

Une Architecture Basée Agents Mobiles Pour la Recherche D'information dans des Sources Hétérogènes et Réparties

Nadia Ben Seghir¹, Okba Kazar¹

¹Département D'informatique, Université de Biskra,
07000, Biskra, Algérie
inf_nadia@yahoo.fr, kazarokba@yahoo.fr

Résumé. De nos jours le besoin en information devient très exigeant et l'interopérabilité des systèmes devient une approche incontournable. Une demande d'information peut être répartie sur plusieurs systèmes répartis d'où la nécessité d'une reconstitution de la réponse par une entité gérante. D'autre part, les échanges des données et la répartition des tâches d'une application distribuée nécessitent l'interaction entre différentes entités à travers le réseau. Aujourd'hui, le modèle "client/serveur" où les échanges se font par envoi des messages à travers le réseau est le modèle le plus utilisé. Ce modèle possède l'inconvénient d'augmenter le trafic sur le réseau et exige une connexion permanente. Dans cet article, nous proposons une nouvelle approche qui utilise les agents mobiles. Ces derniers apparaissent dans ce contexte comme une solution prometteuse facilitant la mise en oeuvre d'applications réparties. Dans ce papier, nous décrivons notre architecture à base d'agents mobiles pour la recherche d'information dans des sources hétérogènes et réparties. Chaque agent mobile de recherche sera utilisé pour visiter le site serveur cible de l'application afin de collecter des informations pour son client, ce que permet à ce dernier d'interagir localement avec le serveur, et donc de réduire le trafic sur le réseau en ne transmettant que les données utiles.

Mots Clés : Agent mobile, Recherche d'information, Système réparti, Web.

1 Introduction

L'évolution des réseaux à grande échelle a permis la naissance d'un grand nombre de nouvelles applications qui se développent autour de ce type de réseau : commerce électronique, recherche d'information sur le web, plate-forme pour calcul réparti, etc. Ces applications sont formées par des entités réparties et la bonne fonctionnalité nécessite la communication et les échanges entre ces entités. Aujourd'hui, le modèle "client/serveur" où les échanges se font par des appels distants à travers le réseau est le modèle le plus utilisé. Dans ce modèle, seul le client représente une application au sens propre du terme et le rôle du serveur est de répondre aux demandes des clients. Le serveur construit ses réponses indépendamment du client, ainsi une partie des données envoyées est inutile augmentant ainsi le trafic sur le réseau. De plus ce

2 Nadia Ben SeghirPIP, Okba KazarPIP

modèle exige une connexion permanente entre le client et le serveur, ce qui n'est pas le cas des terminaux mobiles qui sont exposés à la perte de la connexion. Dans cet article nous proposons une nouvelle approche qui utilise les agents mobiles. Ces agents sont des entités qui se déplacent d'une machine à une autre sur le réseau, sans perdre leurs codes ni leurs états. Ainsi, en envoyant les agents là où les tâches se font, les messages échangés deviennent locaux et libèrent d'autant la charge du réseau.

Une des applications les plus importantes dans le domaine des agents mobiles est la recherche d'informations sur le Web. Dans ces applications (recherche des hôtels, réservation d'un billet d'avion, etc...), des agents se déplacent sur différents sites pour rechercher des informations pour leurs clients. De nombreux travaux ont été élaborés afin d'introduire la technologie d'agents mobiles et les concepts liés à cette dernière pour la recherche d'information dans des environnements dynamiques. Le concept d'agent mobile apparaît dans ce contexte comme une solution facilitant la mise en œuvre d'applications réparties.

Nous avons abordé notre architecture selon ces points de vue. Dans ce papier nous proposons une approche basée agents mobiles permettant de rechercher des informations pertinentes dans un réseau de ressources d'informations, où les ressources sont géographiquement distribuées, hétérogènes de par leur contenu.

Le présent article est structuré comme suit : dans la prochaine section nous allons présenter brièvement un certain nombre de travaux qui lient agents mobiles et la recherche d'information, cette partie est consacrée à la comparaison de notre approche avec des travaux relativement proches. Par la suite, notre nouvelle approche pour la recherche d'information, celle basée sur les agents mobiles, sera détaillée dans la troisième section, en présentant tout d'abord l'architecture générale suivie des composantes de cette architecture. Enfin, une conclusion fait un bilan de cette recherche et présente les perspectives envisagées.

