons ultérieurement ces deux paradigmes en mettant l'accent sur leurs propriétés, leurs standards, leurs problèmes et leurs points bénéfiques.

6. CONCLUSION

Nous avons présenté dans ce chapitre, la situation de la nouvelle économie, les notions et les concepts fortement liés à cette dernière, ainsi que les besoins et les changements nécessaires que doit effectuer l'entreprise. Nous avons constaté que l'adoption du business électronique coopératif est plus que nécessaire pour assurer la continuation et l'évolution de l'entreprise.

Il est très clair qu'une application e-business coopérative bien conçue doit prendre en considération plusieurs concepts et mécanismes compliqués. De ce fait, et face à ces difficultés, le paradigme agent couplé à celui du Web service peut supporter ces préoccupations techniques basées essentiellement sur la multiplicité d'interactions, de coopération, de coordination et d'échanges d'information.

Dans le chapitre suivant nous allons décrire les notions liées au paradigme agent en montrant ses aptitudes, ses mécanismes, ses propriétés ainsi que les travaux fournis dans le domaine du e-business coopératif.

CHAPITRE



Systemes Multi Agents & Business Electronique

1. INTRODUCTION

Le paradigme agent connaît dans ce dernier temps un succès incontestable. Il peut être appréhendé comme un nouveau paradigme de conception et de programmation, doté d'un aspect « communicationnel », hérité du génie logiciel et d'un aspect « mentaliste » ou « cognitif », hérité de l'intelligence artificielle (IA) (Shoham, 1993). Actuellement, ce paradigme est employé dans divers domaines d'application: l'assistance personnelle, l'automatisation des processus industriels, la prise de décisions, la simulation, le support de la mobilité, les cartes à puce intelligentes, les télécommunications....

Comme le Web constitue l'environnement d'exécution le plus visé par la plupart de ces applications, le paradigme agent trouve encore une nouvelle place où il, a bien démontré toutes ces performances. Le business électronique est l'un de ces domaines, où plusieurs travaux et propositions basées sur la notion d'agent ont été fournis ces dernières années.

Nous discuterons dans ce chapitre, les caractéristiques des systèmes multi agents leurs structures et les problèmes qu'ils visent à résoudre au sein du domaine e-business. Nous citerons également quelques typologies d'agent et quelques domaines d'application e-business où l'agent constitua la pierre angulaire de la modélisation de ces dernières.

2. DU HOLON A L'AGENT: Historique de modélisation dans l'entreprise

Comme tous les autres domaines, l'entreprise a bien évolué avec la progression de l'informatique. Cette discipline a donné beaucoup pour l'entreprise en particulier et pour les systèmes d'information en générale. La figure I.1 montre l'historique de l'apparition et de l'exploitation des technologies informatiques dans l'entreprise.

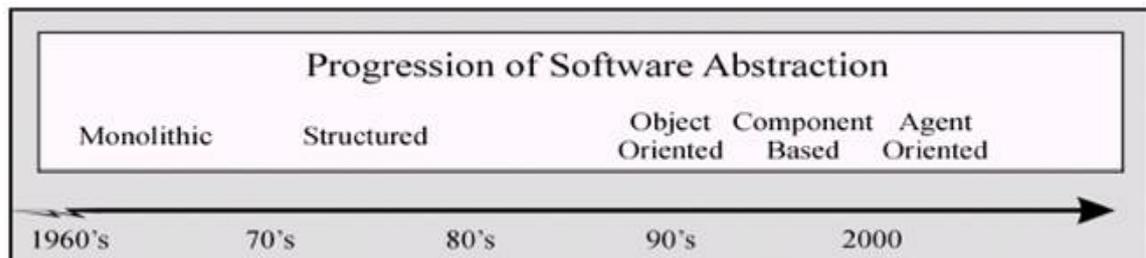


Figure II.1. Progression des solutions informatiques (Sansonet, 2005)

Beaucoup de paradigmes et de solutions ont été proposés pour faciliter les travaux et les interactions de l'entreprise. La première tentative sérieuse surmonte à l'année 1969, avec la proposition des systèmes holoniques. Ce concept a été proposé par Arthur Koestler dans un livre intitulé « *The Ghost in the Machine* » (Koestler, 1969). Cette proposition fait suite à une étude sur le fonctionnement des différentes sociétés non seulement humaines (communistes, capitalistes, dictatoriales, démocratiques) mais aussi animales et végétales (l'arbre ou la plante sont alors vus comme une société composée d'entités). C'est une approche systémique des organisations (Arsène, 2001). Chaque partie du système, appelée holon, est elle-même décomposable en holons. Un holon veut dire l'élément de base d'un système, ce mot combine la racine grecque holos qui veut dire « partie », avec le suffixe –on- qui veut dire « entier ». Le principe holonique veut que chaque partie soit :

- Stable pour pouvoir faire face à des perturbations;
- Autonome pour pouvoir agir d'elle-même afin de réaliser son objectif;
- Coopérante pour se lier à d'autres parties afin de réaliser l'objectif commun du système.

Ces systèmes font l'objet d'un courant de recherche actuellement dans le domaine de la robotique et des systèmes manufacturiers. On peut, en effet, concevoir une entreprise comme une entité composée de sous entités qui, idéalement, devraient être stables et avoir des degrés d'autonomie et de coopération équilibrés (Adam, 1999).

Puisque à l'époque où le concept holon est apparu, la préoccupation majeure des gestionnaires est de faire évoluer et améliorer leurs systèmes de production. Les développeurs ont bien orienté ce concept vers le stade d'automatisme et de production. Ils cherchent fortement d'aboutir aux systèmes de production agiles, dont le sens où l'entreprise de production agile sera capable de développer rapidement des produits entièrement nouveaux. Elle assimile facilement les avances technologiques et les problèmes de ses clients, modifie constamment ses produits pour incorporer de nouvelles optimisations. Les produits qu'elle développe sont conçus pour évoluer rapidement. Au lieu de devoir remplacer un système, les utilisateurs de ces produits peuvent rapidement changer leur configuration ou ajouter de nouveaux modules lorsque les besoins changent ou que de nouveaux produits apparaissent (Christensen & al, 1996). La figure suivante montre un exemple d'un système holonique consacré aux problèmes de production

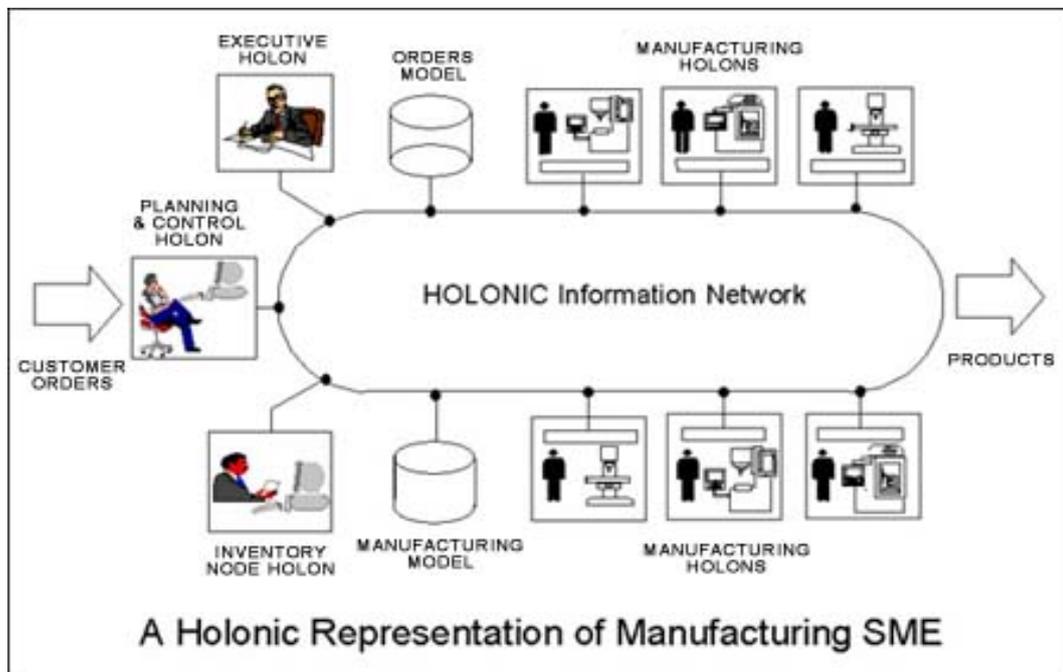


Figure II.2. Exemple de système de production holonique (AMST, 2008)

Actuellement, les systèmes holoniques constituent l'une des infrastructures nécessaires à la création de l'entreprise de production agile. Malgré, l'importance de ce concept très avancé par rapport à l'époque dans laquelle il a été proposé. Aucune exploitation n'est faite dans le domaine informationnel des entreprises. La spécialité qui l'a pris le holon dans les problèmes de production, a poussé l'entreprise à chercher d'autres solutions techniques comme support de leurs systèmes d'information.

A la fin des années 80, la notion d'objet est apparue comme un autre paradigme prometteur. Ce paradigme proche de la réalité a bien attiré l'investissement des gestionnaires. Un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. C'est une structure de données valuées traitées par ses propres méthodes et qui répond à un ensemble de messages à travers ses dernières. Cette structure de données définit son état tandis que l'ensemble des méthodes qu'il comprend décrit son comportement. Comme l'atmosphère du business électronique est fortement basée sur les interactions entre les entités informatiques où ces entités peuvent être distribuées. Cette dernière intègre de multiples concepts tels que : conflit, communication, collaboration, compétition, contrôle, coordination, négociation, etc. L'objet peut soulever les critères de cette atmosphère. Ces critères prennent une valeur plus ou moins importante selon la nature de l'application, l'environnement du travail et les techniques de mise en œuvre utilisées.

De la structure hiérarchique à la structure en réseau de certaines entreprises, l'informatique passe de l'organisation en "mainframe" vers la technique à objets distribués. Les systèmes actuels d'objets distribués comme le Middleware, les bases de données orientées objet, les applets et notamment le langage JAVA actuellement en vogue envahissent les architectures client/serveur des entreprises. Ces applications seront assemblées à partir d'objets préfabriqués ou standard. On devrait donc concevoir des applications permettant aux objets, quelle que soit leur origine, de s'interchanger. Devant l'enjeu commercial que représentent ces normes techniques, trois technologies s'affrontent aujourd'hui pour dominer le marché :

- **L'OMG (*Object Management Group*)**, consortium qui regroupe plus de 700 membres institutionnels (constructeurs, éditeurs de logiciels, etc.) et développe une spécification d'architecture distribuée appelée CORBA (OMG, 2009).
- **Microsoft** qui promet aussi son propre système d'objets distribués appelé OLE et ActiveX (Microsoft, 2009).
- **SUN** avec sa filiale javasoft qui renforce le langage JAVA par l'ajout d'une bibliothèque des objets JDK évoluant dans l'environnement d'objets distribués RMI/Java Beans (Sun, 2009).

Malgré ces apports "applicatifs" dans le domaine d'Internet, le paradigme objet a du mal à s'implanter dans les entreprises. Il faut y voir sans doute les limites de son

adéquation aux besoins réels des entreprises hormis celles qui utilisent le réseau Internet dans leur logique de métiers et c'est le cas du business électronique. La modélisation orientée objet définit l'objet comme un élément statique, non adaptable à son environnement en raison de sa méthodologie de réutilisation des composantes existantes pour des projets différenciés. D'où la limite de cette modélisation qui doit être enrichie à l'aide d'apports significatifs et approfondis de la théorie des systèmes. De plus, les applications e-business sont caractérisées par une dynamique intensive dans leurs processus que dans leurs contenus, ce qui exige des aptitudes liées essentiellement à l'intelligence, l'autonomie et la faculté de supporter de multiples sortes d'interaction. La réponse aux exigences citées précédemment vient avec le paradigme Agent (Ferber, 1997), le successeur actuel de l'objet. Nous allons présenter dans ce qui suit ce paradigme en focalisant sur ses critères que nous voyons utiles dans le développement des applications e-business coopératives.

3. SYSTEMES MULTI-AGENTS : Concepts & principes de base

Aujourd'hui, la plupart des applications e-business nécessitent de distribuer des tâches entre des " entités " autonomes qui interagissent continuellement entre eux afin d'atteindre leurs objectifs d'une manière optimale. Puisque les approches classiques soient en général monolithiques et leur concept d'intelligence soit centralisé, le développement des applications e-business actuelles s'oriente fortement vers le paradigme agent (Papazoglou, 2001).

3.1. AGENTS

Il n'existe pas encore un consensus sur la définition d'un agent. Elle recouvre plusieurs définitions, chacune renvoyant à un courant de recherche particulier dans le domaine de ce qui touche à la nébuleuse «agent». En plus de la relative jeunesse du domaine, une des raisons est que diverses communautés revendiquent ce terme avec des problématiques parfois au départ assez différentes (par exemple en ce qui concerne les agents mobiles) même si ces différentes problématiques sont complémentaires et conduites à se rencontrer à terme. Pour Weiss (Weiss, 1999), un agent est une "entité de calcul", comme un programme informatique ou un robot, qui peut être vue comme percevant et agissant de façon autonome sur son environnement. On peut parler d'autonomie parce que son comportement dépend au moins partiellement de son expérience. Ferber (Ferber, 1995)

ajoute à cette définition un ensemble de propriétés que nous trouvons très adéquates pour notre domaine d'application :

- Se trouve dans un système informatique, ouvert comprenant un ensemble d'applications, de réseaux, et de systèmes hétérogènes.
- Peut communiquer avec d'autres agents.
- Est mu par un ensemble d'objectifs propres (c'est en ce sens que l'on peut parler d'agent intentionnel).
- Possède des ressources propres.
- Ne dispose que d'une représentation partielle des autres agents.
- Possède des compétences (services) qu'elle peut offrir aux autres agents.
- A un comportement tendant à satisfaire ses objectifs, en tenant compte d'une part des ressources et des compétences dont elle dispose, et d'autre part de ses propres représentations et des communications qu'elle reçoit.

Cette définition met l'accent sur l'autonomie de décision qui résulte de l'indépendance avec laquelle un agent tente de satisfaire des objectifs qu'il s'est assigné, en utilisant au mieux les ressources et les compétences dont il dispose.

De ce fait, nous pouvons voir un système multi-agents comme un ensemble de processus informatiques se déroulant en même temps, donc de plusieurs agents vivant au même moment, partageant des ressources communes et communiquant entre eux (Weiss, 1999).

Au regard des caractéristiques d'agent citées ci-dessus, il apparaît que l'approche orientée agent dans le développement de logiciel consiste en une décomposition du problème en agents ayant des interactions, une autonomie, et un objectif spécifique à atteindre. Les concepts clés d'abstraction liés à un système orienté agent sont : agent, interaction, organisation. Bien qu'il existe une similarité superficielle entre objet et agent, la terminologie objet n'est pas adaptée aux systèmes agents (Arsène, 2001):

- les objets sont généralement passifs alors que les agents sont permanents actifs;
- les agents sont autonomes et responsables de leurs actions alors que les objets n'en sont toujours pas;
- on ne peut prévoir tous les comportements des agents dans les systèmes;

- l'approche orientée objet ne fournit pas un ensemble adéquat de concepts et de mécanismes pour modéliser les systèmes complexes dans lesquels les rapports évoluent dynamiquement;
- certains chercheurs définissent un agent comme un objet actif ayant une autonomie et un objectif.

3.2. DIFFERENTES CATEGORIES D'AGENTS

Deux grandes catégories d'agents peuvent être distinguées : les agents réactifs et les agents cognitifs

3.2.1. Les agents réactifs

Les agents réactifs sont souvent qualifiés de ne pas être "intelligents" par eux-mêmes. Ils sont des composantes très simples qui perçoivent l'environnement et sont capable d'agir sur celui-ci (Lestel & al, 1994). Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent pas de croyances, pas de mécanismes d'envoi de messages. Leurs capacités répondent uniquement au mode stimulus/action qui peut être considéré comme une forme de communication. Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents et présente un comportement global intelligent.

Les agents réactifs sont considérés intelligents au niveau du groupe, du système. En conséquence, l'intelligence est distribuée entre beaucoup d'agents réactifs et le comportement intelligent devrait émerger de l'interaction entre ces agents réactifs et l'environnement.

3.2.2. Les agents cognitifs

Les agents cognitifs sont plus évolués. Ils sont les résultats directs des recherches menées dans le domaine de l'intelligence artificielle. Les agents cognitifs ont une représentation globale de leur environnement et des autres agents avec lesquels ils communiquent. Ils savent tenir compte de leur passé et s'organisent autour d'un mode social d'organisation (Ferber, 1995).

Les systèmes multi-agents constitués uniquement d'agents cognitifs sont constitués d'un nombre d'agents assez faible. Ils réclament des ressources plus importantes que les systèmes d'agents réactifs. La convergence du système vers un état décisionnel stable n'est

pas non plus assurée par l'utilisation de ce type d'agents, mais ils permettent de résoudre des problèmes plus complexes et nécessitant une plus grande abstraction.

3.3. INTERACTION ENTRE AGENTS

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. Par la perception, les agents ont connaissance d'un changement de comportement d'un tiers au travers du milieu. 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 (Chaïb-draa, 1996) :

- 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.

Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs composants en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales. C'est par les échanges qu'ils entretiennent, par les engagements qui les lient, par l'influence qu'ils exercent les uns sur les autres que les composants sont des entités sociales et que de nouvelles fonctionnalités peuvent émerger de ces systèmes d'actions mutuelles. Les groupes sont donc à la fois les résultats des interactions et les lieux privilégiés dans lesquels s'accomplissent les interactions. C'est pourquoi il est généralement impossible d'analyser des organisations sociales sans tenir compte des interactions entre leurs membres.

L'interaction d'un agent avec un autre constitue à la fois la source de sa puissance et l'origine de ses problèmes. C'est en effet parce qu'ils coopèrent que des composants peuvent espérer que leur satisfaction sera meilleure que s'ils agissaient d'une manière isolée. Mais c'est aussi à cause de leur multitude qu'ils doivent coordonner leurs actions et résoudre les conflits.

Ferber (Ferber, 1995) définit une situation d'interaction comme *un ensemble de comportements résultant du regroupement des composants qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées, dont ils disposent et de leurs compétences individuelles.*

L'interaction constitue donc un niveau d'abstraction supérieur à la notion de communication et d'action. Le problème est alors de savoir combiner ces éléments (communications et actions) afin de coordonner et de contrôler les échanges entre plusieurs agents pour avoir un comportement collectif cohérent du système.

Pour mieux visualiser les différentes situations d'interaction inter-agents, Ferber (Ferber, 1995) a bien classé ces cas en se basant sur les trois composantes principales de l'interaction, à savoir la nature des buts, l'accès aux ressources et les compétences des agents. Le tableau II.1, envisage les différents cas possibles. Il aboutit ainsi à une première typologie des situations d'interactions : indépendance, collaboration simple, encombrement, collaboration coordonnée, compétition individuelle pure, compétition collective pure, conflit individuel, conflit collectif. Le tableau met en évidence l'influence de trois critères (objectifs, compétences, ressources) sur la situation d'interaction induite. En l'occurrence, la compatibilité des objectifs induira des situations de coopération dès que les ressources ou les compétences sont insuffisantes.

| Buts | Ressources | Compétences | Type de situation | Catégorie |
|---------------|-------------------|--------------------|---|--------------------|
| Compatibles | Suffisantes | Suffisantes | <i>Indépendance</i> | Indifférence |
| Compatibles | Suffisantes | Insuffisantes | <i>Collaboration simple</i> | Coopération |
| Compatibles | Insuffisantes | Suffisantes | <i>Encombrement</i> | |
| Compatibles | Insuffisantes | Insuffisantes | <i>Collaboration coordonnée</i> | |
| Incompatibles | Suffisantes | Suffisantes | <i>Compétition individuelle pure</i> | Antagonisme |
| Incompatibles | Suffisantes | Insuffisantes | <i>Compétition collective pure</i> | |
| Incompatibles | Insuffisantes | Suffisantes | <i>Conflits individuels pour des ressources</i> | |
| Incompatibles | Insuffisantes | Insuffisantes | <i>Conflits collectifs pour des ressources</i> | |

Tableau II.1 : Classification des situations d'interaction (Ferber, 1995)

Pour la situation de coopération qui nous concerne, nous remarquons que cette situation est caractérisée essentiellement par la compatibilité des buts entre les membres de l'interaction et le reste des critères est toujours comblé par ces membres en partageant les ressources et les compétences. Nous détaillons dans ce qui suit toutes les formes concernant la catégorie de coopération.

3.3.1. Collaboration simple

Ce type se caractérise par des buts compatibles, des ressources suffisantes, et des compétences insuffisantes. La collaboration simple consiste en une simple addition des compétences ne nécessitant pas d'actions supplémentaires de coordination entre les intervenants. Cette situation est caractéristique des systèmes communicants dans lesquels toute l'interaction s'exprime sous la forme d'allocation de tâches et de partage de connaissances. Les systèmes multi-spécialisés, dont les agents ne font que partager les connaissances pour aboutir à la résolution d'un problème, ne font généralement intervenir que de la collaboration simple.

3.3.2. Encombrement

Ce type d'interaction regroupe des *buts compatibles, des ressources insuffisantes, et des compétences suffisantes*. L'encombrement est caractéristique de toutes les situations dans lesquelles les composants se gênent mutuellement dans l'accomplissement de leurs tâches alors qu'ils n'ont pas besoin les uns des autres. L'exemple caractéristique est celui de la circulation de voitures, de la régulation du trafic aérien, de l'utilisation optimale des ressources telles que la définition d'emplois du temps, la gestion des stocks, l'affectation des tâches à des processeurs, etc.

3.3.3. Collaboration coordonnée

Ce type se caractérise par des *buts compatibles, des ressources insuffisantes, et des compétences insuffisantes*. La collaboration complexe suppose que les composants doivent coordonner leurs actions pour pouvoir disposer de la synergie de l'ensemble de leurs compétences. Presque toutes les activités industrielles nécessitant une approche distribuée, telles que le contrôle de réseaux, la conception et la fabrication de produits industriels, la régulation distribuée ou la réalisation de sociétés de robots autonomes. Ces activités se situent dans le cadre de la collaboration coordonnée. Il s'agit donc de la plus complexe des situations de coopération, puisqu'elle ajoute aux problèmes d'allocation de tâches des aspects de coordination dus aux ressources limitées.

3.4. COMMUNICATION ENTRE AGENTS

Traiter le problème de l'interaction c'est se donner les moyens non seulement de décrire les mécanismes élémentaires permettant aux composants d'interagir, mais aussi d'analyser et de concevoir les différentes formes d'interaction que des composants peuvent pratiquer pour accomplir leurs tâches et satisfaire leurs buts. Tout d'abord, les composants doivent être capables, par le biais de la communication, de transmettre des informations, mais surtout d'induire chez l'autre un comportement spécifique.

Communiquer est donc une forme d'action particulière qui, au lieu de s'appliquer à la transformation de l'environnement, tend à une modification de l'état mental du destinataire (AFCET/AFIA, 1996).

Il faut distinguer les communications indirectes (ou implicites), où l'environnement propage (et éventuellement déforme) les signaux déclenchés par la réalisation d'une action, des communications directes (ou explicites), où il y a envoi ou diffusion intentionnelle de messages entre les composants.

L'interaction entre agents qui est généralement mise en œuvre par une communication est définie par (Ferber, 1997):

- Le contenu des messages (échange de signaux, de requêtes, de plans, d'offre, de connaissances, d'arguments, de contenu figé ou dynamique...), leur type et leur canal (direct par envoi de messages ou indirect par l'environnement), leur forme (protocole de communication utilisé, destinataire).
- Ce qui la provoque ou la motive (engagement a priori à communiquer, intentionnalité explicite ou interaction suscitée de manière dynamique par une perception de dépendance, par exemple).
- Ce qu'elle induit directement (modification des croyances, apparition de conflits, problèmes de cohérence) ou indirectement (en terme de réaction à une perception au travers de l'environnement).
- Les agents impliqués par cette interaction, humains ou artificiels.

L'intérêt des langages d'interaction entre agents est de réduire les communications en évitant une description exhaustive des messages *ad hoc* et une gamme inutilement étendue de protocoles. Ces langages se focalisent essentiellement sur la manière de décrire exhaustivement des actes de communication d'un point de vue syntaxique et sémantique supportant un langage de représentation des connaissances. Toutefois, il n'est pas facile de garantir un comportement collectif et cohérent du système juste en se basant sur l'aspect ontologique* avec l'utilisation de conventions liées à la conversation. Plusieurs tentatives de normalisation de la communication inter-agents ont été effectuées au sein de la communauté multi-agents ces dernières années, parmi ces tentatives nous citons :

* Ensemble structuré de concepts. Les concepts sont organisés dans un graphe dont les relations peuvent être ou des relations sémantiques, ou des relations de composition et d'héritage (au sens objet). Nous abordons ce concept dans les prochaines sections de ce manuscrit.

3.4.1.1. Langage de communication KQML

KQML est un langage de communication et un protocole de haut niveau pour l'échange de l'information, orienté messages et indépendant de la syntaxe du contenu et de l'ontologie applicable. Le langage KQML (Knowledge Query and Manipulation Language) (Labrou & al, 1997) a été développé au sein du groupe de travail «External Interfaces Group» du projet «Knowledge Sharing Effort» financé par l'ARPA («Advanced Research Projects Agency»). Ce langage peut être considéré comme composé de trois couches : une couche de communication qui décrit le niveau des paramètres de communication; une couche de message qui contient un performatif et indique le protocole d'interprétation (les performatifs indiquent quel type d'acte de langage l'agent désire employer et définissent les opérations possibles que les agents peuvent demander sur les connaissances et les objectifs des autres agents; et une couche du contenu qui contient des informations relatives au performatif soumis.

