SMAIDoC : Un Système Multi-Agents pour l'Intégration des Données Complexes

F. Clerc, A. Duffoux, C. Rose, F. Bentayeb, O. Boussaid

Bâtiment L, 5 avenue Pierre-Mendès-France - France
69676 BRON Cedex – FRANCE
{frederic.clerc, amandine.duffoux, christian.rose}@etu.univ-lyon2.fr,
bentayeb@eric.univ-lyon2.fr, boussaid@univ-lyon2.fr
http ://bdd.univ-lyon2.fr

Résumé. L'expansion du World Wide Web et la multiplication des sources de données conduisent à la prolifération de données hétérogènes (textes, images, vidéos, sons, bases de données) appelées données complexes. Pour explorer ces données, il est nécessaire de procéder à leur intégration dans un format unifié. Cette intégration présente des difficultés de collecte, de structuration et de stockage des données. Plusieurs approches sont apparues dans la littérature pour l'intégration de données, parmi lesquelles celle des médiateurs-wrappers ou de l'entreposage. Dans cet article, nous proposons une nouvelle démarche d'intégration de données complexes qui repose à la fois sur l'approche classique d'entreposage de données et sur l'utilisation des systèmes multi-agents. Cette nouvelle approche se base sur une architecture évolutive grâce à la technologie "agent". Les différentes étapes de la phase d'intégration sont alors considérées comme des tâches assimilées à des services, gérées par des acteurs assimilés à des agents. Pour valider cette approche, nous avons développé le "Système Multi-Agent pour l'Intégration de Données Complexes", SMAIDoC.

Mots-Clés : intégration de données, données complexes, services, agents, Système Multi-Agents

1 Introduction

Les technologies de l'entreposage de données et de l'analyse en ligne ont largement fait leurs preuves dans le domaine de la gestion des données et notamment lorsque celles-ci sont numériques. Avec l'avènement d'Internet, la prolifération des données de nature variée (image, texte, sons, vidéo, bases de données...), appelées données complexes, s'est accrue. Ces données nécessitent l'utilisation d'un outil performant facilitant la phase de leur intégration, qui comporte plusieurs étapes (acquisition, structuration et stockage des données), en vue de leur exploitation.

Dans ce contexte, nous proposons une nouvelle approche d'intégration des données complexes basée sur un Système Multi-Agents (SMA). Nous assimilons les différentes étapes de cette phase d'intégration, techniquement ardues, à des services offerts par des agents :

– collecte des données : cette étape est gérée par des agents dont le rôle consiste à récupérer les caractéristiques des données pour pouvoir les transmettre ensuite

aux agents responsables de la structuration des données ;
- structuration des données : dans cette étape, les agents s'occupent de l'organisation des données selon un modèle bien défini et transmettent ce dernier aux agents responsables du stockage ;
- stockage des données : cette étape est gérée par des agents qui s'occupent de l'alimentation d'une base de données à partir du modèle fourni par les agents de structuration.

Pour valider cette approche, nous avons développé le "Système Multi-Agent d'Intégration de Données Complexes" SMAIDoC (BDD-ERIC 2003). Ce système propose un ensemble d'agents intelligents offrant les différents services nécessaires à l'intégration des données complexes. SMAIDoC permet de mettre à jour les services existants, d'en rajouter, ou de créer de nouveaux agents. Cette approche repose ainsi sur une architecture évolutive offrant une grande flexibilité et une forte structuration.

Cet article est organisé de la façon suivante. La section 2 présente un état de l'art concernant les approches d'intégration de données, ainsi que sur la technologie des agents. Dans la section 3, nous exposons la problématique de l'intégration de données complexes et en quoi l'utilisation d'un SMA est appropriée pour réaliser cette tâche. Dans la section 4, nous présentons les principes de SMAIDoC. L'implémentation de celui-ci est décrite dans la section 5. Enfin, nous concluons cet article et nous proposons des perspectives d'évolution dans la section 6.