2 Travaux Connexes

Dès les premières publications, la recherche d'information a été présentée comme une application potentielle importante des agents mobiles voire comme la "killer application" [1]. Divers travaux significatifs ont été menés dans le domaine de la recherche d'information et s'appuient sur la technologie d'agents mobiles. Parmi ceux-ci on distingue:

- Dans [4], les auteurs présentent une application de recherche de documents textuels à base d'agents mobiles. Il s'agit cependant d'une application de recherche simple dans un réseau local (avec centralisation des informations sur les serveurs via un mécanisme de pages jaunes, sans contrainte de sécurité, ni découverte dynamique de nouveaux serveurs). L'agent de recherche s'appuie sur des agents d'observation de l'état du réseau et sur un agent stationnaire qui sert d'interface avec le serveur d'information local.
- Le système DBMS-Aglet [10] implante une solution à base d'agents mobiles en Java pour l'interrogation de bases de données hétérogènes via le Web. Un agent mobile transporte la requête sur le site serveur où il acquiert dynamiquement le

pilote JDBC qui convient, il pose ensuite sa requête et retourne sur le site client avec les résultats.

- M3 "MultiMedia Database Mobile agents" [9] est un système de recherche de données multimédia par le contenu qui repose sur les agents mobiles, Java et CORBA. L'agent mobile peut mémoriser les informations recueillies sur un site, les utiliser sur les sites visités ensuite, les faire évoluer pendant le parcours. Les problèmes de sécurité sont pris en compte via des mécanismes de sessions indépendantes, les mécanismes de sécurité de CORBA, et des restrictions de droits.
- Enfin d'autres travaux comme AGATHE [8], ARCADIA [5], JAVANE [2], NETSA [6], ISAME [11] proposent une autre alternative pour la recherche d'information : des modèles d'agents mobiles et multi-agents.

3 Présentation de L'architecture Proposée

Dans cette section, sont tout d'abord présentés les objectifs du système proposé, son architecture générale mettant en évidence ses quatre principales couches, son fonctionnement général, et enfin quelques détails d'implémentation sont fournis.

3.1 Objectifs du Système

Chercher, trouver et intégrer de l'information à partir de ressources de données dispersées géographiquement est un problème difficile auxquels se trouvent confrontées de nombreuses organisations. Ces dernières années, la taille et la variété des données accessibles par réseaux ont considérablement augmenté. De nouvelles solutions sont devenues nécessaires pour manipuler l'information issue de ressources dynamiques comme le World Wide Web.

Le concept d'agent mobile apparaît dans ce contexte comme une solution facilitant la mise en œuvre d'applications dynamiquement adaptables, et il offre un cadre générique pour le développement des applications réparties. Dans ce modèle "agent mobile", un agent est un processus possédant un contexte d'exécution, incluant du code et des données, pouvant se déplacer de machine en machine (appelées serveurs) afin de réaliser la tâche qui lui est assignée [7]. A priori, les avantages des agents mobiles sont nombreux :

- L'exécution d'agents spécialisés offre d'avantage de souplesse que l'exécution d'une procédure standard sur les sites serveurs, et permet des transactions plus robustes que les transactions distantes.
- L'asynchronisme et l'autonomie des agents leur permet de réaliser une tâche tout en étant déconnecté du client, ce qui est particulièrement utile dans le cas de supports physiquement mobiles (clients ou serveurs d'information).
- Les agents sont capables de chercher l'information d'une façon plus intelligente, par exemple en cherchant selon des concepts. Les agents sont également capables de corriger les requêtes de l'utilisateur, en se basant sur le modèle de ce dernier ou sur d'autres informations.

4 Nadia Ben SeghirPIP, Okba KazarPIP

- Les agents peuvent créer leurs propres bases de connaissances qui sont mises à jour après chaque recherche. Si l'information change de site, les agents sont capables de la trouver et, par la suite, s'adapter à ce changement. En plus, les agents sont capables de communiquer et coopérer entre eux (et c'est là leur vraie force), ce qui accélère et facilite la recherche.

Comme nous l'avons déjà évoqué, l'architecture proposée dans ce travail est une approche basée agents mobiles conçue pour la recherche d'information dans des sources hétérogènes et réparties. Elle définit l'ensemble des composants (agents) et modules fonctionnels décrits en terme de leurs comportements et interfaces, ainsi que la façon d'interaction de ces composants afin d'accomplir correctement l'ensemble des tâches du système.