En plus, KQML est indépendant du mécanisme de transport (TCP/IP, SMTP, IIOP ou autre), indépendant du langage du contenu échangé (KIF, SQL, Prolog ou autre) et indépendant de l'ontologie utilisé. Voici un exemple simple d'un message KQML (Ricordel & al, 2000) :

```
( Register
: sender agent A
: receiver agent B
: reply-with message 2
: language common_language
: ontology common_ontology
: content '(service Provision Manufacturing :
TaskDecomposition)'
```

3.4.1.2. Langage de communication FIPA-ACL

Récemment, la collaboration internationale des membres d'organisations universitaires et industrielles regroupées au sein de FIPA (FIPA, 2002b) (*Foundation for Intelligent Physical Agents*) a permis de spécifier des standards dans la technologie agent et vise à favoriser l'interopérabilité des applications, des services et des équipements informatiques basés sur le paradigme agent. Ils ont défini un certain nombre de spécifications principales d'agents. Notamment, un standard de langage de communication agent ACL (*Agent Communication Language*) a été proposé et spécifié. Comme KQML,

ce dernier est basé sur la théorie des actes de langage: les messages sont des actions ou des actes communicatifs, car ils sont prévus pour effectuer une certaine action en vertu de l'envoi. Les spécifications de FIPA-ACL se composent d'un ensemble de types de message et de la description de leur pragmatique qu'est, des effets sur les attitudes mentales des agents (expéditeur et récepteur). Les spécifications décrivent chaque acte communicatif avec une forme narrative et une sémantique formelle basés sur la logique modale.

FIPA-ACL est superficiellement semblable à KQML comme le démontre l'exemple qui suit. Sa syntaxe est identique à celle de KQML.

```
(inform
  :sender A
  :receiver B
  :reply-with laptop
  :language KIF
  :ontology ordinateurs
  :content (= (prix HP-Jet) (scalar 1500 USD))
  :reply-by 10
  :conversation-id conv01
)
```

Un point important dans toute approche de l'interaction est de découpler un modèle de communication de celui de l'interaction. Communication et interaction ne sont pas sur le même plan : l'une est le support de l'autre. La communication se rapproche plus de la transmission physique, alors que l'interaction implique une réciprocité que ne possède pas la communication.

L'interaction est plus liée à l'utilisation qui sera faite du contenu de la communication qu'au contenu de l'acte. C'est à ce niveau qu'intervient la notion d'interprétation d'un acte. De plus, l'interaction peut provenir des réactions aux signaux et non des signaux eux même. On peut avoir une communication sans interaction mais pas une interaction sans communication.

Les protocoles d'envoi de messages synchrones ou asynchrones sont de ce point de vue très critiquables car ils mélangent du traitement de signaux et des informations de contrôle d'un niveau sémantique plus élevé (AFCET/AFIA, 1996). Un modèle de communication formalise des propriétés sur l'absence de perte de signaux, sur leur portée limitée ou encore sur la conversation de leur ordre, quel que soit le routage entre un

émetteur et un récepteur donné. Un modèle d'interaction, au contraire, traite de flots d'information et de contrôle : interaction synchrone (initiateur et exécutants décident du moment où ils interagissent) ou interaction asynchrone (l'initiateur émet un événement que des exécutants viendront peut-être consulter ultérieurement).

L'interface entre les deux modèles doit décrire comment un signal est interprété par un message de contrôle ou porteur d'information, et inversement comment coder des signaux. L'approche de Benyon (AFCET/AFIA, 1996) décrit une telle interface : l'interprétation d'un signal par un message dépend d'un contexte sémantique lié à l'interaction.

4. SYSTEMES MULTI-AGENTS & BUSINESS ELECTRONIQUE

Après avoir présenté les points forts qui caractérisent le paradigme agent et qui lui poussent d'être un vrai cadre conceptuel pour les applications distribuées basées sur la coopération et les interactions continues. Nous concluons que les opportunités derrière l'utilisation des agents dans une application e-business sont énormes. Par exemple, ils peuvent être utilisés pour la demande des prix en temps réel, les ventes aux enchères, la gestion des chaînes d'approvisionnement. Les fournisseurs peuvent présenter leurs produits sur le Web et récupérer les offres des clients en temps réel. De ce fait, les rôles des agents dans une application e-business sont multiples ce qui engendre une typologie des agents e-business que nous décrivons ci-dessous.

4.1. TYPOLOGIE DES AGENTS E-BUSINESS

Il est évident d'après plusieurs exploitations que les agents représentent l'une des technologies les plus convenables qui peuvent être utilisé pour assurer les performances exigées dans les applications e-business (Negri & al, 2006). Les différents travaux courants ont bien montré que les SMA peuvent consolider énormément les autres technologies orientées e-business telles que : le workflow, les Webb service et le Web sémantique. De ce fait, et dans un environnement e-business basé agent, il est nécessaire d'organiser les agents dans des catégories différentes selon leurs fonctionnalités et leurs compétences. Papazoglou (Papazoglou, 2001) définit plusieurs types d'agents e-business qui sont :

4.1.1. Agents d'application

Une application e-business est un système en réseau qui comprend un grand nombre d'agents d'application. Chaque agent est spécialisé dans un seul secteur d'expertise et fournit l'accès aux informations et les sources de connaissance disponibles dans son domaine de travail coopérativement avec d'autres agents pour résoudre un problème complexe.

4.1.2. Agents personnels (ou d'interface)

Les agents personnels (ou d'interface) travaillent directement avec les utilisateurs. Ils assistent l'utilisateur dans la présentation, l'organisation, la gestion de son profil et la collection des informations. Un agent personnel permet l'accès facile et efficace à l'ensemble des informations et des services largement distribué sur le web. L'agent personnel observe et dirige les actions prises par son utilisateur et suggère les meilleurs chemins pour exécuter les tâches. Ces agents peuvent aider les utilisateurs pour former des questions et trouver l'emplacement des réponses.

4.1.3. Agents courtiers

Ce sont des agents e-business qui fournissent la facilité pour localiser une information sur le Web ou encore des agents qui interviennent pour résoudre un problème commun, comme celui de la vente et la distribution (pages blanches et jaunes). Un agent courtier possède la capacité de maintenir, mettre à jour, et accéder à des répertoires d'informations concernant l'application e-business.

4.1.4. Agents de négociation

Ce sont des agents qui peuvent négocier les termes d'une transaction e-business comme l'échange de service et le paiement. Le processus de négociation est une "séquence de conversation," où des messages multiples sont échangé selon un protocole bien défini. Les termes d'une transaction e-business peuvent concerner par exemple, les droits de distribution, la politique de remboursement, arrangement pour le paiement et le crédit, le copyright, etc.

Caglayan et Harrison (Caclyan & Harrison, 1998) considère que dans la littérature, il existe d'autres typologies d'agents e-business. Ces auteurs ont parlé des agents dans le monde commercial et industriel. Pour eux, les agents peuvent être classifiés du point de vue de l'utilisateur final. De ce fait, ces auteurs ont distingué trois types d'agents :

- Les agents bureautiques
- Les agents Internet
- Les agents Intranet

Les agents bureautiques ou d'interface sont groupés en trois agents (Fourati, 2006):

- Les agents système : sont des agents d'interface qui aident dans l'utilisation du système d'exploitation.
- Les agents d'application : sont des agents d'interface fournissant une aide à l'utilisateur pour se servir d'une application particulière.
- Les agents de suite logicielle : sont des agents d'interface fournissant une aide à l'utilisateur pour faciliter son travail avec des applications corrélées.

Les agents Internet sont classifiés en sept agents (Fourati, 2006):

- Les agents de recherche dans le Web: sont des agents Internet qui fournissent des services de recherche dans le Web à un utilisateur.
- Les agents serveurs du Web: sont des agents Internet résidant sur un site Web spécifique pour fournir des services.
- Les agents de filtrage d'information : sont des agents Internet qui filtrent des informations selon des critères spécifiés par l'utilisateur.
- Les agents de recherche documentaire : sont des agents Internet qui répondent par un ensemble personnalisé d'information correspondant à la demande de l'utilisateur.
- Les agents de notification : sont des agents Internet indiquant à un utilisateur des évènements qui pourraient l'intéresser.
- Les agents de service : sont des agents Internet fournissant des services spécialisés à des utilisateurs.
- Les agents mobiles : sont des agents Internet se déplaçant d'un lieu à un autre afin d'exécuter les tâches spécifiques d'un utilisateur.

Les agents Intranet sont groupés sous quatre types (Fourati, 2006):

- Les agents de personnalisation coopérative : sont des agents Intranet permettant l'automatisation du workflow à l'intérieur d'une entreprise.
- Les agents d'automatisation : sont des agents Intranet automatisant les tâches d'une entreprise.
- Les agents de base de données : sont des agents Intranet fournissant des services agent à l'utilisateur de bases de données.
- Les agents courtiers de ressources : sont des agents Intranet effectuant l'allocation des ressources dans les architectures client/serveur.

La figure II.3, nous résume cette richesse de typologie par une schématisation arborescente proposée par Papazoglou (Papozoglou, 2001) :

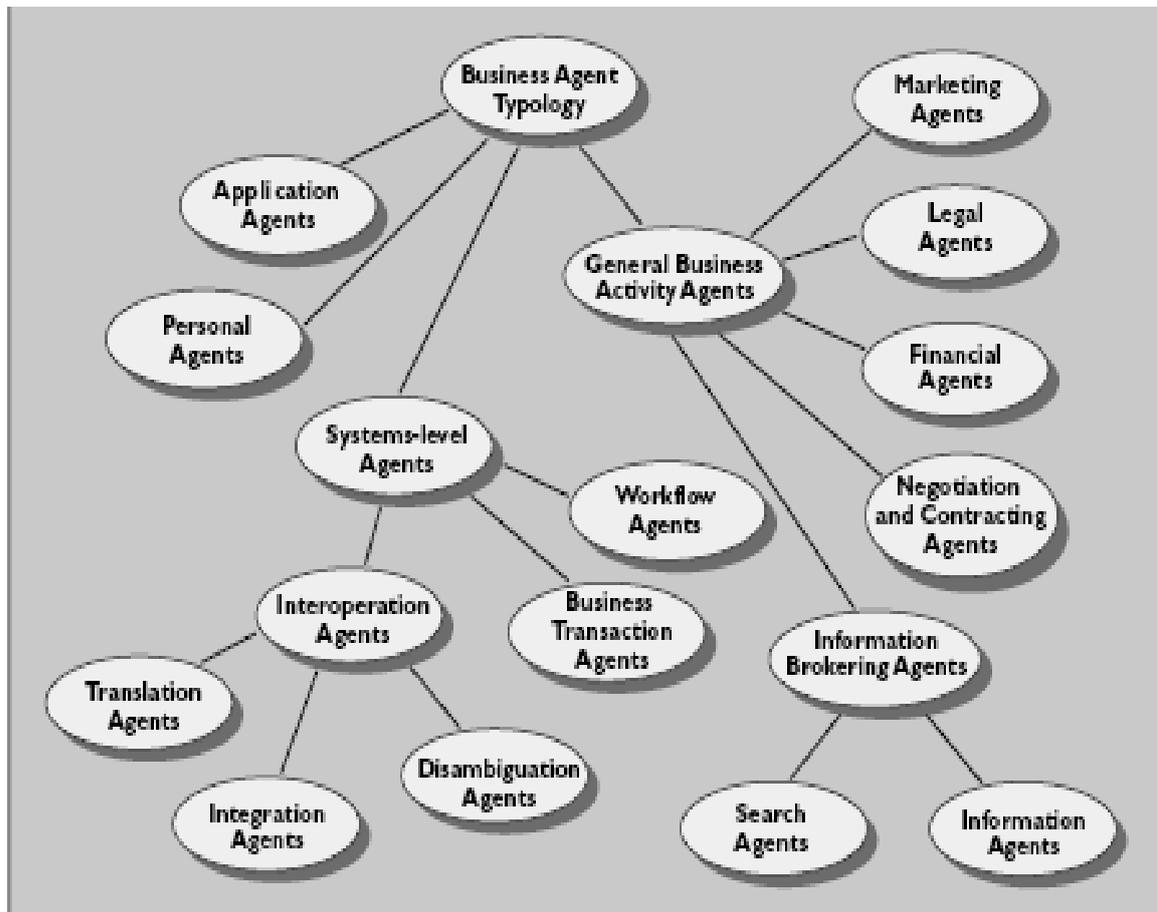


Figure II.3. Typologie des agents e-business (Papazoglou, 2001)

4.1.5. Agents mobiles

En plus de toutes ces sortes d'agents présentés, nous mettons l'accent sur un type particulier qui connaît un grand succès ces dernières années, c'est l'agent mobile. Ce concept dérive de deux domaines différents, les agents venant de l'intelligence artificielle avec les systèmes multi-agents et des systèmes distribués avec la migration de processus (Cubat Dit Cros, 2005) .

La programmation par agents mobiles est un paradigme de programmation des applications réparties, susceptible de compléter ou de se substituer à d'autres paradigmes plus classiques tel que le passage de messages, l'appel de procédure à distance, l'invocation d'objet à distance, l'évaluation à distance. Elle est d'un grand intérêt pour la mise en œuvre d'applications dont les performances varient en fonction de la disponibilité et de la qualité des services et des ressources, ainsi que du volume des données déplacées. Le concept d'agent mobile facilite en effet la mise en œuvre d'applications dynamiquement adaptables, et il offre un cadre générique pour le développement des applications réparties sur des réseaux de grande taille qui recouvrent des domaines administratifs multiples. En pratique, 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. Une application construite à base d'agents mobiles peut se redéployer dynamiquement suivant un plan pré-établi ou en réaction à une situation particulière, afin par exemple d'améliorer la performance ou de satisfaire la tolérance aux pannes, de réduire le trafic sur le réseau, ou de suivre un composant matériel mobile. La mobilité du code offre un premier niveau de flexibilité aux applications. La décentralisation de la connaissance et du contrôle à travers les agents, et la proximité physique entre les agents et les ressources du système renforce la réactivité et les capacités d'adaptation.

5. PLATE-FORMES & APPLICATIONS BASEES AGENTS

Une large variété de systèmes, de langages et d'efforts de standardisation ont été déployés pour concevoir des applications agents; ceci est dû en partie à l'imprécision du terme "agent", et d'autre part de la richesse des possibilités offerte par ce paradigme. Nous exposons ci-après quelques plateformes de développement des applications agents :

| Produit | Société | Langage | FIPA | MASIF | Description |
|------------|-------------------------|---------|------|-------|------------------------|
| AgenTalk | GenSym | LISP | NON | | Coordination |
| Aglets | IBM | Java | NON | | Agents mobiles |
| Concordia | Mitsubishi | Java | NON | | Agents mobiles |
| Grasshoper | IKV++ | Java | OUI | OUI | Agents mobiles |
| JACK | Agent-Oriented Software | Java | OUI | | |
| JADE | CSELT=TILab | Java | OUI | | |
| MadKit | LIRMM (GPL) | Java | OUI | | Modèle organisationnel |
| Voyager | ObjectSpace | Java | | | ORB étendu par agent |
| ZEUS | BT | Java | OUI | | |

Tableau II.2 : Plateformes pour le développement basé agent (Schobbens & al, 2003)

De ce fait, il existe plusieurs organisations de recherche qui se penchent de construire des applications agents pour simplifier la vie des utilisateurs dans le domaine du business électronique et cette approche n'est plus un aspect théorique : il existe plusieurs entreprises qui sont en train de développer des applications agents pour le business électronique et les utiliser à l'Internet. Par exemple : PersonaLogic (Persona, 2006) et Firefly (Firefly, 2005) sont deux systèmes commerciaux afin d'assister le choix des consommateurs. Bargain Finder (bargainfinder, 2005) et Jango (maker, 2006) sont deux systèmes se trouvant à l'étape courtage de marchand, où ces systèmes comparent les alternatifs des marchands, mais pas de produits. Ils se trouvent donc à l'étape courtage du produit, et il s'agit des agents au nom de chaque consommateur. Kasbah, Tête-à-Tête, et ActionBot (auctionbot, 2006) sont trois systèmes dans lesquels on a la notion multi-agent et celle de négociation. Ils fournissent des agents d'achat ou des agents de vente pour des utilisateurs. Donc, l'utilisateur crée son agent, le donne une direction stratégique, et l'envoie au marché central des agents. Chaque agent veut accomplir une transaction en utilisant les choix d'utilisateur (prix souhaité, prix minimum (et maximum) acceptable, la date limite, etc.).

Un petit survol sur les produits e-business basés agent nous permet de conclure que la plupart de ces solutions s'orientent précisément vers le domaine du commerce électronique et elles ne fournissent que des réponses partielles concernant les problèmes de communication de négociation ou encore d'hétérogénéité. L'existence des vrais

modèles complets pour le développement e-business est presque nulle. Nous pouvons expliquer ces lacunes par plusieurs raisons qui touchent essentiellement la nature des systèmes multi agent (Schobbens & al, 2003) :

- Les SMA sont encore du domaine de la recherche active
- Emergence des produits pionniers
- Technologie pas encore complète et stabilisée
- Technologie pas encore bien intégrée et pas interopérable
- Pas encore utilisé très largement
- Plus de prototypes que de produits

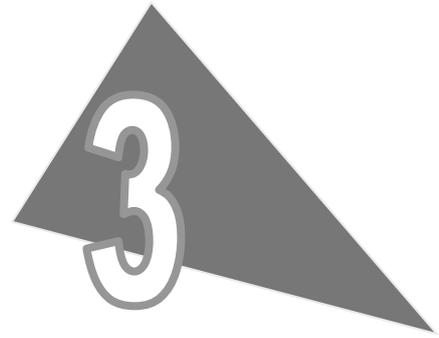
Une des critiques souvent adressée aux SMA porte sur leur prétention à modéliser des comportements humains réels car la représentation des objets et des règles d'action relève parfois de la psychologie et ne peut évidemment pas saisir la complexité des comportements humains. De même, une autre faiblesse relative des SMA concerne la validation des modèles en raison des dangers de la modélisation intégrée. En effet, en cherchant à accroître la précision du modèle sous forme d'agents ou de méthodes, on multiplie le nombre de paramètres. Or plus, il contient de paramètres, plus les possibilités de vérifier la cohérence de la structure deviennent difficiles voire impossibles. Le seul moyen permettant alors de s'affranchir ou du moins de limiter les problèmes liés à la validation est de concevoir un modèle le plus épuré et le plus simple possible.

Une autre raison primordiale, concernant la demande continue d'un middleware universel pour les applications distribuées. Les SMA ont bien essayé d'imposer CORBA (CORBA, 2009) dans le rôle de « Middelware Internet », mais ces tentatives n'ont pas abouti tout simplement parce que ce dernier n'était ni simple, ni léger, ni standard...La recherche d'un middleware substitué était plus que nécessaire jusqu'à l'apparition de la technologie des « Web Services ».

6. CONCLUSION

Les applications agents sont des programmes auxquels on peut déléguer des tâches de plusieurs sortes. Elles diffèrent du logiciel traditionnel par leurs potentialités qui viennent essentiellement de celle de l'agent (autonomie, communication et coopération). Ces qualités rendent les agents particulièrement utiles pour l'environnement du business électronique très riche par les interactions et l'échange continus de l'information. Les SMA peuvent prouver leur efficacité dans de telles applications distribuées. Comme la distribution exige souvent le travail au-dessous d'un middleware. Les entreprises s'orientent fortement vers la technologie des Web services qui tente d'être le middleware universel pour la distribution basée web. Nous abordons dans le chapitre qui suit cette nouvelle technologie et voir comment peut on tirer profit de ses avantages dans notre domaine d'application.

CHAPITRE



Web Services & E-business : Vers un Middleware Universel

1. INTRODUCTION

La technologie des Web services est aujourd'hui de plus en plus incontournable et se présente comme le nouveau paradigme des architectures logicielles distribuées. Cette technologie englobe de nombreux concepts et tend à s'imposer comme le nouveau standard en terme d'intégration et d'échanges des applications au sein et entre les entreprises.

La solution orientée services est une réponse très efficace aux problématiques que rencontrent les entreprises en termes de réutilisabilité, d'interopérabilité et de réduction de couplage entre les différents systèmes qui implémentent leurs systèmes d'information. Les Web services sont devenus la forme de middleware que l'informatique professionnelle attendait depuis des années. Cette forme est constatée même comme un middleware universel pour les applications distribuées de l'entreprise.

Nous allons présenter dans ce chapitre ce nouveau concept et voir comment cette technologie peut combler les lacunes du paradigme agent dans notre solution en offrant des principes suffisamment génériques et souples de manière à permettre un développement efficace et avec moins d'effort.

2. HISTORIQUE DES WEB SERVICES

Les Web services prennent leur origine dans l'informatique distribuée et dans l'avènement du Web. Le but de l'informatique distribuée est de permettre à une application sur une machine d'accéder à un service d'une autre application sur une machine distante et ce, de la même manière que l'appel d'une fonction locale, indépendamment des plateformes et des langages utilisés.

La question alors était de chercher un middleware pour les applications distribuées. La réponse vint avec la proposition des trois standards qui sont les plus imposés dans l'informatique distribuée: CORBA/IIOP (Common Object Request Broker Architecture / Internet Inter-ORB Protocol) (CORBA, 2009), DCOM (Distributed Component Object Model) (Microsoft, 2009) et RMI (Remote Method Invocation) (Java, 2008). Ces modèles offrent des mécanismes de récupération d'espace mémoire, de sécurité, de gestion du cycle de vie des objets, mais sont complexes, peu compatibles avec les pare-feux, et difficilement interopérables entre eux. Ils restent donc souvent confinés à l'intérieur des entreprises (Christophe, 2002).

Le second point qui a favorisé l'essor des Web services, c'est le développement d'Internet par la démocratisation du haut débit, la structuration des données via XML et la recherche d'interopérabilité. Les entreprises publiaient déjà de l'information via des sites Web, utilisaient la messagerie et faisaient du commerce électronique, elle l'utilise maintenant pour leurs applications métiers (ressources humaines, ventes, finances, etc.). La première société à avoir introduit un concept de services distribués, proche de ce qui existe aujourd'hui, est Hewlett-Packard (Hawlett, 2008) avec son produit e-Speak, et ce dès 1999. La suite fut une grande course entre les principaux acteurs du marché qui ont progressivement, par soucis (Christophe, 2002).

Nous allons décrire ci-après la technologie des Web services, et présenter ses impacts sur le domaine du business électronique.

3. WEB SERVICES: Notions & concepts de base

Plusieurs acteurs définissent les Web services par des caractéristiques technologiques distinctives. Mais en général, nous considérons les Web services comme des composants logiciels réutilisables et faiblement couplés qui encapsulent sémantiquement des fonctionnalités discrètes, distribués et accessibles via les protocoles Internet standards (Stencil, 2006). Autrement dit, les Web services sont des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et de protocoles Internet standard. D'après Gillbert (Gillbert & al, 2003), les Web services sont des compléments aux programmes et applications existantes, et servent de pont pour que ces programmes communiquent entre eux. Nous pouvons considérer un Web service comme une application logicielle identifiée par un identificateur de ressource uniforme URI (*Uniform Resource Identifier*). Un Web service est une application autosuffisante en ce sens qu'elle effectue une seule tâche et que ses composantes décrivent ses propres entrées et sorties de telle sorte que d'autres logiciels qui invoquent le Web service puissent interpréter ce qu'il fait, comment invoquer sa fonctionnalité et à quel résultat cet autre service peut s'attendre. Les Web services sont, en quelque sorte, le prolongement de la programmation objet. Ainsi, un Web service est donc une sorte d'objet avec une seule fonctionnalité permettant, avec d'autres Web services, la composition d'une application plus large pouvant avoir plusieurs fonctionnalités. En fait, c'est l'une des briques d'un mur qui en comporte plusieurs. Un URI est la façon d'identifier un point de contenu sur le Web, que ce soit un fichier texte, audio ou vidéo. L'URI le plus connu est l'adresse d'une page Web, par exemple : <http://www.cirano.qc.ca>. Le Web service est donc accessible en spécifiant son URI.

Les Web services se caractérisent par la capacité des interfaces et liaisons (*bindings*) d'être publiées, localisées et invoquées via XML. Un Web service peut-être publié dans un registre situé à l'intérieur ou à l'extérieur d'un système d'information. Un Web service peut être localisé en interrogeant le registre qui l'héberge. Une fois localisé, un Web service peut être invoqué (même par un autre Web service) en envoyant une requête appropriée.

Un autre point bénéfique des Web service, est leur capacité d'interagir avec d'autres composantes logicielles via des éléments XML en utilisant des protocoles de l'Internet. L'une des bases des Web services est l'utilisation de protocoles standards de l'Internet tels que HTTP (*Hypertext Transfer Protocol*, le protocole du Web), SMTP (*Simple Mail*

Transfer Protocol, le protocole du courriel électronique) et XML. Les Web services peuvent donc traverser les pare-feu conventionnels sans problème. Contrairement à une page Web ou à une application de bureautique, les Web services ne sont pas destinés à une interaction humaine directe. Ils sont plutôt conçus pour être utilisés par d'autres logiciels. En plus de ça, les Web services peuvent se voir comme des composantes logicielles légèrement couplées à interaction dynamique. Nous voulons dire par ça, que le Web service et le programme (le consommateur du Web service) qui l'invoque peuvent être modifiés indépendamment l'un de l'autre. Cela veut aussi dire que, contrairement à une composante logicielle qui serait fortement couplée, il n'est pas nécessaire de connaître la machine, le langage, le système d'exploitation ou tout autre détail, pour qu'une communication puisse avoir lieu. Cela offre une flexibilité qui permet aux entreprises d'éviter les coûts engendrés par l'intégration via des communications fortement couplées, telles que requises par l'EDI, par exemple. Par interaction dynamique, on signifie que le consommateur de Web services peut localiser et invoquer ce dernier au moment de l'exécution du programme sans avoir à programmer cette habileté à l'avance. Cela présuppose aussi que l'on peut modifier le Web service sans avoir à modifier le logiciel de chacun des utilisateurs potentiels.