2 État de l'art

2.1 Intégration de données

La nature des données présentes sur le Web est généralement hétérogène (bases de données relationnelles ou objets, documents balisés ou non, images, sons, vidéo). La complexité et la prolifération de ces données posent le problème de leur intégration. Plusieurs approches traitant de ce problème ont émergé.

Dans l'approche basée sur les médiateurs (M.C. Rousset 2002), il est question de maintenir les données dans leurs sources d'origine et de construire des vues abstraites à partir desquelles un médiateur tente de satisfaire des requêtes d'utilisateurs. L'architecture courante d'une telle démarche repose sur des systèmes médiateur-adaptateurs (F. Goasdoué et Al 2000). Ce type de système permet d'interroger des sources de données distribuées et hétérogènes. Il donne l'illusion d'un système centralisé et homogène. Le médiateur réalise l'intégration des données en fournissant une vue homogène et globale du système à l'utilisateur. Sa tâche est de reformuler les requêtes posées par l'utilisateur en fonction des différents contenus des sources de données accessibles (S. Cluet 1999). A ce médiateur correspondent plusieurs adaptateurs, un pour chaque source de données. Leur rôle est d'extraire les données sélectionnées par la requête selon le type de la source de données correspondante. Le principal intérêt de ce système est la reformulation des requêtes par le médiateur (C. Reynaud et Al 2002). En effet, celui-ci fournit à l'utilisateur une certaine souplesse quant à la rédaction et à la formulation des requêtes. De plus, l'approximation de requêtes est intégrée au système afin de permettre

au médiateur d'affiner la requête de l'utilisateur et lui permettre d'obtenir toujours une réponse.

L'approche basée sur la technologie de l'entreposage des données (R. Kimball 1996, W.H. Inmon 1996) consiste à construire à partir de différentes sources de données une nouvelle base appelée entrepôt de données. Dans ce cas, l'intégration correspond au processus d'ETL (*Extracting Transforming and Loading*) chargé de nettoyer et de transformer les données qui sont hétéroclites, avant leur chargement dans l'entrepôt. Ce dernier s'appuie sur un modèle orienté analyse, où les données représentent des indicateurs (mesures) que l'on peut observer selon des axes d'analyse (dimensions). Le modèle est multidimensionnel et caractérise un contexte d'analyse. Les requêtes sont en général complexes (V. Harinarayan et Al 1996). Elles nécessitent des traitements pour résumer les données et faciliter leur interprétation, des moyens sophistiqués pour y naviguer à travers les différentes dimensions et ce à l'aide des opérateurs OLAP (*On Line Analytical Processing*) (E.F Codd 1993). Les réponses aux requêtes de l'utilisateur sont représentées dans des cubes de données où les informations sur les indicateurs sont agrégées et visualisées selon des points de vue différents (S. Chaudhuri et Al 1997).

2.2 Les systèmes multi-agents

Un agent logiciel est un programme classique qualifié d'"intelligent". Les agents intelligents sont utilisés dans de nombreux domaines tels les réseaux (ANT : S. Hassas et Al 2000), les technologies embarquées (LEAP : *Lightweight Extensible Agent Platform* (Leap 2002, B. Burg 2000), ou l'apprentissage humain (Baghera : S. Pesty et Al 2001). Un agent intelligent est supposé avoir les caractéristiques intrinsèques suivantes (M. Wooldridge et Al 1995) : *Intuitivité*, un agent doit être apte à prendre des initiatives et à accomplir les actions qui lui sont imparties ; *Réactivité*, un agent doit être à l'écoute des actions de son environnement et agir en conséquence ; *Sociabilité*, un agent doit être capable de communiquer avec d'autres agents et/ou des utilisateurs. Par ailleurs, les agents peuvent être mobiles et posséder la particularité de se mouvoir de façon autonome à travers un réseau accepteur afin de réaliser diverses tâches (K. Sykara et Al 1995).