3.2 Architecture Générale du Système

L'architecture générale de notre système, illustrée à la figure 1, s'articule autour de quatre principales couches en interaction.

Couche Interface. Cette couche contient l'application qui permet au client d'interroger le système. Son rôle essentiel est de capturer le but de l'utilisateur afin de répondre le mieux possible à son besoin. Elle comprend des agents d'interface interagissant avec l'utilisateur pour l'aider à réaliser une tâche bien précise. Cette interaction se traduit par une transformation des demandes de l'utilisateur, qui transformées en des requêtes XQuery facilitent le fonctionnement de l'agent médiateur.

Couche Médiation. Cette couche regroupe tous les éléments nécessaires à l'exécution de processus de reformulation d'une requête écrite en terme de schéma global à une requête écrite en terme des schémas sources. A la fin de processus de recherche d'information, l'agent médiateur rassemble les informations trouvées sous forme d'une réponse globale qui représente l'information demandée par l'utilisateur. Notre médiateur est réalisé par le processus d'intégration qui peut reposer sur l'approche GAV (Global As View) [3]. Avec cette approche le traitement des requêtes utilisateur est plus facile. La définition d'un schéma médiateur sur des sources hétérogènes nécessite l'utilisation d'un : (i) langage d'interrogation commun ; (ii) format de résultat commun. Pour faciliter le travail de notre médiateur, Nous avons choisi d'utiliser :

- XML: comme format de résultat commun dans lequel les agents de ressources répondront au agent médiateur. Les avantages de l'utilisation de XML comme modèle d'intégration tiennent à la richesse de ce langage : abondance des descriptions et typages des données, clarté, et extensibilité, etc...
- XQuery: comme langage commun dans lequel le médiateur interrogera les agents de ressources. XQuery est un langage de requête riche qui permet d'interroger des documents XML, ses caractéristiques nous permettent de catégoriser les éléments principaux du langage. Il est issu de Quilt et reprend les avantages de XQL, XML-QL et XPath [12].