4. OPPORTUNITES DES WEB SERVICES

L'idée fondamentale derrière les Web Services est de morceler les applications et les processus d'affaires en morceaux réutilisables appelés « Service » de sorte que chacun de ces segments effectue une tâche distincte. Ces services peuvent alors servir à l'intérieur et à l'extérieur de l'entreprise, facilitant l'interopérabilité entre tous ces services. De par leur nature, les Web Services :

- permettent à des portions de logiciels écrits dans différents langages, ou évoluant sur différents systèmes d'exploitation, de communiquer entre elles facilement et à peu de frais;
- permettent à des applications supportant différents processus d'une organisation ou de différentes organisations, de communiquer entre elles et/ou d'échanger des données facilement et à peu de frais.

Les Web services implémentent la logique métier rendue consommable par l'utilisation de standards (majoritairement TCP/IP, URI/URN/URL, MIME, HTTP/SMTP/..., SOAP, SSL/TLS, ... pour le transport, puis XML pour le contenu), ce qui permet à n'importe quelle technologie utilisant ces standards de pouvoir l'exploiter, facilitant ainsi l'interopérabilité des applications.

La création de Web services se justifie par l'architecture orientée service, c'est-à-dire la volonté de rendre accessible un service qui implémente une logique métier cachée à des utilisateurs. Dans ce cadre les avantages derrière l'utilisation des Web services sont multiples (eFORCE, 2003) :

- Les Web services utilisent des standards et protocoles ouverts.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Basés sur le protocole HTTP, les Web services peuvent fonctionner au travers de nombreux firewalls sans nécessiter des changements sur les règles de filtrage.
- En ouvrant la voie à la communication inter-applications, les Web services donnent la clé de la communication inter-organisation. Toutes les perspectives imaginées lors de la vague de l'EDI (Echange de Documents Informatisés) redeviennent possibles.
- les Web services représenteraient tout de même un grand enjeu dans l'informatique inter-entreprises, d'aujourd'hui: permettre enfin l'harmonisation des différents systèmes qui se sont accumulés dans chaque entreprise.
- Les Web services supportaient la vague de l'EAI (Enterprise Application Integration). Les opérations d'EAI sont devenues une priorité évidente pour la plupart des entreprises afin de retrouver un minimum de cohérence. Grâce aux Web services cette priorité est désormais possible.

5. ENTOURAGE TECHNOLOGIQUE DES WEB SERVICES

Actuellement, les Web services sont entourés de plusieurs technologies qui viennent tous de d'origine de la technologie XML (XML, 2009). Cette dernière est considérée dans le monde économique comme un outil de gain de compétitivité basé sur l'effet de levier des nouvelles technologies. Il est attendu que l'utilisation de XML pour le business

électronique soit moins coûteuse que l'utilisation des solutions traditionnelles d'échanges EDI (Echange des Données Informatisées) et des autres technologies existantes. Le XML apporte puissance et souplesse aux applications Web et présente de nombreux avantages pour les développeurs et les utilisateurs des Web services. Ces avantages représentent un vrai argument conduisant à choisir XML à la place des technologies EDI traditionnelles. Comme XML offre plusieurs possibilités en termes d'échange d'informations. Plusieurs produits et standards ont été fournis pour offrir des infrastructures ouvertes basée sur XML permettant à tous les partenaires potentiels du business électronique d'utiliser l'information de manière universelle, interopérable, sécurisée et cohérente. Parmi ces infrastructures nous mettant l'accent sur celles qui manipulent, exploitent et travaillent avec les Web services. Ces technologies sont multiples mais on se limite sur le protocole *SOAP*, l'annuaire *UDDI*, le langage de description *WSDL*, et les architectures *ebXML*, *RosettaNet* et *Biztalk*.

5.1. SOAP (Simple Object Access Protocol)

SOAP (W3C, 2008) est un protocole d'échange de messages dans un milieu distribué. SOAP définit aussi un RPC (Remote Procedure Call), mais n'est pas limité à cette utilisation : il est possible d'utiliser SOAP pour des systèmes asynchrones (SOAP ne fonctionne pas nécessairement sur le modèle requête – réponse). Le RPC défini par SOAP résout certains problèmes soulevés par les technologies telles que CORBA ou DCOM pour les applications distribuées sur Internet. En effet, ces RPC sont souvent difficiles à mettre en oeuvre, et les messages échangés ne passent pas les firewall. De plus, ils ne sont pas adaptés aux besoins des applications distribuées sur Internet, où le seul pré-requis devrait être de pouvoir utiliser le protocole http (afin de pouvoir traverser les pare-feu). C'est l'objectif de la norme SOAP, qui a pour principe d'échanger des messages HTTP (ou SMTP) dont le corps est un fichier XML. SOAP se base sur la spécification XML-RPC (Didier & Tanguy, 2001). Le protocole SOAP est composé de deux parties :

- Une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement,
- Un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre.

SOAP a été initialement défini par Microsoft et IBM, mais est devenu une référence depuis une recommandation du W3C, utilisée notamment dans le cadre d'architectures de type SOA (Service Oriented Architecture) pour les Web services.

5.2. UDDI (Universal Description, Discovery and Integration)

Web service UDDI acronyme d'Universal Description Discovery and Integration (UDDI, 2009), a été créé par IBM, Microsoft et Ariba. C'est une architecture répartie qui permet aux fournisseurs de Web services (Business providers), d'enregistrer leurs services, et aux applications de rechercher les services correspondant à leurs besoins, de façon normalisée. UDDI est donc un annuaire distribué de Web services et d'entreprises (Business/Service Registry). UDDI se comporte lui-même comme un Web service dont les méthodes sont appelées via le protocole SOAP (Didier & Tanguy, 2001).

L'annuaire UDDI est consultable de différentes manières :

- Les pages blanches comprennent la liste des entreprises ainsi que des informations associées à ces dernières. Nous y retrouvons donc des informations comme le nom de l'entreprise, ses coordonnées, la description de l'entreprise mais également l'ensemble des ses identifiants.
- Les pages jaunes recensent les Web services de chacune des entreprises sous le standard WSDL.
- Les pages vertes fournissent des informations techniques précises sur les services fournis. Ces informations concernent les descriptions de services et d'information de liaison ou encore les processus métiers associés.

5.3. WSDL (Web Service Description Language)

WSDL (Web Service Description Language) (W3C, 2008) pour les Web services est l'équivalent de IDL (Interface Definition Language) pour la programmation distribuée (CORBA, DCOM). Il s'agit d'une tentative de normalisation regroupant la description des éléments permettant de mettre en place l'accès à un service réseau (Web service). Il fait notamment référence au langage XML et a été proposé en 2001 au W3C pour standardisation. Ce langage permet de décrire de façon précise les Web services, en

incluant des détails tels que les protocoles, les serveurs, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie, et les exceptions pouvant être renvoyées (voire figure III.1).

```
<definitions>
  <message>
    ...
  </message>

  <portType>
    <operation> ... </operation>
    <operation> ... </operation>
    ...
  </portType>

  <binding>
    ...
  </binding>

  <service>
    <port> ... </port>
    <port> ... </port>
  </service>
</definitions>
```

Figure III.1. Structure d'un document WSDL

Il y eut d'autres tentatives de langages pour résoudre le problème de la définition des Web services. Microsoft a d'abord proposé SDL (Service Definition Language) avec une implémentation fournie dans leur Toolkit SOAP, puis IBM a proposé NASSL (Network Accessible Service Specification Language), dont une implémentation est fournie dans SOAP4J. Microsoft modifia sa première spécification et proposa SCL (SOAP Contract Language), puis les différents acteurs s'accordèrent sur WSDL (Didier & Tanguy, 2001).

5.4. EbXML

EbXML (Electronic Business using eXtensible Markup Language) signifiant commerce électronique en utilisant XML, est une suite de spécifications basées sur le langage XML et définissant un standard pour le commerce électronique. Ce standard est édité par OASIS et UN/CEFACT (UN/CEFACT, 2001). L'objectif est de fournir une infrastructure globale, ouverte, fondée sur XML permettant d'assurer les Echanges Electroniques Professionnels (EEP ou B2B en anglais) de manière interopérable, sécurisée et cohérente.

L'architecture ebXML fournit (Didier & Tanguy, 2001) :

- Une méthode de description des entreprises ;
- Une méthode de description des processus métiers, les documents associés (DTD ou XML-Schéma), un vocabulaire XML commun, et les échanges de messages XML associés ;
- Un répertoire commun où les entreprises stockent leurs informations, enregistrent et découvrent des services proposés;
- Un répertoire où les entreprises peuvent stocker leurs modèles d'affaires avec les messages associés ;
- Une méthode de définition et de négociation d'accords de collaboration, ou collaboration partner agreement (CPA). Le langage TpaML d'IBM (Trading Partner Agreement Modeling Language) (Sachs & al, 2008) permet en particulier de gérer les négociations et de définir les accords de collaboration entre partenaires ;
- Une architecture de transport de messages XML, utilisant des standards tels que : HTTP, TCP/IP, Mime, SMTP et FTP. A noter que cette architecture intègre SOAP.

La figure ci-dessous montre un cas d'utilisation entre deux entreprise A et B qui commencent par la configuration des interfaces ebXML et aboutit à l'exécution d'une transaction d'affaires simple (Recherche d'un partenaire pour l'achat d'un produit) dans un interchange entre les deux entreprises. (Didier & Tanguy, 2001).

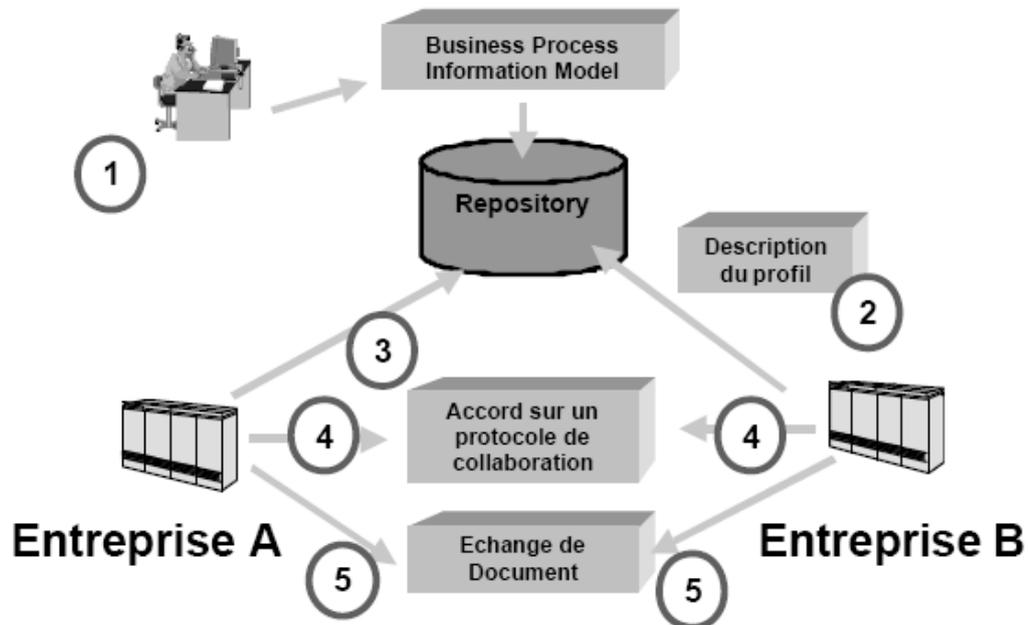


Figure III.2. Interactions entre deux entreprises opérant des transactions ebXML

1. Un organisme normalisateur (RosettaNet, OAGIS, ...) définit le processus 'Demande de devis', sous forme de Business Process Information Model, et l'enregistre dans le répertoire ebXML. Ce modèle précise la séquence des messages échangés au cours du processus, et fournit les documents métiers, communs à toute la profession (devis, formulaire de demande de devis, acquittement, reçu, etc.);
2. L'entreprise B décrit son profil et s'enregistre dans le référentiel ebXML. Elle précise qu'elle fournit le processus 'Demande de devis', et enregistre les détails de l'implémentation qu'elle fournit ;
3. L'entreprise A cherche la liste des entreprises vendant un produit particulier, et fournissant le processus 'Demande de devis' dans le référentiel ebXML, et trouve l'entreprise B ;
4. Optionnellement, les entreprises A et B négocient le protocole de collaboration : infrastructure de communication, format des messages échangés, conditions de sécurité, etc., et produisent un document TPA, base de la collaboration ;
5. Les entreprises A et B s'échangent les documents métiers (demande de devis, réponse, ...) en suivant les règles de séquence du processus défini. Ces documents peuvent alors être utilisés par les applications des entreprises A et B.

5.5. RosettaNet

Créée en 1998, RosettaNet est un consortium d'industriels des secteurs informatiques-électronique & semi-conducteurs rassemblant plus de 400 membres, dont le but est de formaliser, en XML, le dialogue entre les partenaires d'une transaction commerciale, afin de définir les éléments nécessaires à l'automatisation des Web services dans une chaîne logistique.

L'architecture de RosettaNet repose sur trois notions (Unilog, 2003) :

- Des templates, ou modèles de processus métiers, les PIPs (Partner Interface Processes), référencés au sein d'un catalogue RosettaNet
- Des dictionnaires et des codes pour harmoniser les vocabulaires XML
- Un cadre d'implémentation, le RosettaNet Implementation Framework (RNIF)

Grâce à la présence de ces trois notions, il est possible d'échanger des messages RosettaNet, en respectant un enchaînement défini par le PIP, et avec une sémantique compréhensible par les parties, grâce aux dictionnaires et aux codes.

Les PIPs gèrent les transactions entre partenaires de l'échange. Ils représentent des enchaînements de tâches que les entreprises ont rendu publiques, pour gérer leur échange. Les référentiels d'harmonisation de la sémantique sont au nombre de cinq (trois types de codification et deux types de dictionnaires) :

- DUNS (Data Universal Numbering System) est un code à 9 chiffres géré par Dun & Bradstreet, permettant d'identifier les partenaires à un échange
- UN/SPSC, également géré par Dun & Bradstreet, est un code de classification des produits et sous-familles de produits en 4 niveaux
- GTIN (Global Trade Item Number) est un code à 14 chiffres, géré par EAN international (European Article Number) et UCC (Uniform Code Council), pour l'identification des produits
- RNTD (RosettaNet Technical Dictionary), est destiné à la spécification des catalogues et fiches produit
- RNBD (RosettaNet Business Dictionary), est destiné à la description de l'objet métier véhiculé (ex : la facture)

La combinaison des trois codes fournit une description unique et compréhensible par les parties, de l'origine et de la nature des produits échangés. Tandis que la combinaison des deux dictionnaires, caractérise sans erreur possible, les données transmises dans les documents.

5.6. BizTalk

BizTalk est un produit Microsoft (Microsoft, 2008) qui permet de connecter les applications, les employés et les partenaires à travers les processus. Ainsi les développeurs ont des interfaces standard de développement, les informaticiens ont à leur disposition des liens directs vers les systèmes. Avec BizTalk Server, Microsoft a introduit les concepts de l'automation, de la gestion, et de l'orchestration de processus dans les entreprises. Avec BizTalk, Microsoft fournit les avantages de conception de développement, de gestion, et de surveillance des processus métiers. Biztalk est le chef d'orchestre de tout processus d'intégration d'applications, et il est capable de récupérer des données sous n'importe quel format et de les transformer pour les rendre accessibles aux différentes plateformes. Son interface graphique permet de 'dessiner' les processus de transformation des données.

BizTalk comprend un système de gestion de projet, c'est un environnement qui permet de concevoir, organiser, et établir les divers éléments des applications de BizTalk. Ce système de gestion de projet appelé BizTalk Project System introduit le concept de single cohesive application. BizTalk Project system est intégré à Visual Studio .Net et fournit un environnement de conception intégrée. Il est l'un des nombreux systèmes de projets disponibles dans Visual Studio, et nous pouvons utiliser plusieurs des outils standard, comme les éditeurs et autres interfaces utilisateurs disponibles. Par exemple, nous pouvons employer l'explorateur de solution pour développer des projets BizTalk et la fenêtre de propriétés pour voir et éditer des propriétés des dossiers, des projets, et des solutions. Quand nous créons un projet BizTalk, nous incluons généralement un ou plusieurs des types de fichiers suivants: orchestrations, schémas, maps et pipelines. Ceux-ci jouent des rôles bien spécifiques dans la création de solution, et le BizTalk Project system fournit un outil de conception graphique correspondant pour chacun d'eux.

Les orchestrations modélisent les processus métiers et le concepteur d'orchestration simplifie le processus de création des orchestrations. Les schémas décrivent les données envoyées et reçues. BizTalk Editor simplifie le processus de création des schémas. Les

schémas sont employés pour décrire le format des données qui sont traitées dans des entreprises et entre les partenaires commerciaux.

Les maps transforment les données d'un format à un autre. Le mapper de BizTalk présente des schémas sources et des schémas cibles côte à côte et permet de définir des transformations entre les éléments d'informations de différents messages.

Les pipelines exécutent une variété d'opérations pour diffuser ou recevoir les messages. Le Pipeline Designer permet de préparer les messages entrants ou sortant en mettant en application des opérations telles que le chiffage et le déchiffage, compression, restructurant, et validation.

6. WEB SERVICES & WEB SEMANTIQUE

Le Web d'aujourd'hui a été conçu à l'origine pour l'usage et l'interprétation humaine. Néanmoins, La quantité d'information de plus en plus importante qui est disponible sur le Web, conjuguée à des nouvelles perspectives d'applications comme le business électronique ou les Web services, posent de nouveaux défis pour la recherche d'information. La nécessité de manipuler les ressources du Web par des machines et de façon autonome fut introduite dans le cadre du Web sémantique. Ce concept désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et agents logiciels, grâce à un système de métadonnées formelles, utilisant notamment la famille de langages développés par le W3C. L'expression Web sémantique, attribuée à Tim Berners-Lee (Berners-Lee, 2001) au sein du W3C, fait d'abord référence à la vision du Web de demain comme un vaste espace d'échange de ressources entre êtres humains et machines permettant une exploitation, qualitativement supérieure, de grands volumes d'informations et de services variés. Les utilisateurs déchargés d'une bonne partie de leurs tâches de recherche, de construction et de combinaison des résultats, grâce aux capacités accrues des machines à accéder aux contenus des ressources et à effectuer des raisonnements sur ceux-ci.

Le Web sémantique est entièrement fondé sur le Web et ne remet pas en cause ce dernier. Le Web sémantique se base donc sur la fonction basique du Web « classique » : un moyen de publier et consulter des documents. Mais les documents traités par le Web sémantique contiennent non pas des textes en langage naturel (français, espagnol, chinois, etc.) mais des informations formalisées pour être traitées automatiquement. Pour cela les

documents doivent être structurés en RDF (Resource Description Framework) (W3C, 2009). Ce dernier est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées, de façon à permettre le traitement automatique de telles descriptions. Développé par le W3C, RDF est le langage de base du Web sémantique. Ces documents structurés autour du RDF sont générés, traités, échangés par des logiciels. Ces logiciels permettent souvent, sans connaissance informatique, de :

- générer des données sémantiques à partir de la saisie d'information par les utilisateurs ;
- agréger des données sémantiques afin d'être publiées ou traitées ;
- publier des données sémantiques avec une mise en forme personnalisée ou spécialisée ;
- échanger automatiquement des données en fonction de leurs relations sémantiques ;
- générer des données sémantiques automatiquement, sans saisie humaine, à partir de règles d'inférences.

Avec le progrès de ce domaine, les idées deviennent de plus en plus ambitieuses. Le Web sémantique n'est plus restreint sur la représentation et la manipulation des connaissances structurelles, mais aussi sur celles de nature fonctionnelle, c'est-à-dire de représenter le fonctionnement des composants actifs qui y agissent et c'est le cas avec les Web services (Sabouret, 2002). Le couplage entre le domaine des Web services et celui du Web sémantique fait introduire les Web Services Sémantiques (WSS). Ce nouveau domaine représente une évolution nécessaire des Web Services. Le but des WSS est principalement que les Web services soient mieux rendus sans engendrer de surcharge pour les utilisateurs. En d'autres termes, l'idée poursuivie avec les services web est de mieux exploiter les technologies de l'Internet en substituant, autant que possible, les humains qui réalisent actuellement un certain nombre de services (ou tâches), par des machines en vue de permettre une découverte et/ou une composition automatique de services sur l'Internet. Le besoin d'automatisation du processus de conception et de mise en œuvre des web services rejoint les préoccupations à l'origine du web sémantique, à savoir comment décrire formellement les connaissances de manière à les rendre exploitables par des machines, sachant que l'exploitation de ces ressources pose deux grands problèmes : il faut pouvoir les localiser et interagir avec elles pour composer des services plus complexes. La

solution à ces problèmes demande un langage de description sémantique du contenu du web qui soit interprétable par un logiciel d'ordinateurs afin de rendre ces tâches, soit semi-automatiques, soit automatiques (Ankolenak & al, 2003). C'est dans ce contexte que plusieurs langages furent développés. Ils se basent essentiellement sur le langage DAML+OIL proposé par DARPA (the Defense Advanced Research Projects Agency). Ce langage permet de décrire une ontologie pour un domaine particulier. Une ontologie peut être vue comme une librairie qui définit un ensemble de termes communs, de concepts propres à un domaine, partagés par les personnes de ce domaine de telle sorte que la sémantique de ces concepts soit commune à ces personnes (Anderson, 2001). Parmi ces langages de description nous trouvons :

6.1. Semantic Web Services Language (SWSL)

SWSL est un langage basé logique pour la spécification des caractéristiques formelles liées aux concepts du Web service (Daml, 2007). Il inclut deux sous langages: SWSL-FOL, un langage de la logique du premier ordre qui est utilisée pour spécifier l'ontologie du service (SWSO), et SWSL-Rules, un sous langage basé règles qui peut être utilisé comme un langage de spécification et à la fois comme un langage d'implantation. Le sou langage SWSL-Rules est conçu comme un support pour une variété de tâches commençant par la spécification du profil du Web service et terminant par sa découverte. Le langage est organisé en couches afin de simplifier l'apprentissage et l'usage de ses différentes parties spécialisées en plusieurs termes de description.

6.2. OWL-S (Ontology Web Language for Services)

OWL-S est un langage de méta description de services. Il utilise la logique de description et les ontologies définies par OWL (Ontology Web Language) (W3C, 2007), langage de définition d'ontologies. OWL-S est un langage issu des travaux du programme DARPA Agent Markup Language (DAML) et prend la suite de DAML-S (DARPA Agent Markup Language Service). Il définit un ensemble de classes et de propriétés afin de constituer l'ontologie. Les intérêts liés à l'utilisation de ce langage inclut la sémantique et contient les fonctions indispensables à la mise en œuvre des Web services : la description, la recherche et l'invocation de service.

L'approche de OWL-S trouve ses origines dans les domaines de l'Intelligence Artificielle où elle était utilisée pour décrire les fonctionnalités des agents dans des

systèmes complexes et hétérogènes (Cabral & al, 2004). OWL-S combine intelligemment l'expressivité des algorithmes de descriptions, les DL (descriptions logics), avec le pragmatisme que l'on trouve dans les standards actuels (SOAP, WSDL), pour décrire les services pouvant être exprimés sémantiquement, mais aussi les agréger en respectant des types de données formalisés (Cabral & al, 2004).

OWL-S se repose sur le langage DAML+OIL qui permet de définir les catégories abstraites des entités, évènements, etc., en termes de classes et de propriétés. OWL-S utilise ce langage de description d'ontologie pour définir une ontologie particulière, celle des "Web Services". Cette ontologie permet de décrire les propriétés d'un "Web Service" ainsi que son ou ses services rendus disponibles au "Monde". Les classes principales de l'ontologie décrites par OWL-S sont illustrées dans la figure III.3 (Ankolenkar & al, 2002). Au coeur de cette ontologie, on trouve la classe service. Elle reprend les propriétés générales d'un "Web Service". Cette classe présente un "Service Profile", est décrite par un "Service Model" et supporte un "Service Grounding" (Dallon, 2003). Le "ServiceProfile" explique ce que fait le service et ce qu'il exige des autres agents, il décrit le service en fonction de ce qu'il fait afin de permettre à un demandeur de voir si le service proposé lui convient.