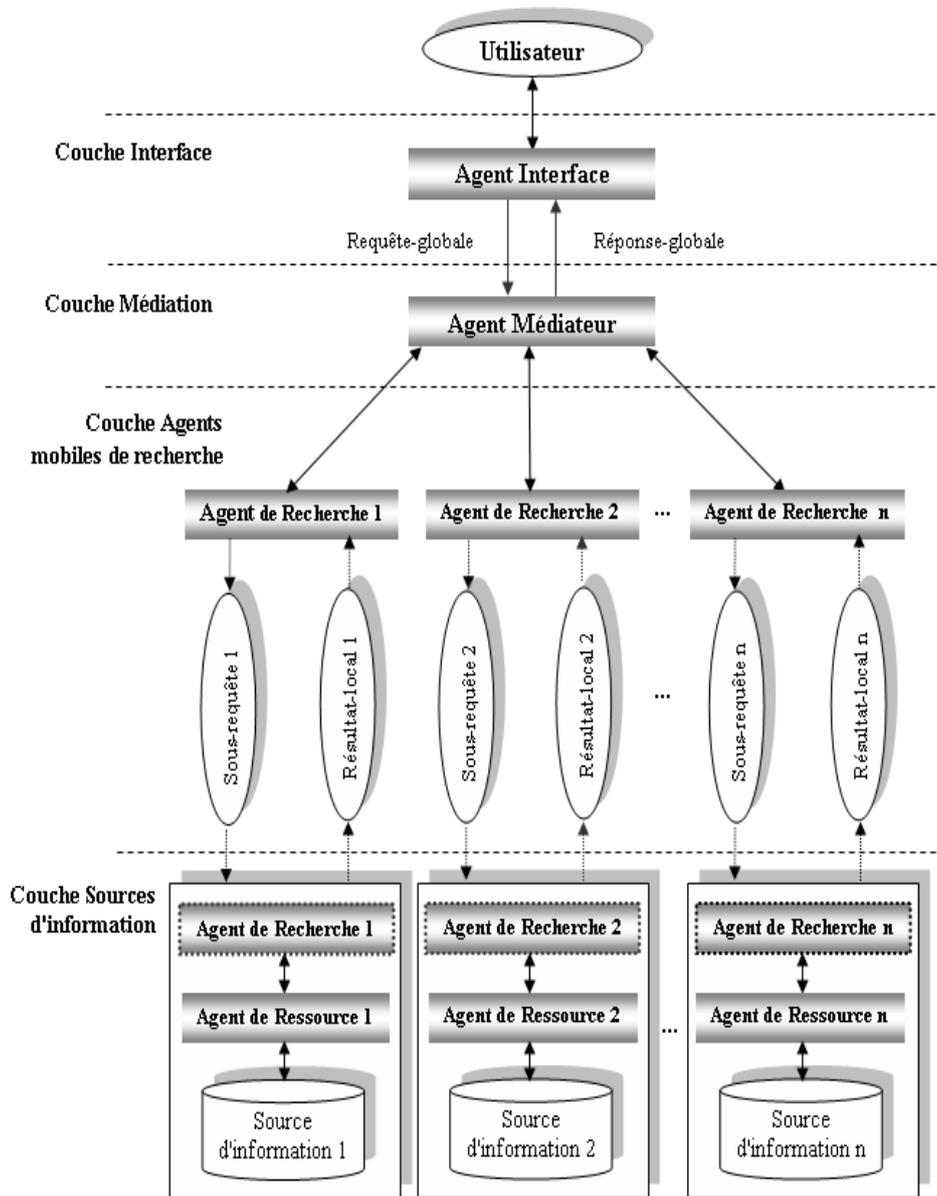


Fig. 1. Architecture globale du système.

Couche Agents Mobiles de Recherche. Cette couche contient l'ensemble des agents de recherche générés par l'agent médiateur. A ce niveau, et pour chaque sous-requête, l'agent médiateur va activer un des agents chercheurs créés. Les agents chercheurs sont des agents mobiles parcourant la toile pour trouver des informations convenables à chaque sous-requête reçue sur différentes sources de données réparties.

Couche Sources D'information. C'est une classe capable de recevoir des requêtes, de les traiter et de retourner les résultats. Cette couche regroupe premièrement l'ensemble des agents de ressource qui sont répartis sur plusieurs sites pour gérer l'ensemble de données associés à chaque site, et deuxièmement les multiples sources qui fournissent les données de base du système.

Nous allons maintenant détailler les différents agents qui comportent ces couches.

3.3 Spécification des Agents

Notre architecture comporte quatre types d'agents, comme indiqué en figure 1, dans des concentrations variables.

Agent Interface. Cet agent peut être vu comme un simplificateur permettant aux utilisateurs d'interagir avec le système (voir Fig. 2). C'est un agent stationnaire qui est responsable principalement d'acquérir toutes les requêtes des utilisateurs, envoyer ces requêtes aux agents adéquats et présenter les résultats aux utilisateurs.

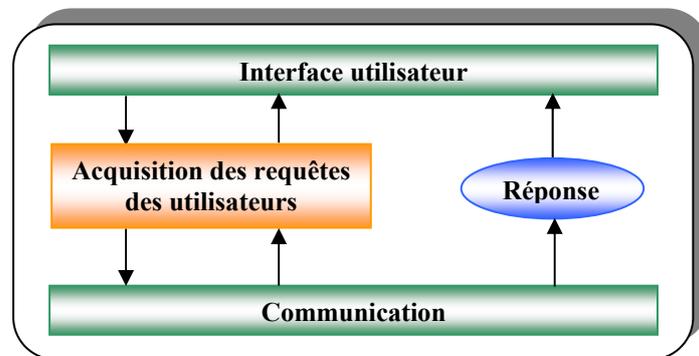


Fig. 2. L'agent d'interface et ses tâches internes.

Agent Médiateur. L'agent médiateur est un agent intelligent qui traite la distribution des données, il effectue une série de traitement sur la requête utilisateur écrite en terme de schéma global afin de la reformuler en une sous-requête écrite en terme des schémas sources. Il joue le rôle d'interface entre l'utilisateur qui pose une requête et l'ensemble des sources de données. Pour chaque sous-requête l'agent médiateur va générer un agent mobile de recherche pour être déplacé vers le site serveur cible afin de trouver les informations adéquates à la sous-requête acquise.

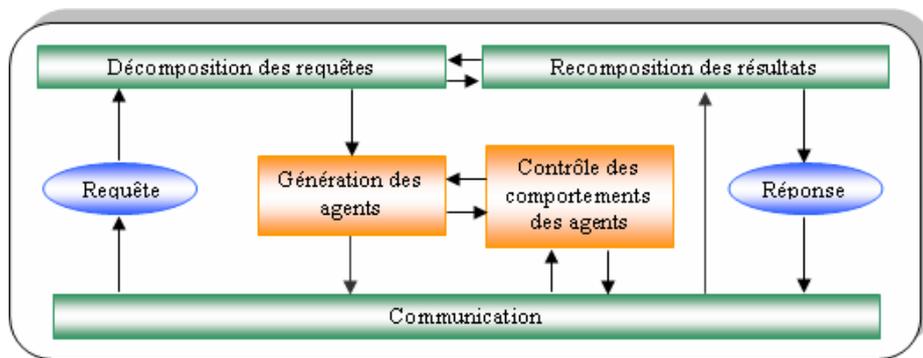


Fig. 3. L'agent médiateur et ses tâches internes.

Lorsque l'exécution des sous-requêtes est terminée, le médiateur rassemble les résultats trouvés par le groupe d'agents chercheurs sous forme d'une réponse globale qui représente l'information demandée par l'utilisateur (voir Fig.3). L'agent médiateur a la faculté de contrôler les motivations des agents mobiles de recherche qui arrivent chez lui ce qui permet d'augmenter le niveau de sécurité du système.

Agent Mobile de Recherche. Les agents de recherche sont des entités qui se déplacent d'une machine à l'autre sur le réseau afin de satisfaire le besoin de leurs clients. Sur chaque machine, ils effectuent des échanges et traitent les informations collectées. Ce traitement (recherche, filtrage, tri,...) permet de réduire la quantité d'informations transportées avec l'agent et par conséquent le trafic sur le réseau. L'agent transporte avec lui les données demandées par l'agent médiateur.

La figure 4 présente notre architecture typique de l'agent mobile de recherche. Cette architecture est de nature très simple car la puissance du système ne s'appuie pas sur la complexité de l'architecture d'un agent, mais sur les interactions entre les différents agents et leurs rôles.

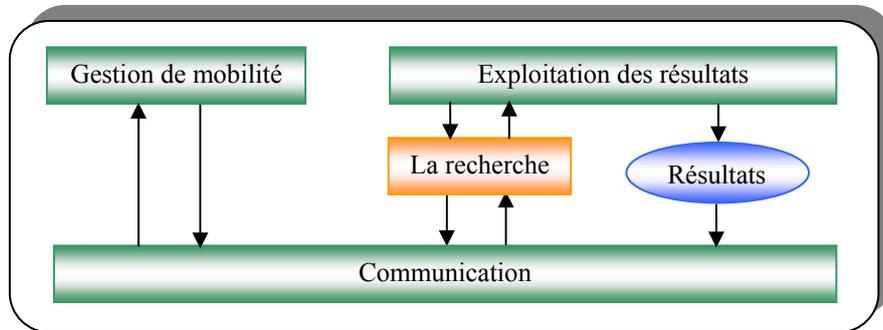


Fig. 4. L'agent mobile de recherche et ses tâches internes.

Agent de Ressource. C'est un agent stationnaire, il agit localement pendant tout son cycle de vie, dans la machine là où il a été implanté initialement. L'agent ressource est aux données ce que l'agent interface est à l'utilisateur. Cet agent reçoit des requêtes formulées en XQuery et les transforme en requête natif (propre à la source) afin d'extraire l'information requise de la base de données locale (voir Fig.5). L'information trouvée est ensuite traduite en XML en réponse à la requête du demandeur (agent mobile de recherche). Principalement, un agent de ressource gère une seule source de données. De toute évidence, plus le nombre d'agents de ressource est grand, plus nous aurons un accès à une information complète et diverse.