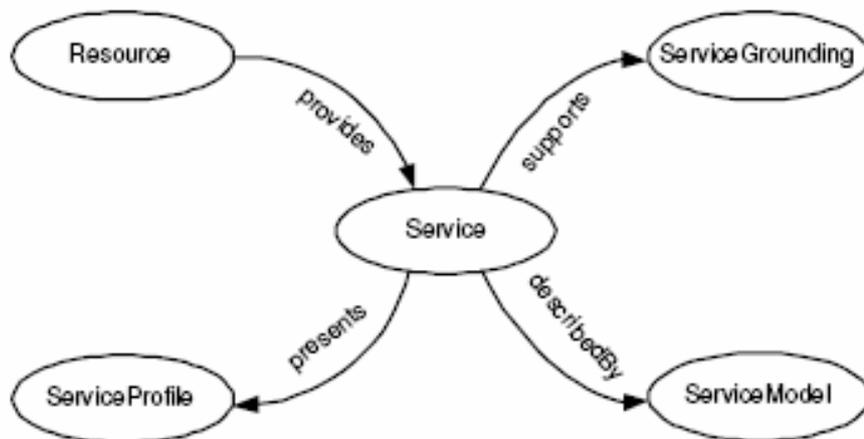


Figure III.3. Classes principales de l'ontologie OWL-S (Dallon, 2003)

Le "ServiceModel" définit le fonctionnement du "Web Service", il sert à expliquer comment le service fonctionne. Un "Web Service" peut être vu comme un processus. C'est pourquoi, le "ServiceModel" possède une sous-classe : le "ProcessModel". Ce dernier comprend deux aspects. Le premier est le processus qui est décrit par l'ontologie de

processus ("Process Ontology"). Le second aspect fait référence au modèle de contrôle du processus défini par l'ontologie de contrôle de processus ("Process Control Ontology"). L'ontologie de processus permet de décrire le processus en termes d'entrées ("inputs"), de sorties ("outputs"), de pré-conditions ("preconditions") et d'effets sur le "Monde" ("effects"). L'ontologie de contrôle de processus sert à définir un modèle de contrôle des instanciations d'un "Web Service" qui doit permettre à un agent de pouvoir surveiller l'état d'avancement du service et de contrôler le service. Le "ServiceGrounding" fournit les informations nécessaires à l'utilisation du "Web Service". En effet, il faudra choisir un protocole à utiliser pour accéder au service, un format de messages, la façon de les sérialiser, quels mécanismes de transport utiliser et quel mode d'adressage employer. Finalement, la classe Ressource donne des informations relatives aux ressources utilisées par le "Web Service". Les "Web Services" ont souvent besoin de ressources pour pouvoir s'exécuter. Ces ressources sont variées et nombreuses. Il est donc intéressant de les définir dans une ontologie. Cette ontologie doit permettre un niveau d'abstraction assez élevé pour couvrir différentes ressources telles que les ressources temporelles, physiques, etc. On peut distinguer deux grands types de ressources appelées "Allocation Types" : celles qui sont consommables ("ConsumableAllocation") et celles qui restent réutilisables ("ReusableAllocation"). Les premières disparaissent avec l'exécution du service, tandis que les secondes sont à nouveau disponibles après l'exécution du service.

7. DEVELOPPEMENT DES WEB SERVICES

Les entreprises qui mettent à profit les nouvelles technologies, en particulier celles, naissantes, des Web services, pour fournir, traiter, qualifier et publier leurs informations, seront celles qui bénéficieront au final d'un nouvel équilibre de la chaîne de valeur ajoutée. Comme les technologies des Web services permettent d'automatiser les adaptations de l'entreprise, réduisant d'autant le coût de l'automatisation des transactions interentreprises et de la diffusion de l'information exigée par le consommateur. Plusieurs entreprises ont déjà utilisés cette technologie pour se bénéficier de ces multiples avantages. Pour bien développer et exploiter les Web services on est besoin de spécifier quels sont les objectifs techniques d'un Web service et comment peut-on se servir des produits disponibles sur le marché informatique.

7.1. CRITERES TECHNIQUES POUR LE DEVELOPPEMENT

Les Web services peuvent servir à l'intérieur et à l'extérieur de l'entreprise, facilitant l'interopérabilité entre tous les partenaires d'un travail commun. De par leur nature, les Web Services devraient permettre plus spécifiquement aux entreprises un certain nombre de point tels que (Patricia Seybold Group, 2002) :

- donner aux clients un accès direct à l'information, aux données et aux fonctionnalités dont ils ont besoin pour interagir avec une entreprise;
- donner aux partenaires d'une entreprise un accès direct à la fonctionnalité dont ils ont besoin pour mieux servir les clients qu'ils ont en commun avec cette entreprise;
- donner aux fournisseurs d'une entreprise un accès direct à l'information et aux fonctionnalités dont ils ont besoin pour leur permettre d'ajuster les inventaires selon le modèle « juste à temps »;
- intégrer des applications de bout en bout, de manière abordable, facile à implanter autant hors des frontières de l'entreprise qu'à l'intérieur de celles-ci;
- avoir des équipes de développement travaillant indépendamment et efficacement sur des systèmes qui interagiront, parce que ces équipes travailleront à développer des interfaces communes plutôt que d'avoir à synchroniser les processus.

Les Web services offrent donc aux entreprises la flexibilité de réponse et d'anticipation des besoins changeant des clients, la rationalisation des infrastructures logicielles et la flexibilité d'interaction et de configuration des alliances externes avec les partenaires et fournisseurs. Il faut aussi bien noter que les divers standards associés aux Web Services et à XML sont encore en mouvance. Cela peut donc créer des difficultés au niveau de l'interopérabilité promise par les Web Services. Dans leur état actuel, les Web Services fournissent une infrastructure logicielle pour les interactions inter-organisationnelles. Cependant, ils ne permettent pas de définir des processus d'affaire inter-organisationnels. Pour ce faire, il faut utiliser des langages (p.ex., WSFL, WSCL ou Xlang) et des architectures se trouvant à des niveaux d'abstraction plus élevés telles que RosettaNet ou ebXML présentés précédemment. Les Web Services fournissent une solution pour le transport de messages entre deux partenaires. Cependant, ils ne décrivent pas comment interpréter les messages transmis. Il appartient donc aux partenaires de s'entendre sur cette

sémantique. Pour l'aspect de sécurisation, ce dernier est actuellement inexistant ou toujours dans sa petite enfance comparée à des normes ouvertes plus mûres de l'informatique répartie telles que CORBA. Actuellement, les Web Services dépendent uniquement de la sécurité offerte par les mécanismes de transport utilisés, tels que SSL (*Secure Socket Layer*), l'utilisation d'un réseau virtuel privé (VPN) ou HTTPS. Ces mécanismes n'offrent habituellement qu'un service de confidentialité des données. Le commerce électronique requiert l'intégrité des données, la non répudiation et le contrôle d'accès ce qui n'est pas nécessairement garanti par les mécanismes de transport. Par l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place au travers des firewalls. Sur un autre volé, les Web services souffrent de performances faibles comparés à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.

D'autre part et loin de l'aspect technique, la venue des Web Services introduit de nombreuses questions de nature légale (Matthew, 2002) :

- Quelle est la nature de la relation entre les entreprises utilisant un Web Services? Quel est le niveau de cohérence et de couplage (léger ou fort)?
- Quel est le niveau de confiance qu'ils partagent?
- Qu'elles sont les nouvelles frontières entre entreprises dans un contexte Web Services?
- À qui appartient quoi dans un contexte d'échange Web Services?
- Qu'elles sont les attentes légales de chaque entreprise transigeant à l'aide des Web Services?
- Quels sont les risques associés à la responsabilité de chacune des parties dans une transaction Web Services? À quel moment précis d'un processus cette responsabilité prend-elle effet? Y a-t-il un partage des risques? Qui est ultimement imputable de la responsabilité?
- Comment assurer la qualité de service par un processus commun?
- Jusqu'à quel point un contrat légal peut-il être modelé et intégré dans un système fédéré?

Les questions et les faiblesses citées au-dessus et bien d'autres sont nécessaires à poser. Les entreprises devront éventuellement trouver réponses au sein des organismes de standardisation, des forums internationaux pour améliorer et profiter pleinement de cette technologie prometteuse.

7.2. OUTILS TECHNOLOGIQUES POUR LE DEVELOPPEMENT DES WEB SERVICES

La disponibilité des produits sur le marché informatique et la facilité de mise en œuvre de cette technologie avec des coûts très bas encouragent de plus en plus les entreprises et les développeurs e-business pour s'orienter vers cette nouvelle solution. Le développement, l'exploitation et le déploiement des Web services peut être réalisé en employant un simple logiciel de serveur d'application tels que:

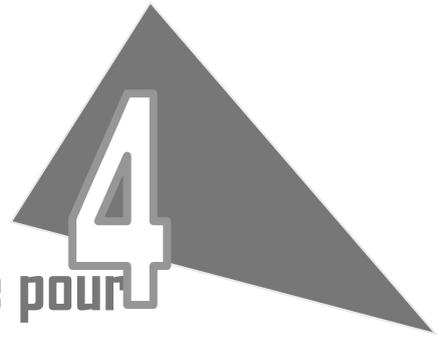
- JAX-WS (Jax, 2007) qui constitue l'implémentation de référence de Java EE est Open Source et intégré dans GlassFish et utilisable dans d'autres environnements. Son extension WSIT (aussi appelée "Project Tango") propose une implémentation de WS-ReliableMessaging, WS-SecureConversation, WS-Trust, ...
- Axis et le serveur de Jakarta Tomcat (deux projets open source d'Apache Software Foundation) (Apache, 2007).
- XFire de CodeHaus offre un framework Java avec une approche différente de Axis.
- ColdFusion MX de Macromedia
- Serveurs HTTP IIS de Microsoft (avec le framework .NET)
- WebLogic de BEA
- WebSphere Application Server d'IBM (basé sur le serveur d'Apache et la plateforme de J2EE)
- Oracle Application Serveur d'Oracle Corporation
- ZenWorks de Novell
- Bibliothèque pour les développeurs de services Web en PHP NuSOAP
- gSOAP: bibliothèque pour les développeurs de services Web en C++
- JBoss Application Server de la société JBoss. Composant du JEMS (JBoss Enterprise Middleware System) dont fait également partie le framework de persistance relationnelle *Hibernate*.

8. CONCLUSION

Fonctionnellement, les Web services n'apportent rien de neuf par rapport aux middlewares, conçus pour appeler des services distants, ou par rapport à l'EDI, qui décrit comment échanger des documents d'affaires. Cependant, la façon avec laquelle les Web services réalisent ces fonctions est entièrement novatrice, découplant le service lui-même de son implantation. Pour les aspects liés aux questions de performance et de sécurité, un couplage de cette technologie avec une autre telle que les systèmes multi agents est jugée nécessaire afin d'atteindre les objectifs prévus du e-business.

Dans cette perspective, nous allons présenter dans ce qui suit une solution basée sur le couplage Agent-Web Service pour le développement des applications e-business coopératives. Nous allons démontrer comment ces deux paradigmes peuvent interagir et combler leurs faiblesses mutuellement et comment peuvent-ils faciliter et assouplir le développement des applications e-business.

CHAPITRE



Un couplage agent-Web Service pour le développement des applications e-business coopératives

1. INTRODUCTION

Les architectures orientées services des entreprises, notamment par le biais de Web services, prennent peu à peu leur place dans les solutions B2B. Néanmoins, même si leur mise en place est déjà bien formalisée (Virdell, 2003), ces architectures ne sont pas adaptées aux contextes inter-organisationnels, distribués et dynamiques. Afin de permettre une prise en charge automatique et dynamique des fonctionnalités fournies par l'entreprise et des Web services, il est essentiel de doter ces architectures par un niveau d'orchestration représenté par des solutions multi agents.

L'intérêt de coupler des agents avec des Web services est de permettre à une entreprise d'exporter, via le réseau Internet, ses compétences et son savoir-faire, ou encore d'ouvrir de nouveaux marchés et de nouveaux supports à la vente de façon automatique, dynamique et essentiellement distribuée. De plus, ce couplage permet aux entreprises d'utiliser à moindre frais les compétences et le savoir-faire des autres entreprises.

Dans ce cadre, et d'après ce que nous avons réclamé comme besoins dans les chapitres précédents, nous proposons dans ce chapitre une solution basée sur le couplage agent-Web service pour le développement des applications e-business coopératives. Cette solution sera présentée sur deux volets. Le premier concerne une solution purement agent et le deuxième représente une phase d'intégration des Web service pour que chaque concept (agent, Web service) puisse combler les faiblesses de l'autre.

2. COUPLAGE AGENT-WEB SERVICE

La progression des paradigmes agent et Web services ont donné lieu à l'apparition de nombreux travaux qui suggèrent une intégration agent et Web service. Selon la multiplicité des stratégies qui visent cette intégration (Clément & al, 2006), nous pouvons distinguer dans ce cadre différentes approches dont l'essentielles:

- Agent et service Web comme des concepts distincts. Les agents sont capables de décrire leurs services comme des services Web et recherchent/utilisent des services Web en faisant des correspondances entre les standards des Web services (WSDL, SOAP) et agents (FIPA SD, FIPA ACL) (Lyell & al, 2003), (Greenwood & al, 2004) (Seghrouchni & al, 2004). Ces approches sont souvent limitées à des agents conformes aux standards FIPA ou utilisent un module d'intégration ou une passerelle. Un aspect difficile de ces travaux consiste à combler l'espace entre les interactions sémantiquement pauvres et synchrones des Web services et les interactions sémantiquement riches et asynchrones des agents.
- Agent et Web service comme le même concept. Les agents et les services Web sont les mêmes entités. Tous les services sont des Web services et sont fournis par des agents (le programme sous-jacent est un agent) (Ishikawa et al, 2004), (Peters, 2005).
- Architecture orientée service basée sur des SMA. Ces travaux ne sont pas directement intéressés dans les interactions entre agents lors de l'échange de service mais plutôt dans l'utilisation des SMA pour améliorer les qualités de services (e.g, la découverte ou la sélection de service) (Maximilien & al, 2003).

En exploitant les architectures citées ci-dessus, nous pouvons aboutir à un rapprochement agent- web service selon différentes directions et pour des raisons précises dont les principales sont les suivantes (Amal & al, 2004).

- L'utilisation des SMA en tant qu'entité médiatrice dans le modèle fonctionnel des services Web. La médiation intervient à plusieurs niveaux. Par exemple, les auteurs des travaux (Mostefaoui & al, 2003), proposent des SMA «Proxy» qui facilitent la

localisation de services Web tandis que dans d'autres travaux tel que (Richarrds & al, 2003) des SMA de planification et de composition de services Web sont présentés.

- L'utilisation des services Web comme cadre architectural et technologique pour mettre en place des SMA accessibles à travers le Web. Dans ce type de sujets nous retrouvons des applications agents qui offrent leurs capacités à travers des Web services. Ici on distingue deux catégories qui diffèrent essentiellement par leur conception.
- Une conception intégrée : ce sont des services Web développés suivant un modèle agent afin de réaliser des tâches complexes telles que la gestion des transactions ou des interactions commerciales (Jin & al, 2003). Ainsi (Petrone, 2003) propose un ensemble de primitives de communication empruntées des langages de communication agents supportant des interactions dynamiques.
- Une conception découplée : à partir d'un SMA donné a priori, une couche à base de services Web rend les capacités de l'agent accessibles à travers le Web que ce soit à d'autres agents de SMA ou à des applications clientes traditionnelles.

Dans notre cas, les Web services ont pour vocation de favoriser une architecture orientée services, intégrant des systèmes hétérogènes complexes, fortement distribués et pouvant coopérer. Les agents ont leurs tâches spécifiques et les Web services s'occupent de toutes les fonctionnalités fortement sollicitées par l'application ou destinées pour le monde extérieur (partenaires, clients et fournisseurs). Dans ce cas un rôle de médiation est assuré par un agent spécifique qui facilite les tâches de recherche et d'invocation en exploitant l'aspect sémantique des Web services.

Nous décrivons ces aspects dans les sections suivantes en mettant en évidence le concept agent et la technologie des Web services ainsi que tous les mécanismes de coopération et de coordination.

3. APERÇU DE L'APPROCHE

Nous allons présenter dans ce qui suit notre solution exploitant le couplage agent-Web Services (voire figure IV.1). Cette solution est composée fondamentalement en deux phases. La première qui représente le noyau de cette solution est basée entièrement sur les agents organisés en un ensemble de groupes (clusters) coopératifs.

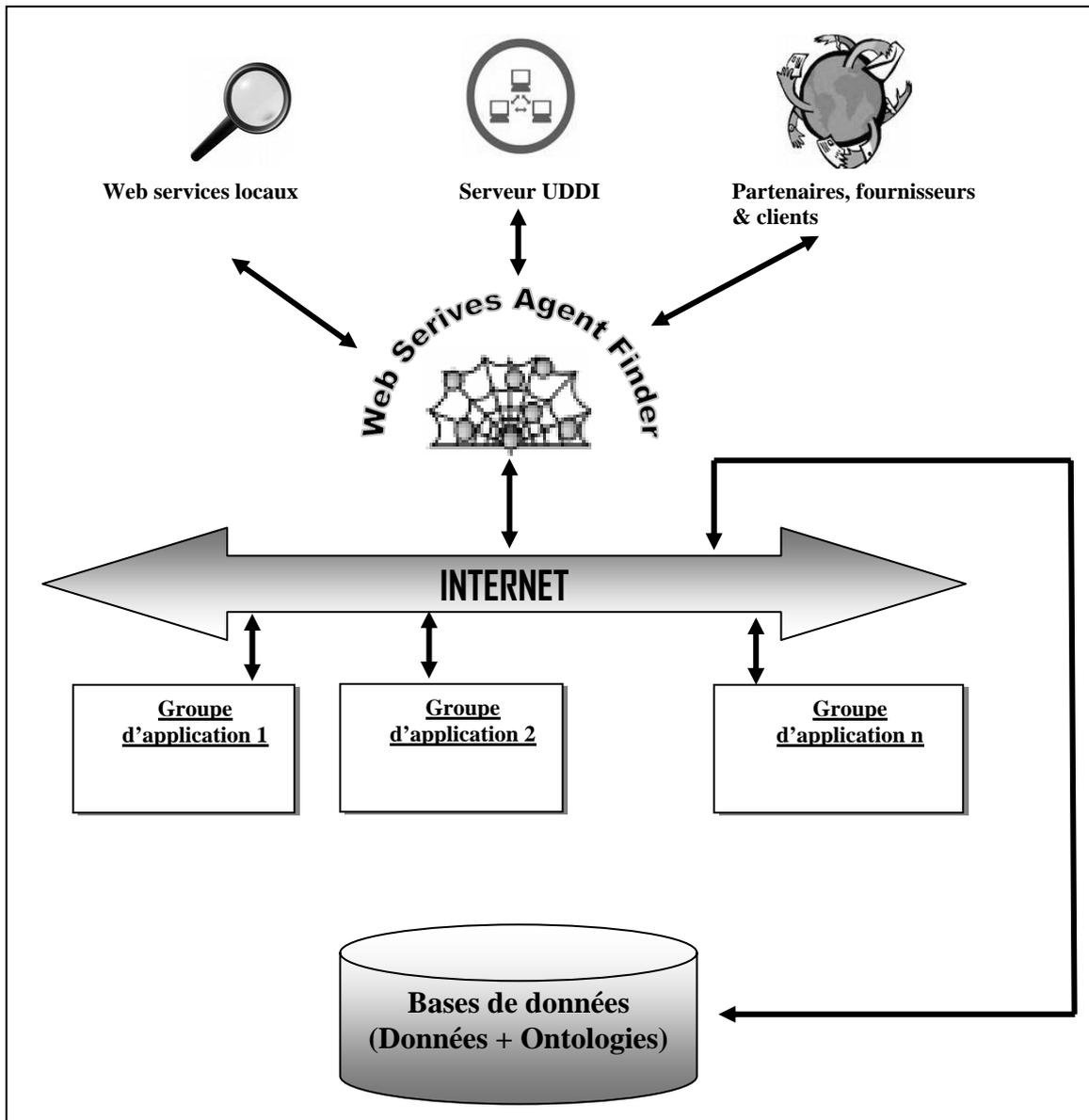


Figure IV.1. Structure globale du modèle proposé

Ces agents interagissent entre eux via leurs agents coordinateurs. La deuxième phase sollicite l'intégration des Web services pour offrir tout ce qui est nécessaire d'être offerts et accessible à travers le Web. La recherche et l'invocation de ces Web services est assurée par un agent intermédiaire qui s'appelle « Web Services Agent Finder ». Cette intégration ne

doit pas dégrader les performances de la partie agent mais au contraire elle libère le noyau agent de toutes les fonctionnalités fortement sollicitées par l'environnement externe. Nous allons détailler dans les prochaines sections chaque composante de cette solution.

4. UN MODELE NOYAU A BASE D'AGENTS

Le modèle noyau dans notre solution est purement basé sur la technologie SMA. L'idée principale avec laquelle on modélise ce noyau est de partager les différentes tâches d'un système (une entreprise) sur plusieurs groupes appelés *Groupes d'Applications*. Ces groupes d'applications sont généralement distribués sur plusieurs sites d'exécution, ils subissent la même structure et les mêmes mécanismes de travail, ils interagissent entre eux comme ils sont accessibles par le monde extérieur.

L'organisation en groupes a été considérée essentiellement par Ferber dans le cadre social des agents à travers son formalisme AGR (Agent Groupe Rôle) (Ferber & al, 2003). Dans le formalisme AGR, un agent est une entité active et communicante. Il peut prendre simultanément plusieurs rôles dans différents groupes. Un groupe est défini par un ensemble d'agents interagissant à travers leurs rôles. Un rôle définit la représentation abstraite de la fonction d'un agent dans un groupe. Le formalisme AGR est commandé essentiellement par un certain nombre d'axiomes dont lesquelles on trouve:

a) Chaque agent est membre d'un (au moins un) groupe:

$$\forall x: \text{Agent}, \exists g: \text{Group}, \text{member}(x,g)$$

b) Deux agents peuvent communiquer seulement si ils sont membres du même groupe:

$$\forall x,y: \text{Agent}, \forall m: \text{Message}, x.\text{send}(y,m) \Rightarrow \exists g: \text{Group}, \text{member}(x,g) \wedge \text{member}(y,g)$$

c) Chaque agent joue (au moins un) rôle dans un groupe:

$$\forall x: \text{Agent}, \forall g: \text{Group} \Rightarrow \exists r: \text{Role}, \text{plays}(x,r,g)$$

d) Un agent est un membre d'un groupe s'il joue un rôle dans ce groupe:

$$\forall x: \text{Agent}, \forall g: \text{Group}, \forall r: \text{Role}, \text{plays}(x,r,g) \Rightarrow \text{member}(x, g)$$

Indifféremment de cette proposition, nous considérons que chaque groupe est spécialisé dans un domaine d'application bien déterminé dont les fonctionnalités sont

assurées par des *Agents d'Applications* qui sont eux même spécialisés dans des tâches spécifiques (Brahimi & al, 2004) .

Chaque agent d'application ne fait partie que d'un seul groupe. La communication, la coordination et la coopération intra et inter groupe sont assurées toujours par un agent spécifique appelé *Agent Coordinateur*. Les agents intervenant dans leur cadre de coopération peuvent communiquer librement avec n'importe quel agent de n'importe quel groupe mais toujours à travers leurs agents coordinateurs. Cela nous conduit à une architecture dont la structure globale représente une forme d'organisation d'agents similaire à celle appelée " systèmes fédérés » (Genesreth & al, 1993)

Nous pouvons formaliser cette proposition à travers les axiomes suivants :

a) Chaque agent est membre d'un (et un seul) groupe:

$$\forall x: \text{Agent}, \exists! g: \text{Group}, \text{member}(x,g)$$

b) Deux agents peuvent communiquer si ils ont besoin de coopérer:

$$\forall x,y: \text{Agent}, \forall m: \text{Message}, x.\text{send}(y,m) \Rightarrow \exists b: \text{Besoin}, \text{Besoin}(x,y) \wedge \text{Besoin}(y,x)$$

c) Chaque agent joue (au moins un) rôle dans (un et un seul) groupe:

$$\forall x: \text{Agent}, \exists g: \text{Group}, \exists r: \text{Role}, \text{plays}(x,r,g)$$

Cette organisation est rigoureuse dans le sens où toutes les interactions passent par l'agent coordinateur avec la possibilité que chaque agent peut interagir avec n'importe quel autre agent dès qu'ils sont besoin l'un de l'autre. Dans le domaine du e-business, une telle organisation est très adéquate pour la gestion des entreprises. Dans la pratique cette organisation correspond fortement à la structuration line (par fonction) où les entreprises s'organisent autour d'un ensemble de services et secteurs spécialisés dans des domaines spécifiques avec des mécanismes de communication et de coordination flexibles. De plus, cette organisation peut offrir de nombreuses solutions sur l'aspect structurel et organisationnel de l'entreprise. Par conséquent, l'indépendance entre groupes est faite dans cette architecture pour assurer que tous les groupes (département, secteurs et services) travail de façon autonome. Un autre avantage derrière cette division en groupes est l'augmentation de la modularité qui est bénéfique dès le développement jusqu'à la maintenance de l'application (Brahimi & al, 2004).

Dans ce qui suit nous détaillons les concepts liés à cette organisation tout en présentant les mécanismes de coopération dans cette structure.

4.1. GROUPES D'APPLICATIONS

Un groupe d'applications est constitué d'un ensemble d'agents d'application qui exécutent des tâches spécifiques et qui possèdent la même structure avec les mêmes mécanismes de travail. Nous considérons un agent d'application comme une entité autonome, réactif et communicante. Les agents d'application possèdent quelques performances, ontologies et expertises (rôles) qu'ils utilisent pour satisfaire leurs besoins ou coopérer avec les autres agents (dans le même groupe ou dans un autre groupe) sous le contrôle et la délégation de leurs agents coordinateurs (voir figure IV.2).

Dans le cas où un agent d'application ne possède pas une compétence dont il a besoin, il demande à son coordinateur de trouver un agent convenable pour l'accomplir. Dans ce cas, le coordinateur doit trouver l'agent convenable en utilisant ses connaissances liées à son groupe. Si aucun agent n'est trouvé dans le même groupe pour satisfaire la demande, le coordinateur envoie la même demande à un autre coordinateur d'un autre groupe en utilisant ses connaissances liées aux autres groupes. Par conséquent le coordinateur récepteur va procéder de la même façon pour trouver un agent fournisseur dans son groupe et envoie le résultat après sa récupération à l'agent émetteur et ce dernier l'envoie de son tour à l'agent demandeur initial.