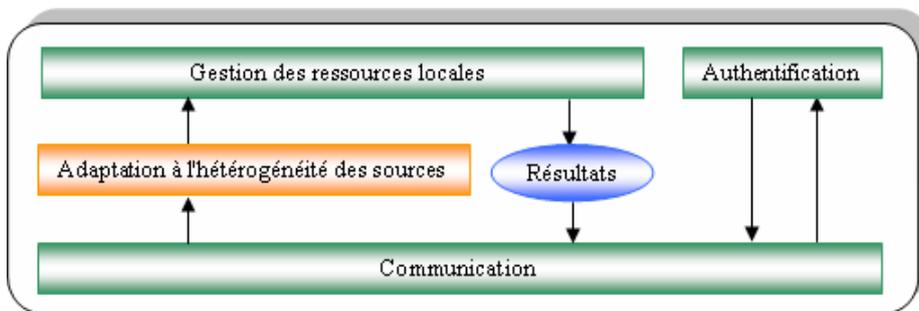


Fig. 5. L'agent de ressource et ses tâches internes.

3.4 Détails D'implémentation

Le scénario pour lequel notre approche a été testée concerne l'organisation de voyage car elle est considérée comme un exemple typique pour la recherche d'information sur

le web. Le scénario montre comment le système peut assister un utilisateur dans sa recherche sur des hôtels et des restaurants dans une ville.

Environnement de Programmation Matériel et Logiciel. Les agents de notre système sont implémentés utilisant le langage JAVA et la plate-forme de développement des agents JADE¹. JADE supporte le développement des agents avec la possibilité de les transporter d'un système à un autre. Le développement multi-agents respecte les recommandations de la FIPA², notamment le langage de communication retenu est ACL-FIPA. Les programmes sont développés avec la version 1.6 du JDK (Java Development Kit). Nous développons les corps des classes dans un environnement de programmation "JBuilder X". Les tests sont réalisés sur un réseau Ethernet qui consiste de quatre micro-ordinateurs P4 3.2 GHz, RAM 512 Mo, réseau LAN 100 Mbit/s.

Résultats D'implémentation. Nous avons crée pour chaque agent dans le système une classe hérite de la classe "jade.gui.GuiAgent" pour donner à chaque agent une fenêtre d'affichage qui permet de visualiser ses interactions avec les autres agents. Les figures 6, 7, 8 présentent respectivement les fenêtres d'affichage de l'agent médiateur, l'agent mobile de recherche, et l'agent de ressource.

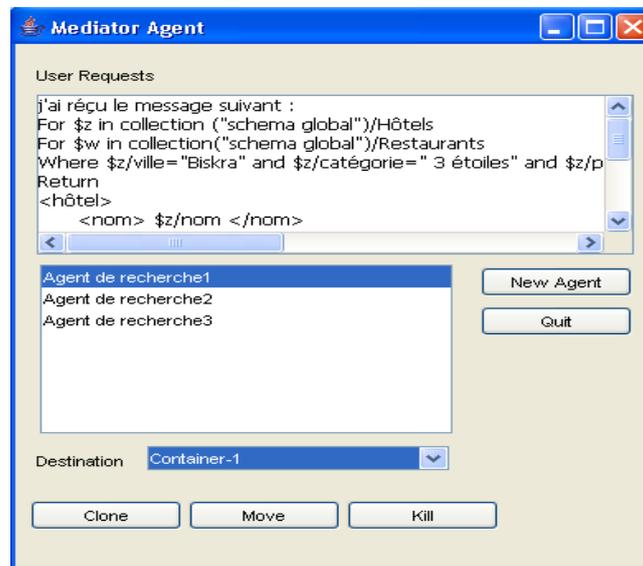


Fig. 6. Fenêtre d'affichage graphique pour l'agent médiateur.

¹ Java Agent DEvelopment Framework. Jade tutorial : <http://jade.tilab.com>

² Foundation for Intelligent Physical Agents, <http://www.fipa.org>

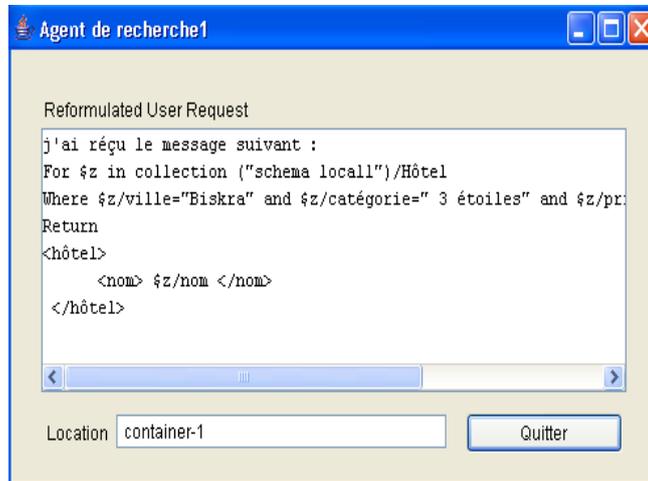


Fig. 7. Fenêtre d'affichage graphique pour l'agent mobile de recherche avec sa sous-requête reçue en XQuery.