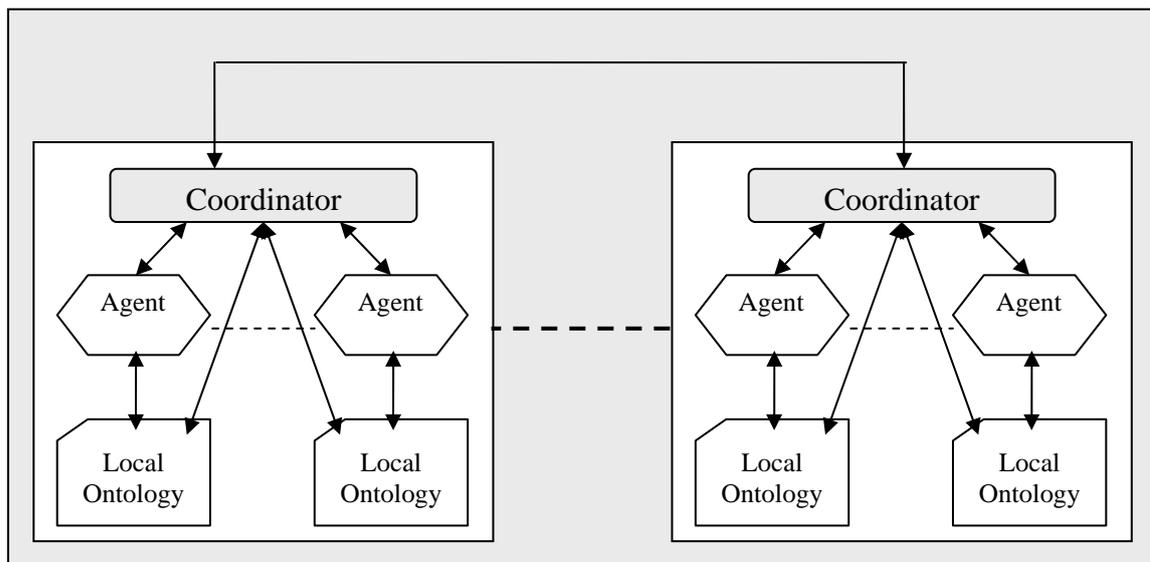


Figure IV.2. Structure des groupes d'application

Pour des raisons de simplicité et efficacité, nous remarquons que la coopération est faite entre agents sans aucune connaissance entre eux. Nous abordons en détail dans ce qui suit les composants de base d'un groupe d'application qui sont : l'agent d'application et l'agent coordinateur. Nous présentons aussi l'interaction et le modèle de la coopération de ce système noyau. Ce modèle est basé sur le protocole Contract Net du FIPA (FIPA, 2002a) en utilisant le langage de communication inter agents ACL toujours du FIPA (FIPA, 2002b).

4.1.1. Agent d'application

Un agent d'application est un agent capable d'exécuter une ou plusieurs tâches. Tous les agents d'application ont la même structure de base (voir figure IV.3).

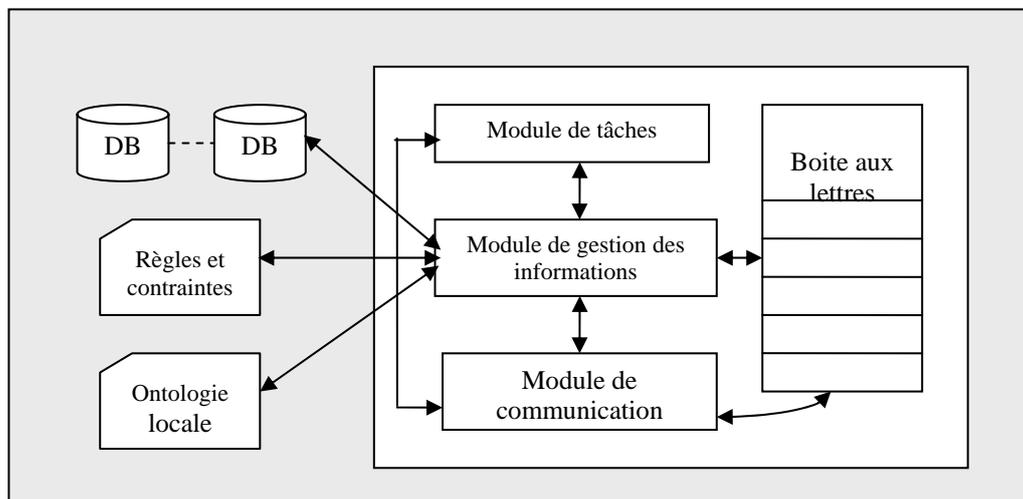


Figure IV.3. Structure d'un agent d'application

Généralement les agents d'application d'un même groupe sont conçus pour offrir des services de la même catégorie présentée par le module des tâches. Pour cette raison, un agent d'application doit travailler avec les autres agents d'application de son groupe d'une manière coopérative plutôt que d'une manière compétitive sous la surveillance de son agent coordinateur. Pour obtenir un meilleur fonctionnement, la structure d'agent d'application est formé de :

4.1.1.1. Module de communication

Il représente un canal ou un support des messages entre un agent et son coordinateur. Il contient tous les protocoles et les mécanismes nécessaires pour émettre et recevoir des requêtes. Toutes les interactions entre l'agent d'application et son environnement passent par ce module.

4.1.1.2. Boîte aux lettres

Quand les messages arrivent alors que l'agent est dans un état d'occupation, le module de communication dépose ces messages dans cette boîte aux lettres. Cette boîte est une file d'attente de type FIFO (First In First Out) utilisée pour stocker les messages afin de les traiter de façon asynchrone.

4.1.1.3. Module de gestion des informations

Il assure la gestion des informations nécessaires pour le travail de l'agent. Ces informations recouvrent essentiellement l'ensemble des règles et contraintes qui contrôlent le fonctionnement de l'agent ainsi que des ressources informationnelles additionnelle (variable, bases de données, fichiers. Etc.). De plus et pour assurer plus de précision et un meilleur fonctionnement, chaque agent d'application utilise une ontologie locale propre à lui pour représenter sa vision, son vocabulaire et ses concepts dans son domaine d'application (figure IV.4).

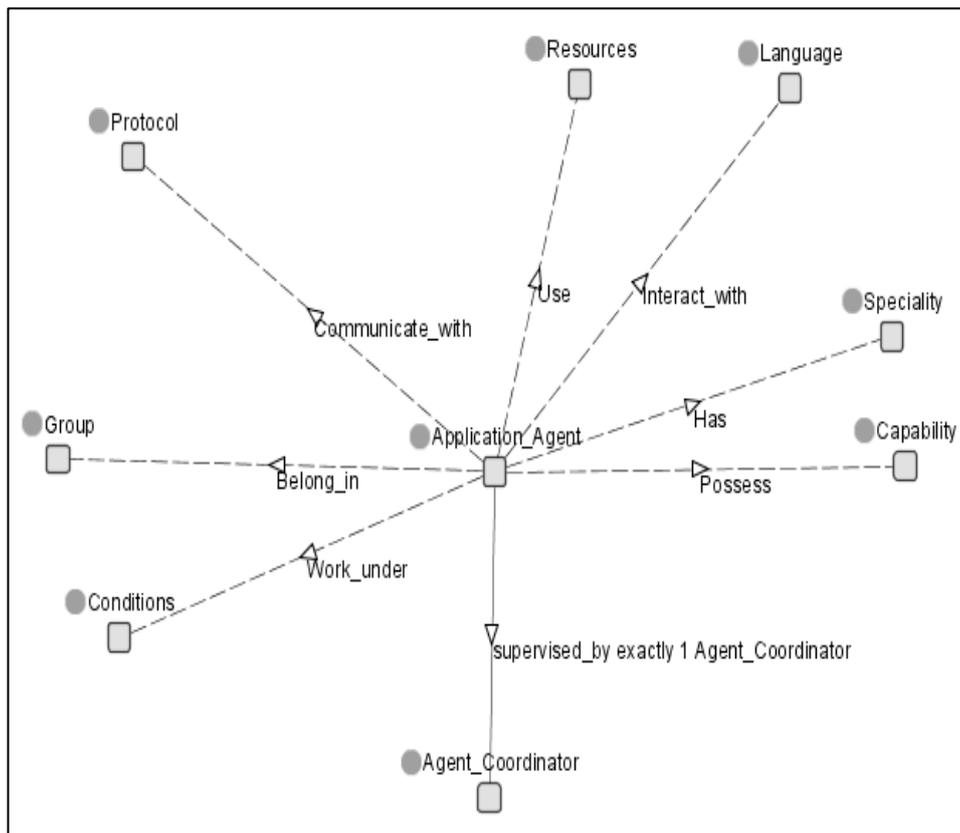


Figure IV.4. Ontologie des agents d'application

4.1.2. Agent coordinateur

L'agent coordinateur constitue un organe central au sein de son groupe. Il s'occupe de tous les besoins de leurs agents subordonnés (voire figure IV.5).

Autrement dit, un agent coordinateur maintient une base des bases de connaissances qui concernent les capacités des agents de son groupe et même des autres groupes. Il utilise ces connaissances pour assister les agents dans leurs tâches d'échange et de coopération. C'est un matchmaker entre tous les membres d'un même groupe et entre tous les groupes de l'application. Pour bien assurer cette assistance, un agent coordinateur doit se composer de :

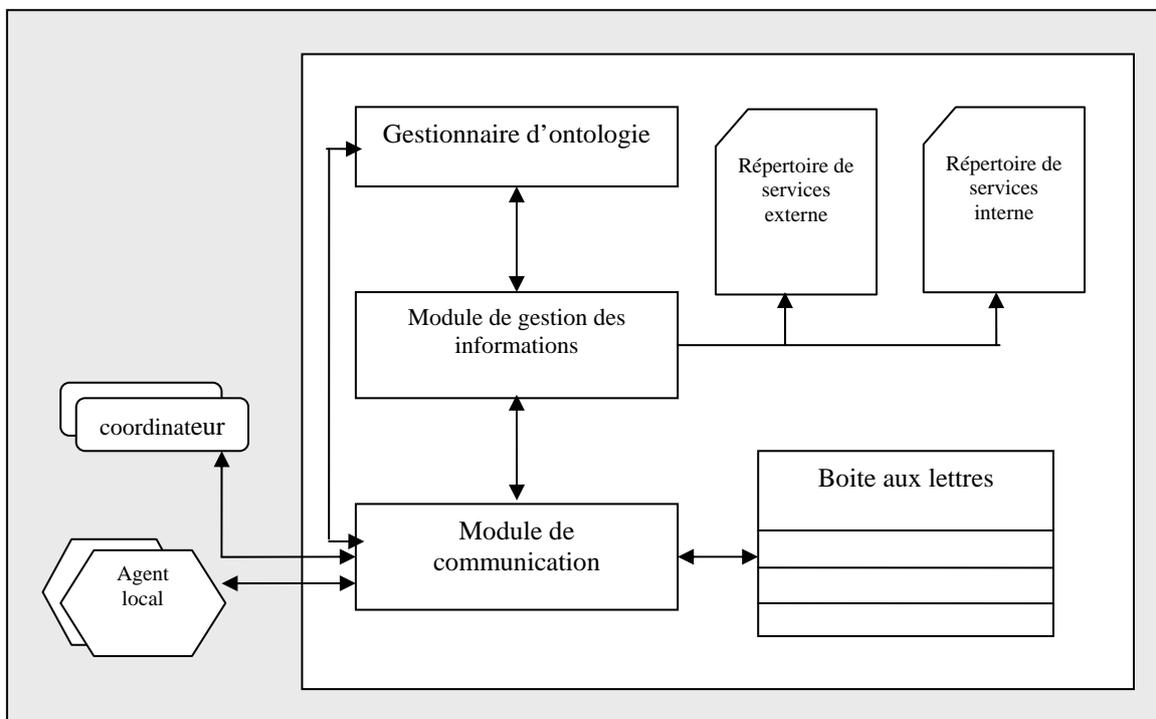


Figure IV.5. Structure d'un agent coordinateur

4.1.2.1. Module de communication

Il assure le même rôle que le module de communication de l'agent d'application. Il permet à l'agent coordinateur d'établir des relations avec les agents d'application de son groupe et avec les autres agents coordinateurs des autres groupes.

4.1.2.2. Boîte aux lettres

Elle est utilisée pour déposer des messages. Cette boîte est une file d'attente de type FIFO (First In First Out) permettant le stockage de messages afin de les traiter de façon asynchrone.

4.1.2.3. Répertoires de services

Un agent coordinateur se dispose de deux répertoires de services. Le premier est consacré pour les services et les performances des agents locaux à l'intérieur d'un groupe (interne). Le deuxième est consacré pour les services et les performances des autres groupes (externe). Ces répertoires permettent à un coordinateur de trouver les fournisseurs de services afin de les lier avec les agents demandeurs. La construction de ces deux répertoires doit être faite d'une façon incrémentale et pendant l'exécution des services coopératifs. Après la découverte du pair adéquat pour une tâche coopérative spécifique en utilisant le gestionnaire d'ontologie (ontology mapper), le coordinateur met cette correspondance dans le répertoire interne (pour les pairs du même groupe) ou dans le répertoire externe (pour des pairs avec des groupes différents) pour les utiliser directement dans le futur et pour des cas similaires.

4.1.2.4. Gestionnaire d'ontologie

Ce module s'occupe de la recherche des correspondances entre les agents afin d'exécuter une tâche coopérative. Il cherche les mêmes termes ainsi que les relations comme "est", "est-un", "partie de", et "se compose de", pour trouver le pair d'agent adéquat pour une tâche coopérative.

4.1.3. Protocole d'interaction & de coopération

Notre modèle de coopération est basé sur le protocole FIPA-Contract-Net (FIPA-Reseau-Contractuel) (FIPA, 2002a). Ce protocole est l'un des modèles de coopération dans les SMA les plus utilisés et le plus complets à ce jour. Il consiste en un contrat élaboré entre deux participants : Le contractant et le gestionnaire.

Le contractant est garant de l'exécution de la tâche et de la transmission de ses résultats au gestionnaire. Le gestionnaire, quant à lui se porte garant de la gestion de la tâche et du traitement des résultats. La figure IV.6 présente le protocole FIPA-Contract_Net. Le gestionnaire fait une requête d'action; chaque contractant peut refuser ou

faire une proposition (propose (p2)) en réponse; le gestionnaire peut accepter ou rejeter la proposition p2.

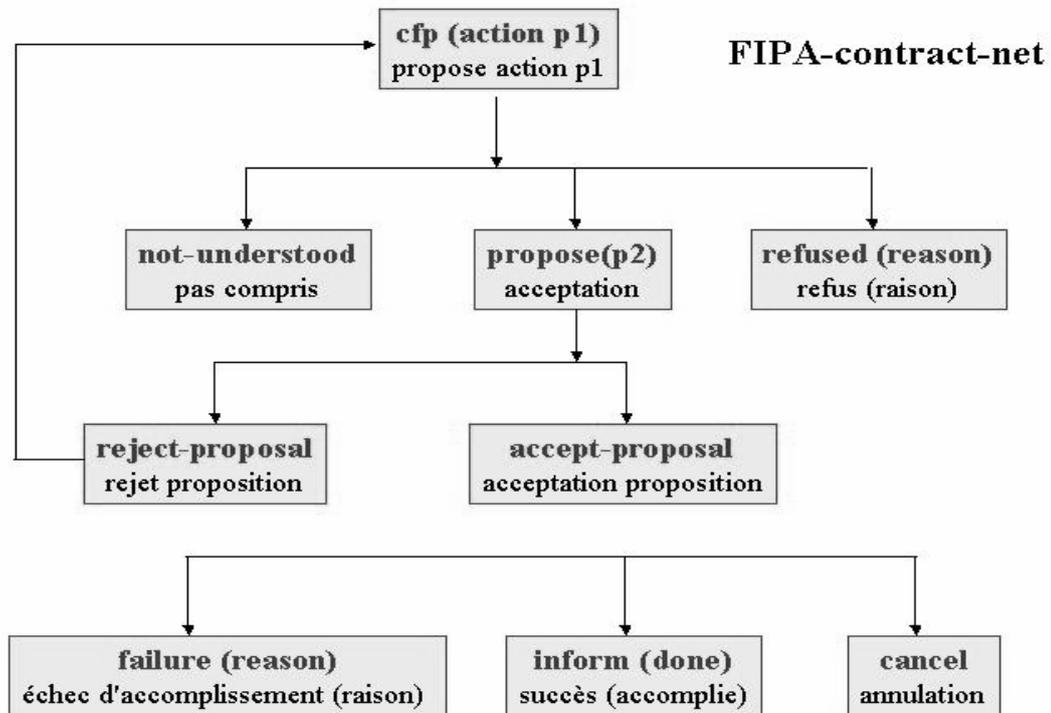


Figure IV.6. Structure d'un agent coordinateur

Un problème qui peut survenir avec le réseau contractuel est qu'une tâche puisse être attribuée à un contractant moins qualifié pour la résoudre si un contractant plus qualifié est occupé au moment de l'annonce des tâches. Le protocole réseau contractuel a été utilisé pour la résolution coopérative des problèmes, c'est à dire la résolution des problèmes par des agents coopératifs.

D'après la figure ci-dessus et d'après l'organisation des interactions que nous avons (Agent-Coordinateur-Agent) et pour avoir un protocole enrichi et complet, nous devons doter ce protocole par de nouvelles situations et échanges de messages exprimé par le langage inter agent FIPA-ACL (FIPA, 2002). Nous pouvons classer nos cas d'interactions et de coopération en deux catégories: interne (dans le même groupe) et externe (entre différents groupes). Nous mentionnons que toutes les interactions sont identifiées avec un identificateur unique pour garder la trace de cette interaction parmi le nombre élevé des interactions qui se produisent dans cette solution (Brahimi & al, 2009).

4.1.3.1. Interaction interne :

Pour une interaction interne entre deux agents du même groupe (Agent A demande un service qui peut être exécutée par Agent B). Nous distinguons deux sous-situations:

- Cette paire d'agents a déjà travaillé ensemble et déjà découvert et connu par leur coordinateur (cette information à propos de cette paire d'agents est déposée dans le répertoire des services internes du coordinateur comme (agent Requester, agent provider, Task). Dans ce cas, l'agent A passe sa requête à son coordinateur, ce dernier consulte son répertoire des services internes, il détermine et passe la requête à l'agent B. La récupération du résultat subisse le même chemin dans le sens inverse (Agent B - Coordinateur -Agent A). En utilisant le diagramme de séquence d'UML nous pouvons présenter ce scénario comme suit :

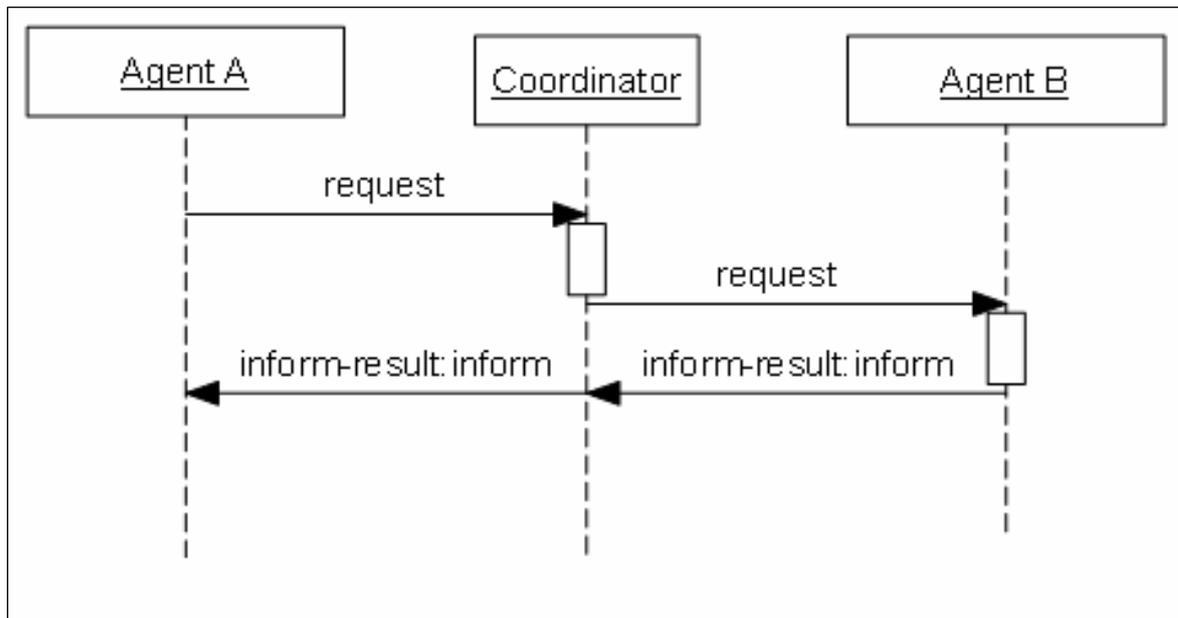


Figure IV.7. Interaction & coopération internes (Cas 1.1)

- Les agents A et B interagissent entre eux pour la première fois. L'agent A passe la requête à son coordinateur, ce dernier consulte les répertoires de services internes et externes. Si aucun agent fournisseur n'est trouvé, il analyse les ontologies de ses agents avec son module mapper afin de trouver des correspondances entre la requête et les tâches des agents de son groupe. Il propose un ensemble d'agents avec les meilleures correspondances auprès de l'agent A. Après ça, l'agent A étudie les propositions pour choisir une et confirme son choix à son coordinateur.

Ce dernier enregistre cette information dans son répertoire interne (Agent A, Agent B, Task) pour l'utiliser directement dans les futurs cas similaires. Ce scénario peut être présenté en UML comme suit:

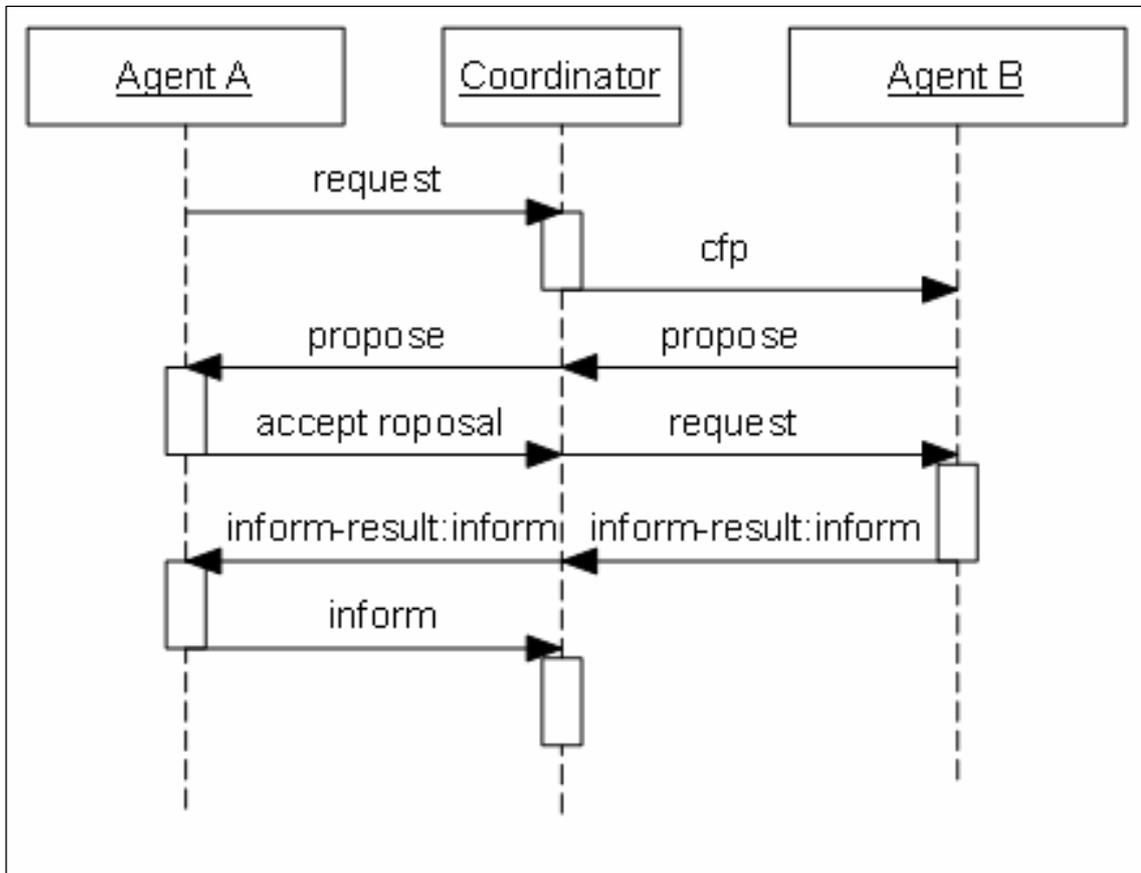


Figure IV.8. Interaction & coopération internes (Cas 1.2)

4.1.3.2. Interaction externe :

Pour une interaction externe entre deux agents de deux groupes différents (Agent A du groupe A demande un service qui peut être exécutée par Agent B du groupe B). Nous distinguons deux sous-situations:

- Cette paire d'agents a déjà travaillé ensemble et déjà découvert et connu par leur coordinateur (cette information à propos de cette paire des agents est déposée dans le répertoire des services externes du coordinateur comme (Requester agent, Provider groupe, Provider agent, Task). Dans ce cas l'agent A passe sa requête à son coordinateur, ce dernier consulte ses répertoires de services internes et externes et focalise l'agent fournisseur B qui fait partie du groupe B afin de lui

passé cette requête. La récupération de résultat subisse le même chemin dans le sens inverse (Agent B – Coordinateur B – Coordinateur A- Agent A). Avec le diagramme de séquence d’UML nous pouvons présenter ce scénario comme suit :