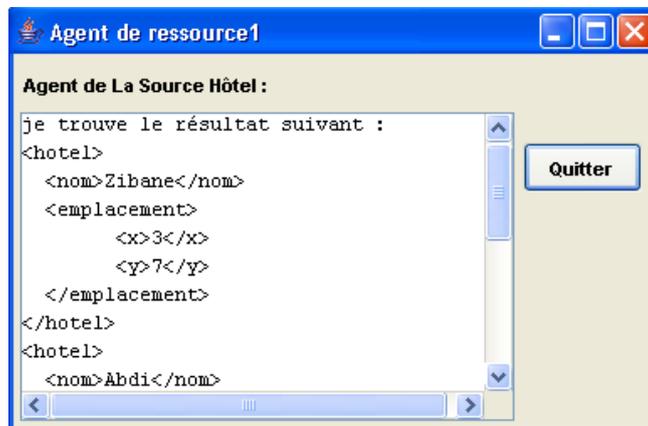


Fig. 8. Fenêtre d'affichage graphique pour l'agent de ressource avec son résultat trouvé en XML.

Pour bien comprendre le fonctionnement de notre architecture, nous présentons dans la figure suivante l'interface graphique de l'agent "sniffer" de la plate-forme JADE qui permet de visualiser l'échange de messages entre les différents agents lors du traitement d'une requête utilisateur.

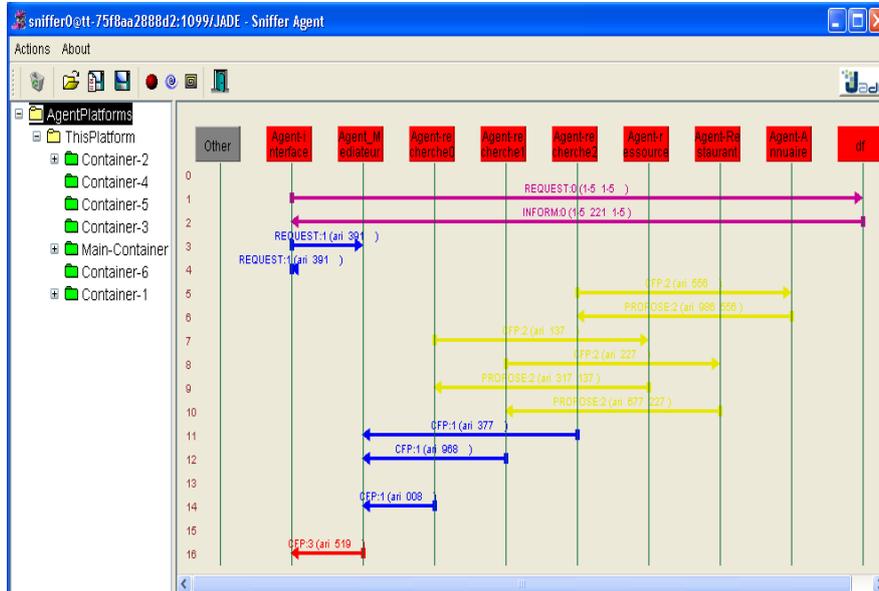


Fig. 9. La communication entre les agents de système.

4 Conclusion

Dans ce papier, nous nous sommes intéressés à la technologie d'agents mobiles et à son utilisation pour la recherche d'information dans des environnements répartis. Le but du déplacement de ces agents est généralement d'accéder localement à des données ou à des ressources initialement distantes, d'effectuer le traitement en local et de ne déplacer que les données utiles. C'est pourquoi nous proposons dans ce contexte, une nouvelle approche à base d'agents mobiles pour la recherche d'information dans des sources hétérogènes et réparties.