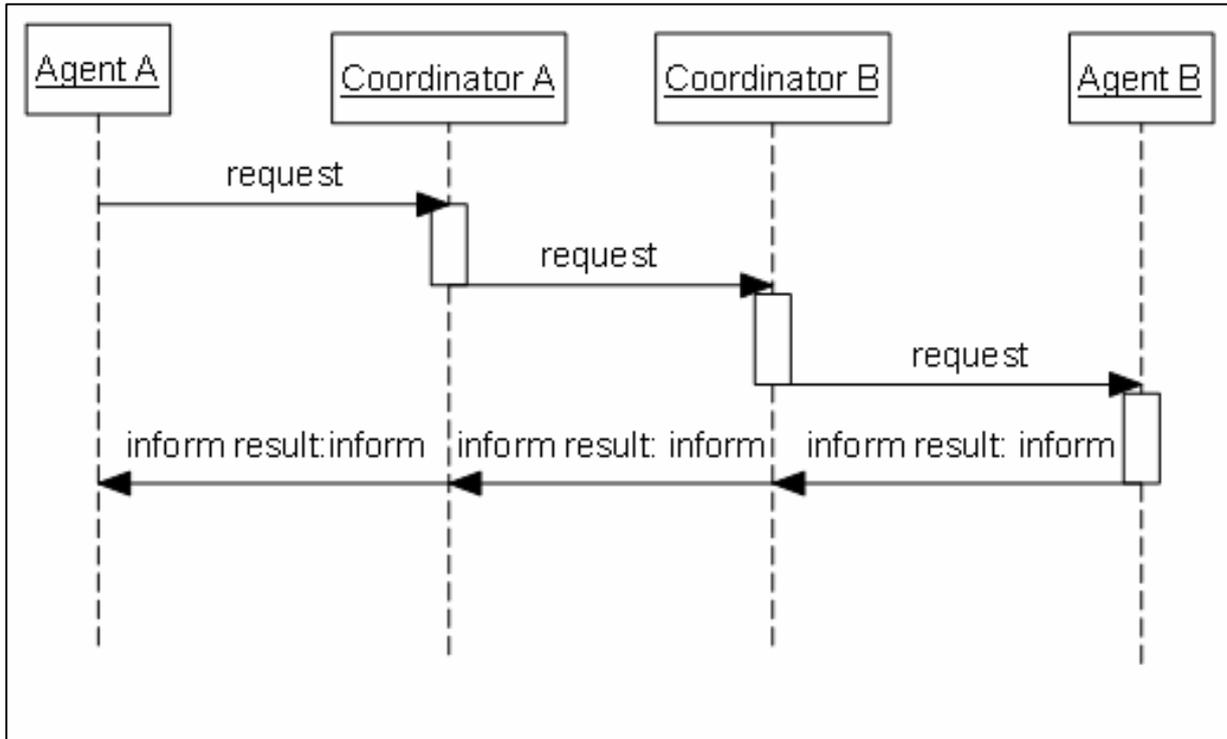


Figure IV.9. Interaction & coopération externes (Cas 2.1)

- Les agents A et B interagissent entre eux pour la première fois. Les premières phases de ce cas sont similaires à celui du (1.2), mais avec la situation qu’aucune proposition interne n’est fournie. Dans ce cas le coordinateur lance un appel de proposition (Call For Proposal) au près des autres groupes. Les coordinateurs des autres groupes vont proposer des agents fournisseurs s’ils existent des correspondances entre leurs tâches et ce qui est demandé en analysant leurs ontologies. L’agent A étudie les propositions pour choisir une et confirme son choix à son coordinateur. Ce dernier enregistre cette information dans son répertoire externe (Agent A, groupe B, Agent B, Task) pour l’utiliser directement dans les futurs cas similaires. Ce scénario peut être présenté en UML comme suit:

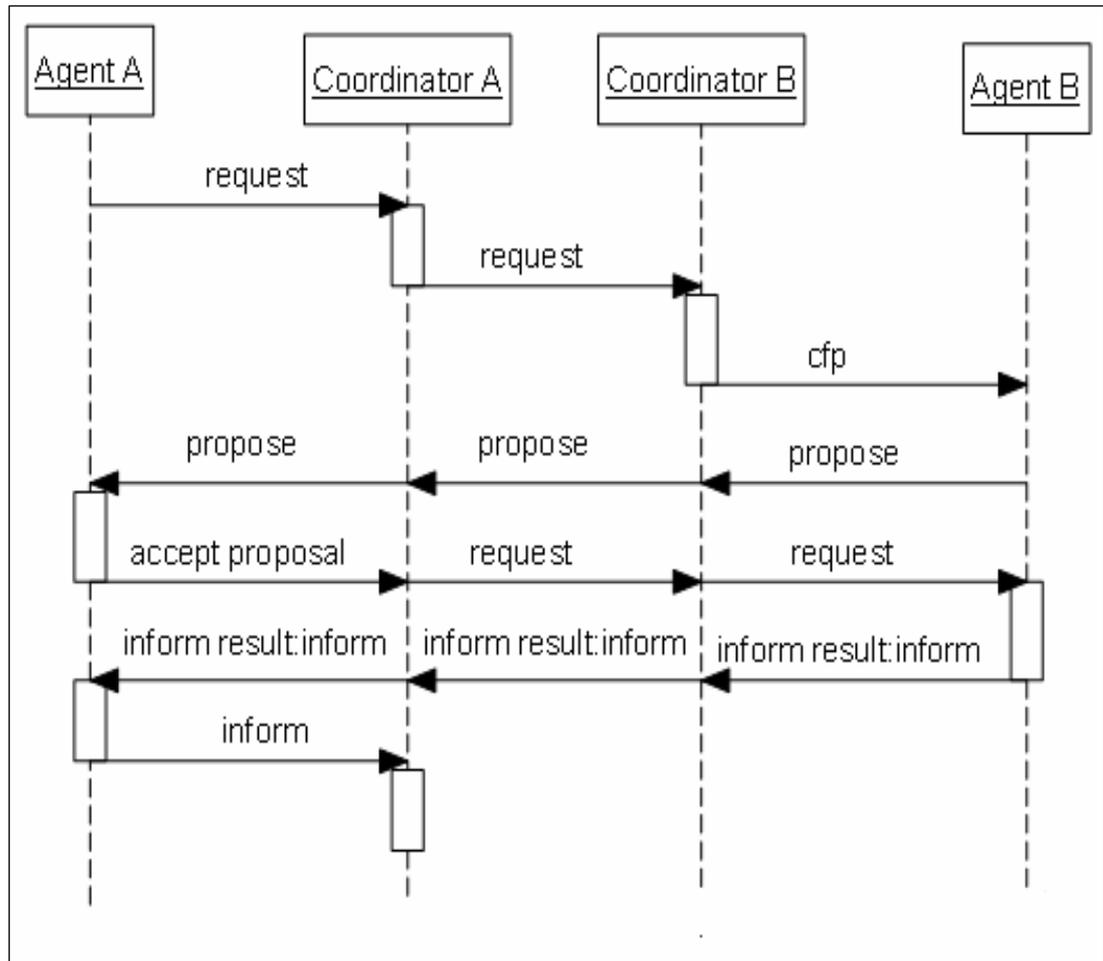


Figure IV.10. Interaction & coopération externes (Cas 2.2)

5. INTEGRATION DES WEB SERVICES

Dans cette phase, les services destinés au grand public (clients) et aux participants externes (fournisseurs et partenaires) sous forme de Web services. Nous trouvons aussi les services les plus sollicités par les agents de l'application. Ces web services vont constituer un portail Web accessible à l'intermédiaire d'un agent spécifique appelé Web Services Finder (Brahimi & al, 2006).

Pour pouvoir sélectionner et invoquer les Web services de façon adéquate et automatique, nous sollicitons le langage OWL-S pour décrire ces Web service. OWL-S est un langage qui définit une ontologie de services web (Burstein & al, 2008). Il est basé sur le langage OWL (McGuinness & al, 2003). Il permet de réaliser les deux tâches suivantes :

1) Découverte automatique de services web : Cette tâche est possible parce que OWL-S permet d'exprimer et de résoudre des requêtes avec contenu sémantique. Par

exemple une requête comme : « Trouver des services web qui réservent des chambres d'hôtels dans une ville spécifique et qui permettent de payer avec une carte de crédit particulière ».

2) Invocation automatique de services web : OWL-S fourni un ensemble d'APIs pour que l'invocation à un service web soit automatique.

OWL-S a trois parties principales (Ricardo, 2004) :

- Le profil du service : Pour faire de la publicité et découvrir des services
- Le modèle du processus qui donne une description détaillée d'une opération du service.
- Le « grounding » qui fournit les caractéristiques techniques pour établir la communication avec le service au moyen de messages.

Ces trois éléments s'occupent d'une description approfondie qui touche les points suivants :

| <i>Service Profile</i> | <i>Process Model</i> | <i>Service Grounding</i> |
|---|---|---|
| - Présenté par un service - Représenter ce que le service fournit - Usage Principales: +Publicité des capacités +Demande de Web service qui a donnée un ensemble de sa capacité | - Décrit comment ce service fonctionne: les processus internes du service - Spécifie le protocole d'interaction de service - Spécifie les messages abstraits: l'information transmis sur type d'ontologie -Faciliter l'invocation et la composition des services, la surveillance de l'interaction | - Fournit des spécifications d'information d'accès de service - Service Model+Grounding donne tous les choses nécessaire pour utiliser ce service - Construire sur WSDL pour définir la structure de message et la couche physique «binding » - Spécifier le protocole communication, mécanismes de transport, langages de communication, etc. |

Table IV.1. Composants du OWL-S service (Phan, 2005)

Cette puissance de description sera combinée avec la puissance de l'agent médiateur (Finder Agent) pour pouvoir sélectionner, invoquer et coordonner les interactions et le travail de coopération entre ces Web services et les autres parties de l'application. Nous

présentons dans ce qui suit notre agent médiateur (Web services agent finder) avec ses composants, ses mécanisme de travail ainsi que ses relations avec les autre partie de l'application.

5.1. WEB SERVICES AGENT FINDER

Cet agent interprète les spécifications OWL-S. Il est sollicité par les coordinateurs des groupes et le monde externe (clients, fournisseurs et partenaires) pour trouver les Web services adéquats a leurs besoins (voir figure IV.11).

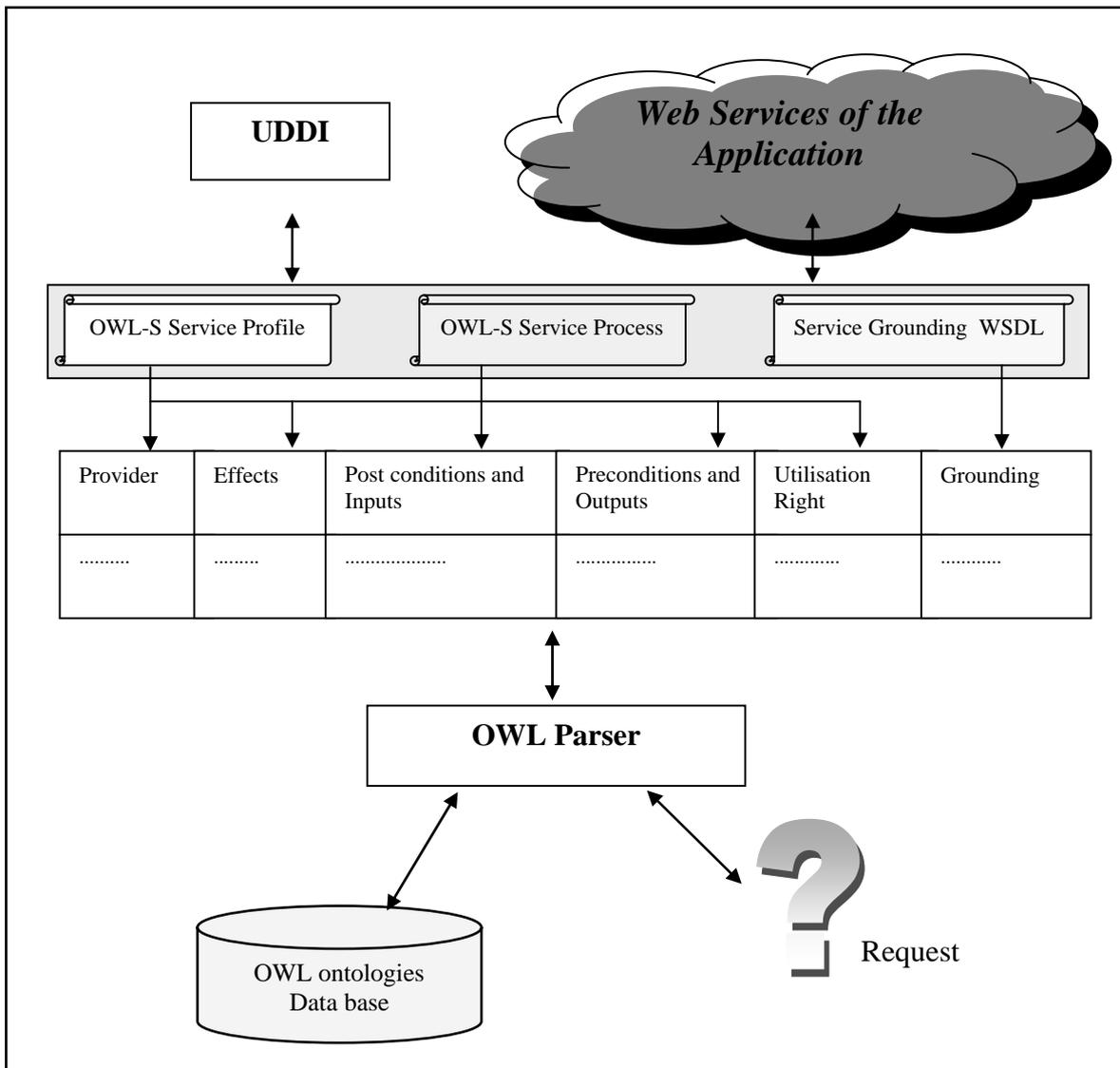


Figure IV.11. Structure de l'agent Web services Finder (Brahimi & al, 2006)

Cet agent possède les caractéristiques suivantes:

- il maintient une base de données contenant l'ensemble des ontologies concernant les Web services ainsi que le nécessaire des ontologies en rapport avec la matière de l'application. Ces ontologie permettent a cet agent finder d'accomplir ses tâches et encore d'assister les demandeur pour trouver leurs services adéquats.
- il est capable d'analyser et interpréter les spécifications OWL-S pour chercher et invoquer les services utilisés par l'application. Ce rôle est assuré par le module *OWL-S Parser*. Comme le langage OWL-S est basé a l'origine sur le format RDF et plus généralement sur la technologie XML, plusieurs techniques pour analyser et interpréter le contrenu XML (précesement RDF) sont disponibles. Nous revenons sur ce point dans le chapitre suivant consacré a l'aspect technique et l'implémentation de ce travail.
- Pour entreprendre une recherche autonome et efficace, l'agent finder maintient une table qui contient les propriétés différentes des services en rapport avec leur découverte et leur sélection telles que: fournisseur du service, droits d'utilisation, entrées, sorties, pré-conditions, post-conditions et effets . Ces propriétés sont extraites de la spécification OWL-S service profile et celle du OWL-S service model. La dernière partie de la table contient les règles d'invocation (Grounding). Le module parser extrait ces propriétés pour les comparer directement avec celle de la requete et par conséquent pour invoquer directement le service adéquat. Comme la spécification du OWL-S est caractérisée par l'insuffisance de la partie OWL-S Grounding pour l'invocation des Web service, nous proposons l'usage du WSDL qui est meilleur pour la description du niveau grounding (Messnarz & al, 2000).

5.2. CAS DE CORRESPONDANCES DES WEB SERVICES

Après la réception d'une requête, le module parser analyse et extrait les propriétés clés de la requête. Ces propriétés seront comparées avec le contenu de la table de description des services pour choisir celui le plus adéquat. Dans ce cas nous pouvons distinguer quatre niveaux de correspondance possibles :

- **Précise:** les propriétés de la requête correspondent exactement à l'un des services enregistré dans la table de description des Web services. Dans ce cas, le service sera invoqué directement grâce à sa partie *grounding* pour accomplir sa tâche demandée.
- **Large :** un ou plusieurs services correspondent largement à la requête. Ce cas se produit quand les propriétés de la requête sont incluses dans les propriétés des services correspondantes. Dans ce cas, l'agent finder choisit le service optimum contenant le minimum des propriétés satisfaisantes. Cette stratégie permet toujours d'exploiter les Web services avec le minimum de ressources.
- **Partielle:** les services trouvés correspondent partiellement à la requête. Dans ce cas, le demandeur doit être sollicité pour choisir l'un de ces services trouvés. Si le demandeur refuse tout les offres possibles, l'agent finder doit dans ce cas rediriger cette requête vers le serveur UDDI afin de trouver un service plus adéquat. Cette situation exige que l'agent finder doit se disposer de l'ensemble nécessaire des techniques qui lui permettent de communiquer avec le serveur UDDI. Si le demandeur met son accord sur l'un des services UDDI trouvé, l'agent finder enregistre ce service et met à jour sa nouvelle ontologie dans la base de données pour utiliser directement ce service dans les futures cas similaires.
- **Non adéquate:** il n'y a aucun service adéquat pour la requête. Dans ce cas, l'agent finder redirige directement la demande au près du serveur UDDI et le reste de la procédure est semblable avec celle du cas précédent.

Pour un service complexe (service composé au fil du temps en plusieurs services), chaque service simple doit contenir les post-conditions, les sorties et les effets qui sont compatibles avec les prè-conditions et les entrées du service successif. Cette composition rend possible la modelisation du workflows (business process) de façon autonome et efficace entre les composants de l'application .

6. CONCLUSION

Nous avons présenté dans ce chapitre les aspects conceptuels de notre modèle pour le développement des applications e-business coopératives. Pour répondre aux besoins de nos objectifs, ce modèle est basé essentiellement sur deux concepts clés : l'agent et le web service. La partie agents est le noyau de ce modèle, elle est représentée par un ensemble de groupes identiques. Chaque groupe possède un ensemble d'agents d'application dirigés par un agent coordinateur.

Nous avons proposé les fonctionnalités publiques ou celle destinées au monde externe comme Web services afin de libérer le noyau agent. Ces Web services sont manipulés à l'aide d'un agent médiateur appelé Agent Finder. Dans ce modèle, la notion d'agent avec toutes ses potentialités et la technologie du Web service avec sa souplesse constituent un soutien énorme pour la résolution des problèmes du domaine du e-business coopératif.

Nous passons dans le chapitre suivant à la phase d'implantation sous l'environnement DotNet tout en détaillons cette solution à l'aide d'une étude de cas réelle.

CHAPITRE



Implémentation & Etude de cas

1. INTRODUCTION

Après avoir présenté notre contribution dans le chapitre précédent, nous passons dans ce chapitre aux concepts techniques liés à l'implantation de notre solution proposée. Nous commençons par la présentation d'une brève étude comparative entre les deux plateformes J2EE de Sunsoft et DotNet de Microsoft qui sont capable d'implanter ce travail. Nous opterons pour la technologie .Net et nous allons motiver ce choix dans ce qui suit avec la présentation des outils nécessaires pour le développement.

Une étude de cas est bien présente aussi dans ce chapitre. Cette étude concerne une petite société de vente de livre et permet de mieux comprendre et cerner les éléments nécessaires de notre solution. Elle combine les activités du e-commerce et celles de l'administration.

2. OUTILS ET ASPECTS TECHNIQUES POUR L'IMPLANTATION

L'implantation d'une application e-business coopérative doit se fonder sur le choix d'une plate-forme applicative qui se présente sous la forme d'une suite logicielle comprenant l'ensemble des briques nécessaires au déploiement d'une application client/serveur de haut niveau. A savoir : une application ou plusieurs applications serveur accessibles - généralement en mode Web - depuis des postes de travail ou des terminaux Internet. Le marché informatique actuel nous offre essentiellement deux principaux modèles technologiques de plate-forme applicative. Le premier est J2EE (pour Java 2 Enterprise Edition) Lancé par Sun (Sun, 2009) en 1998, autour du langage Java, il s'articule autour d'une infrastructure standardisée, couvrant les principales couches d'une plate-forme applicative (serveur d'applications, infrastructure de portail et serveur d'intégration), ainsi que les liens avec la base de données (ou persistance). J2EE est aujourd'hui implémenté par les principaux éditeurs de serveur d'applications du marché, parmi lesquels on compte IBM, Oracle et BEA. Le deuxième est .Net qui a été lancé par Microsoft (Msdn, 2009) en 2001 en réponse à J2EE. Cette plate-forme, rebaptisée en 2003 Windows Server System, s'adosse à la manière de J2EE à une logique de développement et de déploiement de nouvelle génération (orientée objets). Son principal point fort : aussi structurante et riche fonctionnellement que son équivalent Java, elle n'en offrirait pas moins une approche de travail beaucoup plus simple.

Dans ce travail, nous opterons pour la technologie .Net pour des raisons qui se dévoilent avec la présentation des outils utilisés dans notre travail de mise en œuvre. La technologie .Net comporte trois grandes parties (Giller, 2006) :

- un ensemble extensible de langages dont C#, VB.NET, J#, Delphi#, etc. Ces langages doivent respecter la spécification Common Language Specification (CLS) du Common Language Infrastructure (CLI) ;
- un ensemble de classes de base utilisées par les applications. C'est ce que l'on appelle le framework .NET ;
- une couche logicielle nommée CLI qui est responsable de l'exécution des applications .NET.

Le framework .NET propose un API objet de très haut niveau qui permet de créer des applications complexes plus rapidement. Le .NET respecte un bon nombre de standards d'organisations comme le World Wide Web Consortium (W3C), l'Institute of Electrical and Electronics Engineers (IEEE), l'Organization for the Advancement of

Structured Information Standards (OASIS), l'Internet Engineering Task Force (IETF) et la Web Services Inspection (WSI). Tous ces critères permettent de développer avec toute convivialité la même application dans plusieurs langages: Perl, VB, C#, Eiffel, etc. ..., et de réaliser avec excellence des applications et des services Web.

Nous allons présenter dans ce qui suit l'ensemble des outils appartenant à la grande famille .Net et qu'on juge importants pour la mise en œuvre de notre solution. Ces outils seront présentés à travers les différents concepts de notre solution avec une manière qui motive notre choix au niveau de chaque aspect technique de l'application.

2.1. DEVELOPPEMENT DES AGENTS

C'est la partie la plus intéressante dans notre solution où le langage C# constitue l'élément essentiel dans le développement technique. Le C# est, d'une certaine manière, le langage de programmation qui reflète le mieux l'architecture Microsoft .NET qui fait fonctionner toutes les applications .NET, et en est par conséquent extrêmement dépendant. Les types natifs correspondent à ceux de .NET, les objets sont automatiquement nettoyés par un ramasse-miettes, et beaucoup de mécanismes comme les classes, interfaces, délégués, exceptions, ne sont que des moyens explicites d'exploiter les fonctionnalités de la bibliothèque .NET. Pour achever de marquer cette dépendance, le CLR (Common Language Runtime) est obligatoire pour exécuter des applications écrites en C#, comme l'est la JVM (Java Virtual Machine ou Machine virtuelle Java) pour des applications Java.

D'après l'expérience des développeurs c'est un langage de programmation orienté objet très proche du langage Java dont il reprend la syntaxe générale ainsi que les concepts (la syntaxe reste cependant relativement semblable à celles de langages tels que le C++ et le C). La ressemblance du C# avec le langage Java nous donne une très bonne possibilité de développer nos agents avec les mêmes performances offertes par le langage Java (Grosso & al, 2003).

2.2. DEVELOPPEMENT DES WEB SERVICES

Dans cette partie de développement, l'ASP.NET constitue naturellement notre choix trivial. C'est un ensemble de technologies de programmation web créé par Microsoft. Les programmeurs peuvent utiliser ASP.NET pour créer des sites web dynamiques, des applications Web ou des Web Services XML. La technologie est accessible grâce à l'installation d'un serveur web compatible ASP (IIS) ou à l'intérieur de Visual Web Developer Express Edition .

Bien qu'ASP.NET tienne son nom de l'ancienne technologie de développement de Microsoft, "ASP", les deux sont assez différentes. Microsoft a complètement repensé ASP.NET, en se basant sur le Common Language Runtime (CLR) partagé par tous les logiciels Microsoft .NET. Les programmeurs peuvent écrire du code ASP.NET en utilisant n'importe lequel des langages de programmation supportés par le Framework .NET, généralement C#, Visual Basic.NET, Delphi.NET ou JScript .NET, mais aussi d'autres langages "indépendants" de Microsoft tels que Perl et Python. ASP.NET bénéficie de meilleures performances que d'autres technologies basées sur des scripts car le code côté serveur est compilé en quelques simples DLL sur le serveur web. Lors du développement, quand le code source est finalisé, la solution est precompilée avant d'être placée sur le serveur d'hébergement (publication) (www.ASP.net).

ASP.NET permet aux développeurs de passer plus facilement du développement classique d'applications Windows au développement d'applications Web en offrant la possibilité de créer des pages web composées de Widget (ou zone de contrôle) similaires à celles des interfaces d'applications Windows habituelles.

2.3. DEVELOPPEMENT, PARSING & INTEROGATION DES ONTOLOGIES

Dans notre solution, nous distinguons le développement et l'utilisation des ontologies pour deux raisons qui sont:

1. Représentation des connaissances avec le langage OWL (voir chapitre 3) où le terme ontologie fait référence à un module constitué d'un vocabulaire spécifique utilisé pour décrire un modèle d'un domaine particulier du monde réel, vocabulaire auquel est associé un ensemble d'hypothèses ou assertions. Le vocabulaire et les assertions qui composent l'ontologie sont censés apporter une aide aux « agents », pour résoudre leurs problèmes et partager leurs connaissances.
2. Description des Web services à travers le langage OWL-S (voir chapitre 3). Cette description permet de décrire notamment :
 - le fournisseur du Web Service;
 - les informations que ce dernier peut donner ;
 - le format des requêtes et
 - où trouver ce Web Service.

Nous mentionnons que pour les deux raisons citées précédemment nous utilisons les mêmes outils pour développer ces ontologies (Editeur Protégé-2000), pour les parser (SemWeb) et pour les interroger (Langage Sparql) :

2.3.1. Protégé-2000 Version 3.4

Protégé-2000 (Protégé, 2008) est un éditeur qui permet de construire une ontologie pour un domaine donné, de définir des formulaires d'entrée de données, et d'acquérir des données à l'aide de ces formulaires sous forme d'instances de cette ontologie. Dans le contexte du web sémantique des « plugin » pour les langages RDF, DAML+OIL, OWL et OWL-S ont été développés pour Protégé. Ces « plugin » permettent d'utiliser Protégé comme éditeur d'ontologies de domaines et pour décrire nos Web Services.

2.3.2. SemWeb Version 1.062

SemWeb est une bibliothèque développée par Joshua Tauberer (Joshua, 2008) sous les termes de la licence GNU GPL (version 2 ou plus). Elle est écrite en C# et destinée pour la plate-forme DotNet. Cette bibliothèque permet de lire et d'écrire RDF (XML, N3), garder RDF en stockage persistant (mémoire, MySQL, etc.). Elle permet également l'utilisation d'un simple graphe de matching et le langage SPARQL engine comme composant externe (spareql-core.dll) pour l'interrogation d'un contenu RDF.

2.3.3. Langage d'interrogation SPARQL

SPARQL (Standard Protocol And RDF Query Language) est un API Java. C'est langage de requête, devenu le 15 Janvier 2008, dans le cadre de l'activité Web sémantique du W3C, une recommandation W3C (W3C, 2008). Sparql n'utilise pas le langage naturel et constitue l'un des piliers du Web sémantique. Le langage Sparql définit la syntaxe et la sémantique nécessaire à l'expression de requêtes sur une base de données de type RDF et la forme possible des résultats. SPARQL est adapté à la structure spécifique des graphes RDF, et s'appuie sur les triplets qui les constituent. En cela, il est différent du classique SQL (langage de requête qui est adapté aux bases de données de type relationnelles), mais s'en inspire clairement dans sa syntaxe et ses fonctionnalités. Il a aussi quelques traits de ressemblance mineurs avec Prolog.

2.3.4. IKVM Version 0.38

Pour pouvoir utiliser l'API Sparql développé sous Java au sein de notre solution développée sous .NET. Le composant IKVM développé par Jeroen Frijters permet de nous assurer cette interopérabilité (IKVM, 2008). IKVM est un freeware qui inclut les composants suivants :

- Une Machine Virtuelle Java (JVM) implantée en .NET ;
- Une implantation .NET pour les bibliothèques de classe Java ;
- Un ensemble d'outils permettant l'interopérabilité entre Java et .NET.

Avec IKVM, nous pouvons exécuter directement des codes Java compilés (Byte Code) dans des applications .NET. Le byte code Java est convertis en CLI du .NET et exécuté par la suite.

3. ETUDE DE CAS

Nous allons décrire dans ce qui suit comment utiliser de point de vue pratique notre solution proposée. Une étude de cas va nous permettre de mieux comprendre et de mieux cerner tout les éléments intervenant dans cette solution. Cette étude de cas s'inscrit essentiellement dans le cadre du commerce électronique qui représente l'activité la plus exercée dans le business électronique avec des fonctions qui font parties de l'administration électronique.

3.1. PROBLEMATIQUE

L'entreprise « *Mon-Livre* », est une société de vente de livres de toutes catégories. Nous voulons mettre en place une stratégie pour avoir un système très sophistiqué où les responsables de la société peuvent vendre sur Internet, et ainsi augmenter le taux de l'opération d'échange de produits, d'informations, et des services avec les clients, les fournisseurs (maisons d'édition), et les partenaires de la société. Nous allons montrer comment notre solution va apporter de l'aide aux responsables de la société pour instaurer leur nouveau système avec toutes les ambitions et les services attendus.

3.2. DEVELOPPEMENT DE L'APPLICATION

Au début, il convient de disposer d'un modèle multi-agents sur lequel repose le système « Mon-Livre ». Les agents doivent être répartis en plusieurs groupes (plusieurs groupes d'application similaires) encapsulant la même structure interne et les mêmes mécanismes de travail. Pour cela nous devons passer par les étapes suivantes :

1. Chercher et allouer les tâches aux agents individuels.
2. Selon les tâches spécifiées, décomposer le système en un ensemble de groupes d'agents.
3. Spécifier les fonctions qui seront offertes comme des Web Services.

Réellement, dans notre étude de cas et pour des raisons de simplification, nous n'avons considéré que les fonctions qui seront au service des clients et des administrateurs de la société. Le diagramme de contexte suivant (figure V.1) détermine les fonctions fournies par notre étude de cas pour les administrateurs et pour les clients :

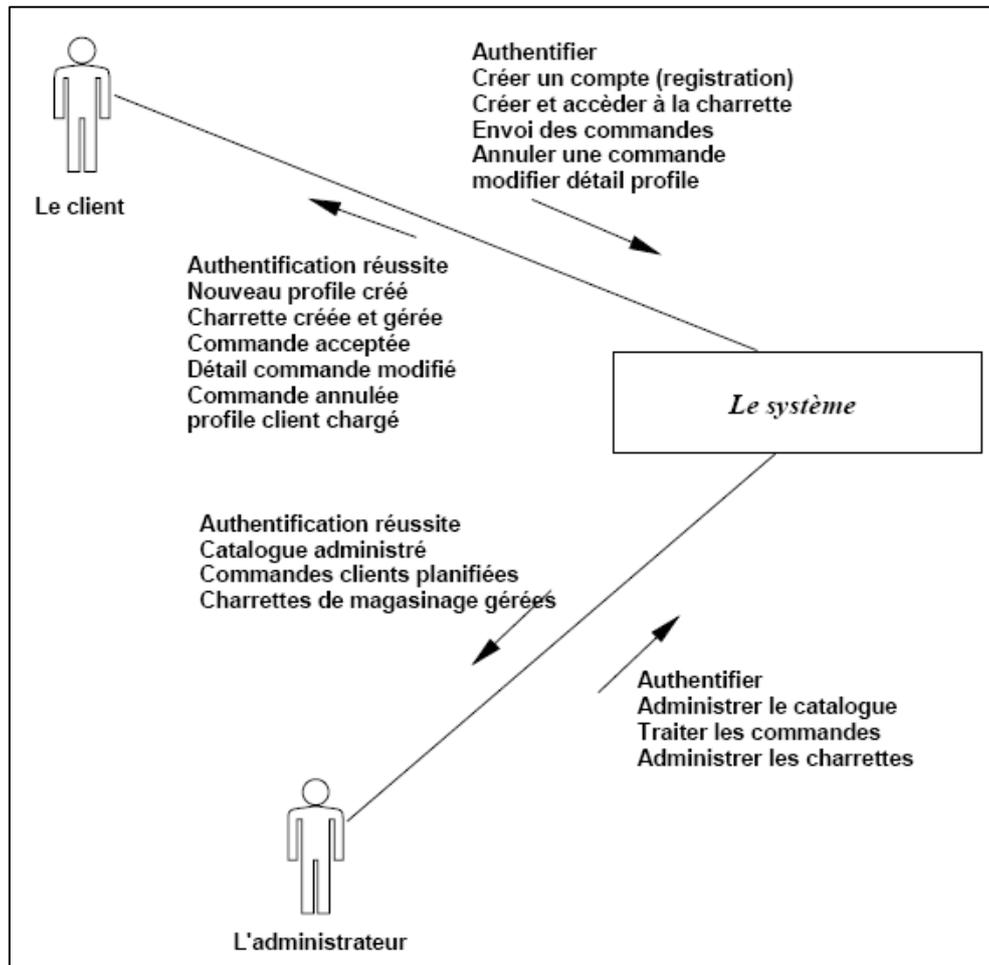


Figure V.1 : Modélisation du contexte pour le système « Mon-Livre »

Après définition des grands objectifs du système, nous décomposons ces objectifs en un ensemble de tâches avec les enchaînements des événements qui décrivent le comportement du système. Cette étape s'achève à un ensemble de tâches avec les itinéraires de communication. Au premier lieu ces tâches peuvent être comme suit :

- i. **Naviguer dans le catalogue** : gérer une fonctionnalité permettant aux visiteurs de naviguer dans la liste des livres selon des catalogues spécifiques.
- ii. **Localiser un livre** : gérer une fonctionnalité permettant aux visiteurs (clients) de localiser un livre du catalogue.

- iii. **Administrer le catalogue** : gérer une fonction d'administration et de gestion des informations du catalogue, afin de rendre possible l'ajout, la modification, et la suppression des livres et des catégories de livres.
- iv. **Administrer les charrettes de magasinage** : gérer une fonction permettant à l'administrateur de manipuler les charrettes des clients, comme la suppression des anciens éléments pour ne pas laisser les charrettes agrandis.
- v. **Administrer les commandes des clients** : gérer une fonction de planification et d'exécution des commandes des clients et créer un mécanisme pour calendrier les ordres.
- vi. **Ajouter des comptes utilisateurs** : gérer une fonction de création et d'ajout de comptes utilisateurs afin de fidéliser les clients.
- vii. **Charger un profile client** : gérer une fonction d'authentification et chargement de profile d'un client, afin de lui permettre de modifier ses informations personnelles et de suivre ses commandes.
- viii. **Accéder à la charrette de magasinage** : permettre au client de créer sa propre charrette de magasinage, afin de le permettre sélectionner et enregistrer des livres souhaités avant de les commander.
- ix. **Ajouter des commandes** : il doit exister un mécanisme permettant aux visiteurs (clients) de placer des commandes.
- x. **Mettre à jour, annuler des commandes** : gérer une fonction d'annulation ou modification des détails d'une commande d'un client.
- xi. **Recevoir les paiements** : gérer une fonction de réception de paiements des clients.
- xii. **Authentifier** : gérer une fonction d'authentification des utilisateurs.

En fin, nous allons créer les groupes d'agents avec leurs itinéraires de communication la création doit respecter la spécialité des tâches pour que les agents de chaque groupe soient homogènes. Les résultats de distribution en groupes s'expliquent par la figure ci-dessous.

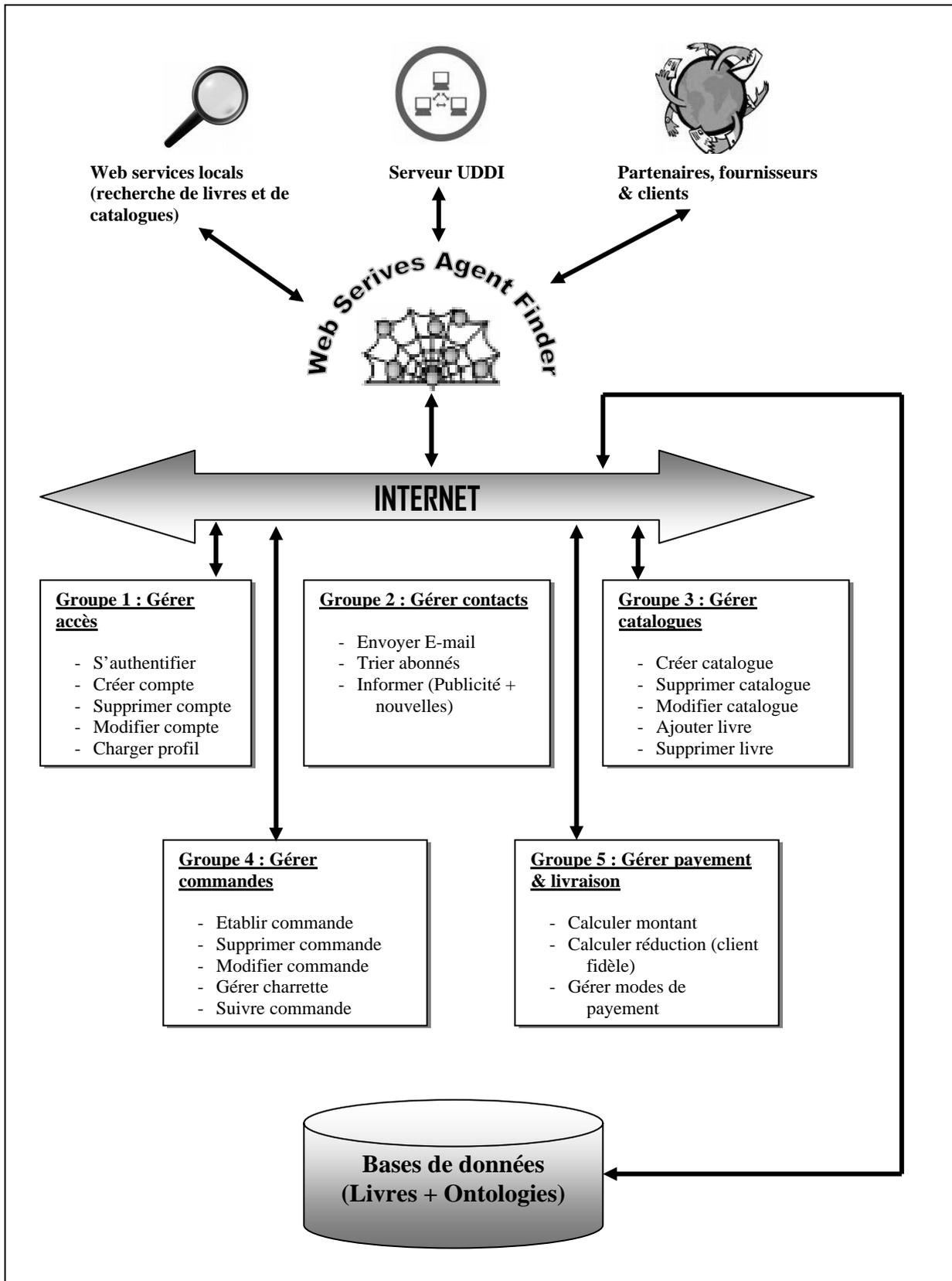


Figure V.2 : Groupes d'application du système « Mon-Livre »

3.3. CRITERES POUR L'IMPLANTATION

Nous présentons dans cette section quelques guidelines techniques pour la mise en oeuvre de cette application. Nous considérons à ce niveau les possibilités de programmation offerte par la plate-forme .Net.

3.3.1. Coté Agents

Le développement des agents se rend pratiquement au développement des objets. Pour cela le langage C# et par sa nature purement objet très similaire a son concurrent Java, offre de nombreux moyens efficaces pour définir des classes, tels que fournir différents niveaux d'accès, hériter des fonctionnalités d'autres classes et permettre au programmeur de profiter de la richesse de classes offerte par le .Net.

3.3.2. Coté Web Services

Les Web services utilisés dans notre application doivent être décrits par le langage OWL-S. Dans notre étude de cas nous avons considéré un Web service concernant la navigation dans les catalogues et les listes des livres. Ce service nommé « SearchBook », peut avoir la description OWL-S suivante :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
xmlns:list="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#"
xmlns:swrl="http://www.w3.org/2003/11/swrl#"
xmlns:time="http://www.isi.edu/~pan/damltime/time-entry.owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns="http://www.owl-
ontologies.com/unnamed.owl#" xmlns:expr="http://www.daml.org/services/owl-
s/1.1/generic/Expression.owl#" xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
xmlns:service="http://www.daml.org/services/owl-s/1.1/Service.owl#"
xmlns:grounding="http://www.daml.org/services/owl-s/1.1/Grounding.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
xml:base="http://www.owl-ontologies.com/unnamed.owl">
<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Grounding.owl"/>
<owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrlb.owl"/>
<owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrl.owl"/>
<owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl"/>
</owl:Ontology>
<process:Output rdf:ID="Price">
```

```

<process:parameterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.w3.org/2001/X
MLSchema#float</process:parameterType>
</process:Output>
<process:Input rdf:ID="Title">
<process:parameterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.w3.org/2001/X
MLSchema#string</process:parameterType>
</process:Input>
<process:Output rdf:ID="Found_Title">
<process:parameterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.w3.org/2001/X
MLSchema#string</process:parameterType>
</process:Output>
<process:AtomicProcess rdf:ID="search_Book">
<process:hasOutput rdf:resource="#Found_Title"/>
<process:hasOutput>
<process:Output rdf:ID="Authors">
<process:parameterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.w3.org/2001/X
MLSchema#string</process:parameterType>
</process:Output>
</process:hasOutput>
<process:hasInput rdf:resource="#Title"/>
<process:hasOutput rdf:resource="#Price"/>
</process:AtomicProcess>
<grounding:WsdGrounding rdf:ID="Search_Book_WsdlGrok_unding">
<service:supportedBy>
<service:Service rdf:ID="Find_Book">
<service:supports rdf:resource="#Search_Book_WsdlGrok_unding"/>
</service:Service>
</service:supportedBy>
</grounding:WsdGrounding>
<grounding:WsdAtomicProcessGrounding
rdf:ID="WsdAtomicProcessGrounding_10"/>
</rdf:RDF>

```

En plus de ça, le noyau de ce service qui effectue réellement la recherche à travers les critères fournis par l'utilisateur doit être développé par langage ASP.NET. On considère à ce niveau les services Web XML qui offrent aux applications la possibilité de communiquer dans un environnement faiblement couplé en utilisant des échanges de messages prédéfinis reposant sur des protocoles standard, tels que HTTP, XML, XSD, SOAP et WSDL.

Les services Web XML peuvent être utilisés pour permettre aux applications de communiquer, qu'elles résident sur le même ordinateur ou non, même si elles résident sur

des plates-formes informatiques ou des périphériques différents. En plus, nous pouvons utiliser n'importe quel langage de programmation .NET au sein des Web services développés sous ASP.net. Par exemple notre Web service « SearchBook » pourra être développé sous Asp.net avec le langage C# comme suit :

```
<%@ WebService Language="C#" Class="SearchBook" %>
```

```
using System;
using System.Web.Services;

public class SearchBook : WebService {

    [WebMethod] public String Book() {
        // The code of the SearchBook Web Service
    }
}
```

Pour la recherche des Web services au niveau du répertoire UDDI, l'utilisation du SDK UDDI de Microsoft et notamment la classe *FindService* sera très utile. Dans notre cas, si le service « SearchBook » ne renvoi pas des résultats satisfaisants, nous lançons une recherche au niveau de l'UDDI par le code C# suivant :

```
using System;
using Microsoft.Uddi;
using Microsoft.Uddi.Services;
public class FindMyService
{
    public static void Main(string [] args)
    {
        try
        {
            // Create a connection to the UDDI server that is to be accessed.
            UddiConnection myConn = new
            UddiConnection("http://test.uddi.microsoft.com/inquire");

            // Create an object to find a business.
            FindService fs = new FindService("Bibliography");

            Console.WriteLine(fs.ToString());

            // Send the FindBusiness request over the connection.
            ServiceList servList = fs.Send(myConn);

            // Report the service name and unique identifying key.
            foreach (ServiceInfo servInfo in servList.ServiceInfos)
            {
                Console.WriteLine("Service: {0} ({1})", servInfo.Names[0].Text,
                servInfo.ServiceKey);
            }
        }
    }
}
```

```
    }  
  }  
  catch (Microsoft.Uddi.UddiException e)  
  {  
    Console.WriteLine("UDDI error: " + e.Message);  
  }  
  catch (Exception gen)
```

3.3.3. Coté interaction

Toutes les interactions dans notre solution (messages entre agent, interrogation sparql, requêtes et résultats des requêtes) sont présentées sous forme de flux XML. De ce fait, on a besoin d'un outil qui permet d'écrire et lire des fichiers XML. Les classes *XmlTextReader* et *XmlTextWriter* du langage C# permettent de lire ou écrire, nœud après nœud un document au format XML. L'invocation de ces classes se fait à la déclaration suivante :