L'avantage de cette architecture est qu'elle utilise les agents mobiles comme une entité de communication. Il s'agit ici de réduire le trafic sur le réseau et de diminuer la quantité d'informations échangées ; dans ce cas l'agent se déplace vers la source d'informations et effectue des échanges locaux. Un autre point fort dans cette architecture vient de la décomposition des requêtes utilisateur en une structure interne susceptible d'être manipulée facilement par les différents composants du système, et la génération des agents mobiles pour chaque sous-requête, ce qui permet d'accélérer et de faciliter le processus de recherche.

Afin de montrer la faisabilité de l'architecture proposée d'un côté, et pour mieux comprendre son fonctionnement d'autre côté, nous avons choisi comme cas à étudier l'organisation des voyages car elle est considérée comme un exemple typique pour la recherche d'information sur le web. Dans ce contexte, un seul agent mobile est créé. Il est chargé de se déplacer pour réaliser la tâche demandée par le client. Nous pensons

que l'utilisation de plusieurs agents mobiles pour la recherche d'informations permet d'améliorer la qualité de la solution proposée et de réduire le délai d'attente du client. Deux cas sont à envisager : 1) les agents se connaissent et peuvent communiquer entre eux afin de réaliser la tâche demandée ; on parlera d'agents coopératifs (c'est une perspective). 2) Les agents ne se connaissent pas et dans ce cas, chaque agent réalise sa tâche indépendamment des autres agents (c'est le cas de notre approche). Les différents résultats obtenus au cours de l'expérience relatée montrent que les agents mobiles peuvent contribuer avantageusement au développement d'un système d'information distribué. De nouvelles activités de recherche peuvent être lancées afin d'améliorer le travail présenté. Les perspectives que nous proposons peuvent donc s'orienter vers les directions suivantes :

- L'intégration de techniques de traitement des langues naturelles notamment au niveau d'interface utilisateur, ceci afin de rendre le système plus performant.
- Prendre en compte l'aspect d'adaptabilité des agents.
- Utiliser des mécanismes de coopération entre les agents mobiles afin de réaliser effectivement les tâches demandées.
- Etendu l'architecture à l'usage de Services Web (WS).

Références

1. J. P. Arcangeli, V. Hennebert, S. Leriche, F. Migeon, M. Pantel : Tutoriel JavAct (version 0.5.0), IRIT (2003).
2. J. P. Arcangeli, V. Hennebert, S. Leriche, F. Migeon, M. Pantel : JavAct 0.5.0 : principes, installation, utilisation et développement d'applications. Rapport de recherche, IRIT/2004-5-R, Février (2004).
3. X. Baril : Un modèle de vues pour l'intégration de sources de données XML : VIMIX. Thèse de doctorat présentée à l'Université des Sciences et Techniques du Languedoc. Décembre (2003).
4. B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, D. Rus: Mobile agents in distributed information retrieval. In Intelligent Information Agents. Springer-Verlag (1999).
5. V. Camps, M. P. Glezles : Une technique multi-agent pour rechercher des informations réparties. Institut de Recherche en Informatique de Toulouse. Actes des cinquièmes Journées Francophones IAD et SMA, Editions Hermès, Avril (1997).
6. M. Côté, N. Troudi : NetSA : Une architecture multiagent pour la recherche sur Internet. Université Laval. Département d'informatique. Pavillon Pouliot. Ste-Foy, Canada (1998).
7. S. E. Falou : Programmation répartie, optimisation par agent mobile. Thèse du Doctorat de l'université de CAEN, soutenue le 29 Novembre (2006).
8. F. Freitas, B. Espinasse, S. Fournier : AGATHE: une architecture générique à base d'agents et d'ontologies pour la collecte d'information sur domaines restreints du Web (2007).
9. H. Kosch, M. Doller, L. Boszormenyi: Content-based indexing and retrieval supported by mobile agent technology (2001).
10. S. Papastavrou, G. Samaras, E. Pitoura: Mobile Agents for World Wide Web Distributed Database Access, IEEE Transactions on Knowledge and Data Engineering, (2000).
11. S. J. Pelletier, S. Pierre, H. H. Hoang: Modeling a Multi-Agent System for Retrieving Information from Distributed Sources. CIT "Computing and Information Technology", (2003).
12. N. Travers: Optimisation Extensible dans un Médiateur de Données Semi Structurées. Thèse du Doctorat, soutenue le 12 Décembre (2006).