```
using System.Xml;  
using System.IO;
```

Le parcours du document se fait du début vers la fin et même pour des documents dans la taille est potentiellement très grande.

4. CONCLUSION

Nous avons présenté dans ce chapitre l'ensemble des outils et des techniques utilisés dans le travail d'implantation. L'adoption de la plate-forme DotNet constitue un très bon choix vu les performances offertes par cette dernière.

Nous avons présenté aussi à travers une étude de cas l'ensemble des moyens conceptuels et techniques qui peuvent servir à la bonne compréhension de notre approche. Cette étude de cas ne démontre qu'une catégorie partielle d'applications qui peuvent être développée au-dessus de notre approche.

CONCLUSION GENERALE

Ce travail s'inscrit dans le domaine du e-business coopératif. Ce domaine est devenu de plus en plus important et indispensable. Il commence à sortir de son stade d'enfance pour entrer dans celui de la maturité. Il propose une nouvelle vision de l'informatique couplée par une autre de l'économie. Généralement, les applications de l'e-business coopérative sont caractérisées par leur complexité et leur difficulté de mise en œuvre vu l'entrelacement d'un nombre important de notions et de besoins conceptuels, techniques et de gestion.

Nous avons constaté que le couplage Agent-Web service dans le développement des applications e-business coopératives permet de prendre en compte non seulement la dimension technique mais aussi les considérations liées aux nouveaux besoins du marché. Cela est dû d'une part à la puissance du paradigme agent sur le coté intelligence (autonomie et communication) et d'autre part à la souplesse et l'efficacité du paradigme Web service qui vient de s'imposer comme un middleware universel.

Dans ce qui suit, nous allons résumer notre contribution apportée, les limitations de cette contribution, et par conséquent suggérer quelques perspectives et préconiser quelques aspects pour les travaux futurs.

Dans l'optique du e-business coopératif, nous avons proposé dans cette thèse un modèle à base d'agents avec l'intégration des Web services pour le développement des applications e-business coopératives. Ce modèle assure la coordination et l'échange d'informations entre les différentes activités de l'application, tout en respectant leur autonomie et leur distribution.

En plus de notre état de l'art qui a été consacré à la synthétisation de la nouvelle économie, le domaine du e-business coopérative, les systèmes multi-agents et les Web services tout en mentionnant la nécessité de coupler ces deux derniers paradigmes. Nous avons apporté dans notre thèse une contribution qui peut être résumée comme suit :

- Nous avons proposé un modèle noyau purement basé sur la technologie agent. L'idée principale avec laquelle on modélise ce noyau est de partager les différentes tâches

d'un système (une entreprise) sur plusieurs groupes appelés Groupes d'Application supervisé par de agent coordinateurs. Ces groupes d'applications sont généralement distribués sur plusieurs sites d'exécution. Ils subissent la même structure et les mêmes mécanismes de travail et interagissent entre eux via leurs coordinateurs tout en étant accessibles par le monde extérieur.

- Les fonctionnalités destinées au grand public (clients) et aux participants externes (fournisseurs et partenaires) ou même les services les plus sollicités par les agents de l'application sont représenté sous forme de Web services. Ces web services vont constituer un portail Web accessible à l'intermédiaire d'un agent spécifique appelé Web Services Finder Agent. Pour pouvoir sélectionner et invoquer les Web services de façon adéquate et automatique, nous sollicitons le langage OWL-S pour décrire ces Web services.
- Nous avons validé notre contribution par une étude de cas avec présentation des différents outils et aspects techniques utilisés dans l'implantation de notre solution. Le langage C# de la technologie .NET est l'élément principal pour le développement des agents. Le langage ASP.NET permet d'avoir des Web service de haute qualité. Finalement le développement et la manipulation des ontologies ont été effectués par SemWeb et le langage d'interrogation Sparql. Nous avons utilisé cette panoplie de techniques pour développer une étude de cas concernant la gestion et la vente des livres dans la société « Mon-Livre ». Cette étude de cas a permis de mieux montrer le fonctionnement de notre solution.

Le travail réalisé dans cette thèse constitue un champ ouvert pour toutes participations et continuations. Pour l'amélioration de ce travail nous pouvons envisager quelques perspectives. Nous pensons en particulier à :

- l'amélioration conceptuelle du modèle en raffinant le cas d'utilisation des Web services composites sous forme de séquence BP (Business Process).
- L'introduction de plus de formel dans le modèle proposé pour bien contrôler le comportement des agents et des Web services. La rigueur du formel s'ajoute à la rigueur de la distribution des agents en groupes avec la supervision de leurs agents coordinateurs.

- Le raffinement des concepts du Web sémantique pour pouvoir introduire plus d'autonomie et plus d'intelligence au niveau des Web services et au niveau de notre agent Finder.
- La transformation et l'implantation de ce travail sous la forme d'un framework afin de l'utiliser comme un moule pour générer des applications e-business coopératives. La notion de Framework facilite la tâche d'instauration de cette solution dans n'importe quelle entreprise.
- Finalement le développement des agents de ce modèle sous forme de Web service en combinant l'aspect structurel des Web services avec l'aspect comportemental des agents.

REFERENCES BIBLIOGRAPHIQUES

- (Adam et Kolki, 1999). Adam. E et Kolki. C. (1999). « Étude comparative de méthodes de génie logiciel en vue du développement de processus administratifs complexes », 1999.
- (AFCET/AFIA, 1996). Groupe de travail Interaction IAD & SMA – AFCET/AFIA «La Co-Détermination Agent-Collectif Par l’Interaction », 1996.
- (Alloui & al, 2002). Alloui. I, Jean Marc. A, Olivier. B, Mihnea. B, Stefania. C et Karim. M. « e-Alliance: a software infrastructure for concurrent inter-organisational alliances », ISPE Concurrent Engineering (CE)’02, Cranfield University, UK. pp. 27-31. July, 2002
- (Amal & al, 2004). Amal. E, Serge. H, Tarak. M et Alexandru. S. « Interopérabilité des Systèmes Multi-Agent en utilisant des Services Web ». BOISSIER O., GUESSOUM Z., Eds., 12th Journées Francophones sur les Systèmes Multi-Agents, Paris, France, November 2004, Hermès
- (AMST, 2008). Advanced Manufacturing Systems & Technology center. “Manufacturing Systems Engineering ». <http://wolftest.lboro.ac.uk/research/manufacturing-technology/amstc/Research.htm>
- (Ankolenkar & al, 2002). Ankolenkar. B, Hobbs. M, Lassila. J.R, Martin. O, McDermott D.L , McIlraith. D, Narayanan. S.A, Paolucci. S, M. Payne, Sycara. T.R. “DAML- S : Web service description for the semantic web”, 2002
- (Ankolenak & al, 2003). Ankolenkar. B, Hobbs. M, Lassila. J.R, Martin. O, McDermott D.L, McDermott. D, McIlraith. D, Narayanan.S.A, Paolucci. S, Payne. M, Sycara. T.R, Son. K, Zeng. H. « DAML-S : Semantic markup for web services », 2003
- (Anderson, 2001). Andersen. B. “What is an ontology?” 2001
- (Apache, 2007). Site web. www.ws.apache.org/axis/
- (Arsène, 2001). Arsène. S. « Système multi-agents : Une analyse comparative des méthodologie de développement ». Département de mathématiques et d’informatique. Université du Québec à Trois-Rivières, 2001.
- (auctionbot, 2006). Site web. www.auctionbot.com
- (AWT, 2007). Agence Wallonne des Télécommunications, « Qu'est-ce que l'e-business ? ». <http://www.awt.be>.
- (Bacus, 1997) : Bacus. M. « Modélisation du choix de partenaires dans la coopération entrePMI », Thèse de doctorat, Université Paris IX. 1997.

- (bargainfinder, 2005). Site web. www.cdrom-guide.com/bargainfinder.htm
- (Berners-Lee, 2001). Berners-Lee Tim, Hendler. James & Lasilla. Ora. « The Semantic Web », Scientific American, May 2001
- (Boutigny, 2004). Boutigny. E. « Coopération dans l'entreprise et compétence collective », AGRH 2004, Actes du congrès, Tome 3, La gestion des compétences.
- (Brahimi & al, 2009). Brahimi. M, Seinturier. L & Boufaïda. M. «A Multi-Agent Architecture for Developing Cooperative E-Business Applications». In the International Journal of Information System and Supply Chain Management, 4(2), 2009, pp 43-62. IGI Global, USA.
- (Brahimi & al, 2006). Brahimi. M, Boufaïda & M, Seinturier. L. « Integrating Web Services within Cooperative Multi Agent Architecture ». In proceeding of the international conference on Internet and Web Applications ICIW'06, pp 197-202. Guadeloupe, France, 2006. Washington: IEEE Computer Society.
- (Brahimi & al, 2004). Brahimi. M, Boufaïda & M, Seinturier. L. «A federated Agent Based Solution for Developing Cooperative E-Business Applications». In proceeding of the fourth international workshop in Web Based Collaboration WBC'04, pp 289 – 297. Saragossa, Spain, 2004. Washington: IEEE Computer Society.
- (Burstein & al, 2008). Burstein. M, Ankolenkar. A, Paolucci. M & Srinivasan. N. «OWL-S: Semantic Markup for Web Services». <http://www.daml.org/services/owl-s/1.0/owl-s.pdf>
- (Cabral & al, 2004). Cabral. L, Domingue. J, Motta. E, Payne & T. Hakimpour. F. «Approaches to Semantic Web Services: An Overview and Comparisons», 2004.
- (Caglayan & Harrison, 1998). Caglayan. A & Harrison. C. « Les agents. Applications bureautiques, Internet et Intranet ». Masson, Paris, 1998.
- (Cerisier, 1999). Cerisier. J. F. « Environnements d'apprentissages collectifs en réseaux », Poitiers, Paris 8, Groupe de recherche sur l'apprentissage et les médias en éducation, 1999.
- (Chandler, 1988). Chandler. A. D, « La main visible des managers ». Economica, Paris, 1988.
- (Chaïb-draa, 1996). Chaïb-draa. B. «Interaction between Agents in Routine, Familiar and Unfamiliar Situations». International Journal of Intelligent & Cooperative Information Systems, 5(1), pp 1-25.1996
- (Christensen & al, 1996). Christensen. J, Thouati. M & Struger. O. « Les système holoniques », Cimax, Edition Terrain, N° 7, Janvier-Fevrier, 1996.
- (Christian, 2007). Cristian Bilalès. « La nouvelle économie en question », 2007.

- (Christophe, 2002). Christophe. N. « Définition des services- Web ». Université de Bourgogne, 2002.
- (Clément & al, 2006). Clément. J, Pascal. D & Stefano A. C. « Intégration orientée service des modèles Grid et Multi-Agents », 14èmes Journées Francophones sur les Systèmes Multi-Agents, JFSMA06, pp. 271-274, Annecy, France, Octobre 2006
- (Corba, 2009): Site Web. www.corba.org
- (Cubat Dit Cros, 2005). Cubat Dit Cros. C. « Agents Mobiles Coopérants pour les Environnements Dynamiques » Thèse de doctorat, Institut National Polytechnique de Toulouse, 2005.
- (Dallon, 20032). Dallon. G. « DAML-S : interactions, critique et évaluation », 2003.
- (Dameron, 2000).Dameron. S. « Génération de la coopération dans l'organisation : Le cas d'équipes projet », Thèse, Université Paris IX Dauphine, 2000.
- (Daml, 2007). Site web. www.daml.org/services/swsf/1.0/swsl/#sec-language
- (Didier & Tanguy, 2001). Didier. G, Tanguy. C. « XML pour l'entreprise. Xml et les applications d'entreprise ». <http://www.application-servers.com/livresblancs/xml/>
- (Lestel & al, 1994). Lestel. D, Grison. B & Drogoul. A. « Les agents réactifs et le vivant dans une perspective d'évolution coopérative ». *Intellectica*, 19 (2), pp 73-90. 1994.
- (Dussauge & Garrette, 1995). Dussauge. P & Garrette. B. « Les stratégies d'alliances », Editions d'Organisation, Paris.1995.
- (eFORCE, 2003). eFORCE, « Comment tirer profit des Web services dans le monde réel », eFORCE France, 2003.
- (ETD, 2008). Site Web. www.projetdeterritoire.com
- (Ferber, 1995). Ferber. J. «Les systèmes multi-agents » InterEdition 1995
- (Ferber, 1997). Ferber. J. «Les systèmes multi-agents : Un aperçu général » *Techniques et sciences informatiques*, 16 (8), pp 979-1012. juillet 1997.
- (Ferber & al, 2003). Ferber. J, Gutknecht. O & Michel. F. «From Agents to Organizations: an Organizational View of Multi-Agent Systems». *Proc of the Conference AOSE'03*, Melbourn, Australia, pp. 214-230. 2003.
- (FIPA, 2002a). «Fipa contract net interaction protocol specification». Retrieved January 15, 2008 from <http://www.fipa.org>.
- (FIPA. 2002b). «Fipa acl message structure specification». Retrieved January 15, 2008 from <http://www.fipa.org>.

- (Firefly, 2005). Site web. www.ffly.com/
- (Fourati, 2006). Fourati. F. «Veille stratégique : l'évaluation de l'utilisation des agents intelligents ». Thèse de sciences de gestion. Université Paris Dauphine, 2006.
- (Genesreth & al, 1993). Genesreth. M. R & Sing. N. P. « A knowledge sharing approach to software interoperability » Technical report logic-93-1, Department of computer science, Stanford University. 1993.
- Giller, 2006). Giller A. « J2EE vs .NET Une comparaison à l'aide de l'application Cybercoach ». Travail de master, Université de Fribourg, Suisse. 2006.
- (Grosso & al, 2003). Grosso. A, Gozzi. A, Coccoli. M & Boccalatte. A. «An Agent Programming Framework based on the C# language and the Common Language Infrastructure». 1st International Workshop on C# and .NET Technologies on Algorithms, Computer Graphics, Visualization, Computer Vision and Distributed Computing. Plzen, Czech Republic. Science Press.
- (Gillbert & al, 2003). Gilbert. B, Michel. L. « Les Web services et leur impact sur le commerce B2B », CIRANO, 2003.
- (Greenwood & al, 2004). Greenwood. D & Calisti. M. « Engineering Web Service - Agent Integration », IEEE Systems, Cybernetics and Man Conference, The Hague, Netherlands, October 2004. Washington, IEEE Computer Society.
- (Hammonds, 2001). Hammonds. K. «Michael Porter's Big Ideas». In Fast Company Learning, Issue 44, pp 150.
- (Hawlett, 2008). Site web. www.hp.com
- (Hennart, 1988). Hennart. J.F. «A transaction cost theory of equity joint ventures». Strategic Management Journal, 9(4). pp361-374. 1988.
- (IKVM, 2008). Site web. <http://www.ikvm.net>
- (Ishikawa et al, 2004): Ishikawa. F, Yoshioka. N & Tahara. Y. « Toward Synthesis of Web Services and Mobile Agents ». 2nd International Workshop on Web Services and Agent Based Engineering, WSABE'04, pp 48-55, 2004. New York, NY, USA
- (Java, 2008). Site web. www.java.sun.com/javase/technologies/core/basic/rmi
- (Jax, 2007). Site web. www.jax-ws.dev.java.net/
- (Jérouville, 2001). Jérouville. P. «la force de l'union». Entreprise d'aujourd'hui, n°41. février 2001,
- (Jin & al, 2003). Jin. T & Goschnick. S. « Utilizing Web Services in an Agent Based Transaction Model (ABT) », Workshop on Web services And Agent-based engineering, Melbourne, Australia, 2003.

- (JND, 2002). Journal Du Net, « L'entreprise étendue: discours ou réalité ? », 2002, <http://www.journaldunet.com/>.
- (Joshua, 2008). Site web. <http://www.razor.occams.info/software.xpd>
- (Kadlec & al, 2002). Kadlec P & Mares. M. «eCommerce Business Opportunity for SMEs in Central and Eastern Europe». eBusiness and e-Work, Conference proceedings Challenges and Achievements in E-Business and E-Work, Part 1. pp 195–202, 2002. IOS Press, Amsterdam.
- (Karssenty, 1994). Karssenty. A. «GroupDesign : un collecticiel synchrone pour l'édition partagée de documents ». Thèse de Doctorat d'état, Université Paris XI Orsay. Février1994.
- (Khubaib & Mansoor-uz-Zafar, 2008). Khubaib Ahmed Qureshia, and Mansoor-uz-Zafar « Intelligent ubiquitous computing and e-Business solution». International Journal of Information Management. 28(2), pp 128-135.
- (Koestler, 1969). Koestler. A. « The Ghost in the machine », ISBN 0 41 019192 5, 1996.
- (Konsynski, 1993). Konsynski. B. « Strategic Control in the Extended Enterprise »; IBM Systems Journal, 32(1), pp 111-142. 1993.
- (Labrou & al, 1997). Labrou. Y & Finin. T. « A proposal for a new KQML specification » Technical report, Computer Science and Electrical Engineering Department, University of Maryland Baltimore Country, Baltimore, Maryland, February 1997.
- (Lestel & al, 1994) : Lestel, D., Grison, B. & Drogoul, A. «Les agents réactifs et le vivant dans une perspective d'évolution coopérative». Intellectica, 19(2), pp 73-90.
- (Lipnack & al, 1993). Lipnack J & Stamps. J. « The TeamNet factor: Bringing the Power of Boundary Crossing into the Heart of your Business ». Oliver Wight Publication, 1993
- (Lyell & al, 2003): Lyell. M, Rosen. L, Casagni-Simkins. M, Nprris. D. « On Software Agents and Web Services: Usage and Design Concepts and Issues ». 1st International Workshop on Web Services and Agent Based Engineering, Melbourne, Australia, July 2003.
- (Maker, 2006). Site web. <http://maker.media.mit.edu>
- (Matthew, 2002). Matthew K. H. « Legal Considerations for Web Services». Federated Systems and E-Commerce, Copyright 2002.
- (Maximilien & al, 2003): Maximilien. E. M & Singh. M. P. « Agent-based architecture for autonomicWeb service selection ». 1st International Workshop on Web Services and Agent Based Engineering, WSABE'03, Sydney, Australia, July 2003.

- (McGuinness & al, 2003): McGuinness D.L. and van Harmelen F. «OWL Web Ontology Language Overview». <http://www.w3.org/TR/owl-features/> , Août 2003, World Wide Web Consortium - Candidate Recommendation
- (Messnarz & al, 2000). Messnarz. R, Biro. M, Velasco. G. O. & Suilleabhain. G. G. «The Victory Project – A Virtual Enterprise for SPI». In Proceedings of the EuroSPI2000 Conference, Copenhagen Business School., 2000.
- (Microsoft, 2009). Site web. www.microsoft.com
- (Microsoft, 2008). Site web. www.microsoft.com/biztalk
- (Mittman, 1994).Mittmann. R. «The HMS Test Case». In First European Conference on Holonic Manufacturing Systems, Hannover, Germany, December 1994.
- (Mohini & al, 2004). Mohini. S & Dianne. W. «E-Business Innovation and Change Management». Idea Group Publishing , 2004.
- (Mostefaoui & al, 2003). Mostefaoui. S. K, Mostefaoui. G. K. « Towards A Contextualisation of Service Discovery and Composition for Pervasive Environments ». Workshop on Web services And Agent-based engineering, Melbourne, Australia, 2003.
- (Msdn, 2009). Site web. www.msdn.microsoft.com
- (Negri & al, 2006). Negri. A, Poggi. A , Tomaiuolo. M & Turci. P. «Agents for e-Business Applications », AAMAS'06, Hakodate, Hokkaido, Japan. ACM, 2006.
- (Nouvelle, 2008). Site web : www.nouvelle-economie.net.
- (OMG, 2009). Site web www.omg.org
- (Ouchi, 1980). Ouchi. W. « Markets, Bureaucracies, and Clans », Administrative Science. pp 129-141. 1980.
- (Papazoglou, 2001). Papazoglou. M. P. « agent oriented technology in support of e-business ». Communication of the ACM, 44(4). 2001.
- (Patel & al, 2000). Patel. K & McCarthy. M. P. « The Essentials of eBusiness Leadership DIGITAL Transformation». McGraw Hill, USA, 2000
- (Patricia Seybold Group, 2002). Patricia Seybold Group's executive series «An executive's guide do web services, How to optimize Web Services investments to improve your customer experience». 2002.
- (Peters, 2005). Peters. J. « Integration of Mobile Agents and Web Services ». 1st European Young Researchers Workshop on Service Oriented Computing (YR-SOC'05), Leicester, UK, Software Technology Research Laboratory, De Montfort University, pp 53-58. 2005.

- (Petras & Tadas, 2004): Petras. Barđauskas, Tadas. Darapovas, « Problems of E-business implementation in SMEs” ISSN 1392–1258. EKONOMIKA 2004 65
- (Petrone, 2003). Petrone. G. « Managing flexible interaction with Web Services », Workshop on Web services And Agent-based engineering, Melbourne, Australia, 2003.
- (Persona, 2006). Site web. <http://www.personalogic.com/>
- (Phan, 2005). Phan. Q. « Ontologies et Web Services ». Rapport final, Institut de la Francophonie pour l'Informatique, 2005.
- (Pigneur, 1997). Pigneur. Y. « Systèmes d’information, entreprise étendue et commerce électronique ». Inforsid, Toulouse, France, 1997.
- (Pine & al, 1995). Pine. J, Peppers. D & Rogers. M « Do you want to keep your customers forever? » Harvard Business Review, 73(2), pp. 103-114.1995.
- (Porter, 1985). Porter. M. E. & Millar. V. E. « How Information Gives You Competitive Advantage »; Harvard Business Review, Vol. 63, No. 4, July-August 1985, pp. 149-160
- (Protégé, 2008). Site web. <http://www.protege.stanford.edu>
- (Puthod, 1996). Puthod. D. « Comprendre les alliances de PME (à partir d’une base de connaissances construite sur les préoccupations de l’entrepreneur) », 5ème Conférence de l’AIMS, Lille. 1996.
- (Ricardo, 2004). Ricardo. D. « Découverte et Sélection de Services Web pour une application Mélusine », thèse de Master en Mathématique Informatique, Genève, Suisse, 2004.
- (Richards & al, 2003). Richards D, Van Splunter. S, Brazier. F. M & SABOU. M. « Composing Web Services using an Agent Factory ». Workshop on Web services And Agent-based engineering, Melbourne Australia, 2003.
- (Ricordel & al, 2000). Ricordel. P. M and Demazeau. Y. «From Analysis to Deployment : a MultiAgent Platform Survey». 1st International Workshop on Engineering Societies in the Agents World (ESAW), pp 93-105, Springer Verlag, Germany. 2000.
- (Sabouret, 2002). Sabouret. N. «Programmer des composants actifs dans le Web sémantique », Journées de l'Action Spécifique STIC « Web Sémantique ». 2002. es composants actifs dans le Web sémantique
- (Sachs & al, 2008). Sachs. M & Ibbotson. J. «TpaML», <http://www.ibm.com/developer/xml/tpaml/tpaspec.pdf>
- (Sansonnet, 2005). Sansonnet. J-P. « Applications agents pour le workflow et le BPM (business process management) ». Université Paris Sud-XI, Février 22, 2005

- (Saubesty, 2003): Saubesty. C «dynamique de la construction de coopérations transversales» thèse de doctorat, centre de recherche DMSP Université Paris IX Dauphine-France, 2003.
- (Saul, 1991). Saul. G. «Computer-supported Cooperative Work and Groupware», Mapping Strategic Thought, 1991.
- (Seghrouchni & al, 2004). Seghrouchni A. E. F, Haddad S, Melitti. T & SUNA. A. « Interopérabilité des systèmes multi-agents à l'aide des services Web ». BOISSIER O., GUESSOUM Z., Eds., 12th Journées Francophones sur les Systèmes Multi-Agents, Paris, France, Hermès, pp 91-104. 2004.
- (Schmidt, 19994). Schmidt. K. « Modes and Mechanisms of Interaction in Cooperative Work », Riso National Laboratory, Denmark, 19994.
- (Schobbens & al, 2003). Schobbens. P.Y, Massonet. P & Ponsard. C. « Les agents intelligents au service de la gestion électronique de l'entreprise ? » Rencontre INFOPOLE, 2003.
- (Sheng-Uei & Fangming, 2002):Sheng-Uei Guan and Fangming Zhu, «Agent Fabrication and Its Implementation for Agent-based Electronic Commerce». International Journal of Information Technology and Decision Making (IJITDM), 7(3), pp 473-489.
- (Shoham, 1993). Shoham. Y. « Agent-oriented programming », Artificial Intelligence, Vol. 60, pp. 51-92. 1993
- (Singh, 2000). Singh. M. "Electronic commerce in Australia: Opportunities and factors critical for success". Proceedings of the 1st World Congress on the Management of Electronic Commerce. Hamilton, Ontario, Canada, January 2000
- (Smith & al, 19995). Smith, Caroll & Ashford. « Intra and inter-organizational cooperation: toward a research agenda », Academy of Management Journal, vol. 38, n°1, pp.7-23. 1995.
- (Soubie & al, 1996). Soubie. J.L, Buratto. F, Chabaud. C. « La conception de la coopération et la coopération dans la conception », in Coopération et conception, Octares Editions. 1996.
- (Stencil, 2006). Site web. www.Stencilgoup.com/ideas_scope_200106wsdefined.htm
- (Sun, 2009). Site web. www.sun.coml
- (Thiberge, 2006). Thiberge. P. « Présentation des techniques juridiques de coopération interentreprises ». Site Web : <http://www.cultureco.com>.
- (UDDI, 2009). Site web. uddi.xml.org
- (UN/CEFACT, 2001). « ebXML Technical Architecture Specification v 1.04 », 2001

- (Unilog, 2003). Unilog management. « les Web services dans l'EAI-BtoB : opportunité technique ou choix stratégique pour l'intégration étendue ? », 2003
- (Vernadat, 1996). Varnadat. F. « Enterprise modeling and integration : principles and applications ». 1st ed., London; New York: Chapman & Hall. 1996
- (Virdell, 2003). Virdell. M. « Business processes and workflow in the Web services world». developer Works, January, 2003.
- (Wang & al, 2006): M. Wang, W. K. Cheung, J. Liu and Z. Luo, «Agent-supported web service composition for supply chain management». Proceedings of IEEE Joint Conference on E-Commerce Technology and Enterprise Computing, E-Commerce and E-Services (CEC' & EEE' 06), IEEE Computer Society Press, San Francisco, USA, June 2006, pp.328-332.
- (Weilnhammer, 2000). Weilnhammer. U. «Electronic Commerce – A Framework for Technological Transformation in a Business-to-Business Environment». Master Thesis, Walter A. Haas School of Business, University of California at Berkeley and Chair for General and Industrial Management, Technical University Munich, May 2000 (p. 16)
- (Weiss, 1999). Weiss. G. «Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence». The MIT Press, Cambridge, Massachusetts. 1999.
- (Whinston & al, 1998). Whinston. A, Stahl. D & Choi. S. « The economics of electronic commerce». Indianapolis, Indiana: Macmillan Technical Publishing, 1998.
- (W3C, 2009). Site web. www.w3.org/RDF
- (W3C, 2007). Site web. www.W3.org/TR/owl-features
- (W3C, 2008). Site web. www.w3.org/TR/wsdl
- (Xml, 2009). Site web. www.xml.com

"Contribution à la coopération dans les systèmes d'information basés web"

Résumé

Dans le monde économique actuel et face à l'évolution rapide des aspects et besoins technologiques liés aux nouvelles technologies de l'information et de la communication, les défis de l'entreprise sont devenus de plus en plus compliqués. L'entreprise doit émerger, modifier et adopter des nouvelles stratégies de management et notamment informatiser son ancien système d'information. L'adoption du business électronique tout en engageant dans des alliances de coopération avec les autres éléments de l'environnement économique, semble plus que nécessaire pour que l'entreprise puisse assurer sa continuité dans un monde de plus en plus concurrentiel.

Dans cette perspective, notre objectif est de proposer un modèle à base d'agents avec l'intégration des Web services pour le développement des applications e-business coopératives. Pour répondre aux besoins de nos objectifs, ce modèle est basée essentiellement sur deux concept clés : l'agent et le web service. La partie agents est le noyau de ce modèle, elle est représentée par un ensemble de groupes identiques. Chaque groupe possède un ensemble d'agents d'applications dirigés par un agent coordinateur. De plus, nous avons proposé les fonctionnalités publics ou celles destinées au monde externe comme Web services afin de libérer le noyau agent. Ces Web service sont manipulés à l'aide d'un agent médiateur appelé Agent Finder. Dans ce modèle, la notion d'agent avec toutes ses potentialités et celle du Web service avec sa souplesse constituent un soutien énorme pour la résolution des problèmes liés au domaine du e-business coopératif.

Mots clés: e-business; cooperation; agent; web services; DotNet.

"Contribution to cooperation in Web based information systems"

Abstract

The fast evolution of the World Wide Web and the changes made in the economic world motivate companies to invest more and more in the e-business. Since e-business became an economic cooperation activity, we can say that it has changed the rules of the market and the role of information technology (IT). The rapid growth of the technical and organizational needs creates several intelligent solutions for the e-business domain. The main perceived obstacle to e-business is the fewest number of methodologies that exist for the development of complete e-business projects. In order to ensure the effective establishment of e-business applications in a company, it must be developed and implemented from the business requirements and not from the technological viewpoint.

In this perspective, we propose in this thesis a multi-agent approach that provides solutions to the problems raised during the development of cooperative e-business applications. This approach is organized in the form of cooperative application groups representing the different parts of a company. Agent coordinators orchestrate the cooperative work of these groups. The most requested functionalities inside the company and those offered to the external world can be exported as Web Services. These Web Services are described with DAML-based Web Service ontology (OWL-S) and managed with an intermediate agent called Web Service Finder Agent. The proposed solution provides a new vision of the cooperation context where the companies and their partners share knowledge and offer functionalities as agents and Web Services.

Keywords: e-business; cooperation; agent; web services; DotNet

“إسهام في مفهوم التعاون في مجال نظم المعلومات القائمة على الويب”

ملخص:

في الوقت الحالي ومع تقدم التقنيات الجديدة للإعلام والاتصال وجدت المؤسسات نفسها مجبرة على تبني العمل بصيغة الصفقات الالكترونية. إضافة إلى ذلك وفي ظل التنافس الحاد بين المؤسسات أصبح العمل بصيغة تعاونية مع مختلف المؤسسات الأخرى شيئاً لازماً لازدهار واستمرار المؤسسة. إلا أنه يجب أن ننوه بقلة الحلول التقنية الشاملة في هذا المجال المهم.

في هذا الإطار تقدم في هذا العمل حلاً تقنياً لتطوير أنظمة الصفقات الالكترونية التعاونية. يركز هذا الحل على استغلال تقنية العون مع خدمات الويب. حيث نقتراح تقسيم التطبيقات إلى مجموعة أفواج من الأعوان المتجانسة تؤدي أدواراً محددة حيث يقوم بالإشراف على كل فوج عون منسق يربط الاتصال بجميع الأعوان الأخرى وبالعالم الخارجي. تكون الخدمات الأكثر طلباً خصوصاً من العالم الخارجي على شكل خدمات ويب بحيث يسيرها عون خاص يشرف على تنظيمها وإجادةها لمن يريدتها. وتستغل على هذا المستوى ومن أجل فعالية أكبر تقنية الانطولوجيا للحصول على دقة أكبر في البحث والتعامل مع خدمات الويب هذه.

لكلمات المفتاح: الصفقات الالكترونية، لتعاون، العون، خدمات الويب.