Un Système Multi-Agents désigne un ensemble plus ou moins étendu d'acteurs qui communiquent entre eux (K. Sykara et Al 1996). L'ensemble de cette communauté a pour but la réalisation d'une tâche bien précise, où chacun a un objectif propre et offre des services. C'est ainsi que l'on introduit la notion de service : chaque agent est capable de réaliser certaines tâches, de façon autonome et communique les résultats obtenus à un acteur récepteur (humain ou logiciel). Les SMA se doivent de respecter les normes de programmation définies par la FIPA (*Foundation for Intelligent Physical Agents*) (fipa 2002).

3 Intégration des données complexes

L'exploration des données complexes nécessite une phase de pré-traitement afin de les rendre accessibles aux outils classiques de stockage et d'analyse. Il existe plusieurs approches qui associent la technologie des entrepôts de données avec celle du Web

comme le *data webhousing* (R. Kimball et Al 2000) tandis que d'autres s'articulent sur des architectures distribuées (J.F. Goglin 1998).

La démarche que nous proposons considère le Web comme une source principale d'alimentation en données d'un dispositif basé sur la notion de services pour le stockage et l'exploitation de données complexes. Nous avons fait le choix de donner la possibilité à l'utilisateur de cibler ses données et de les rapatrier pour construire une source locale de données. La complexité des données est essentiellement due à la diversité de leur type. D'autre part, elles renferment des informations sur leur structure ainsi que sur leur contenu. Nous avons proposé dans (J. Darmont et Al 2002a) une modélisation conceptuelle, logique et physique de ces données complexes. Celles-ci sont définies comme des objets complexes formés d'un ou de plusieurs sous documents pouvant avoir des types différents. Ces objets sont représentés par un diagramme de classes UML (Fig 1). Ce modèle UML est général et renferme les descripteurs de bas niveau qui constituent les meta données (S. Miniaoui et Al 2001). Il est appelé à être complété par une partie spécifique pour contenir des descripteurs sémantiques. Toute donnée complexe recueillie sur le web ou en provenance de toute autre source peut être représentée à l'aide du modèle conceptuel (Fig 1). De plus, grâce à une DTD XML, le modèle est traduit ensuite sous forme d'un document XML pouvant être stocké dans une base de données relationnelle (J. Darmont et Al 2002b).

4 Un système multi-agents pour l'intégration des données complexes (SMAIDoC)

4.1 Motivation

L'intégration de données complexes consiste en une suite de tâches à effectuer. Nous assimilons celles-ci, qui ne sont pas obligatoirement séquentielles, à des services offerts par des acteurs bien définis dans un système destiné à accomplir un tel processus d'intégration. Dans cette optique, il est naturel d'introduire la notion d'agent. La technologie agent apporte une structuration claire, modulaire, et donc une réutilisation aisée et une maintenance idéale. Par conséquent, cette structuration permet d'ajouter très facilement des services ou des agents au système (M. Klusch 2001). En outre, les agents sont capables de communiquer entre eux sans intervention externe. Les échanges d'informations sont alors simplifiés et l'exécution des tâches est clarifiée, chaque agent effectuant un ensemble de tâches bien défini. De plus, le concept de mobilité est un aspect très important. En effet, l'agent peut se déplacer là où se trouve l'information et revenir à son emplacement initial. Il pourra ainsi permettre à terme de créer de façon autonome des bases de données distantes, associées à d'autres agents capables de fournir des services conséquents à l'utilisateur, en vue de la collecte d'informations pertinentes depuis le Web, mais aussi à travers un réseau local, par exemple.

4.2 Architecture de SMAIDoC

Nous présentons dans cette section l'architecture globale de notre système SMAIDoC (Fig 2) basée sur une plate-forme d'agents génériques. Nous avons instancié cinq

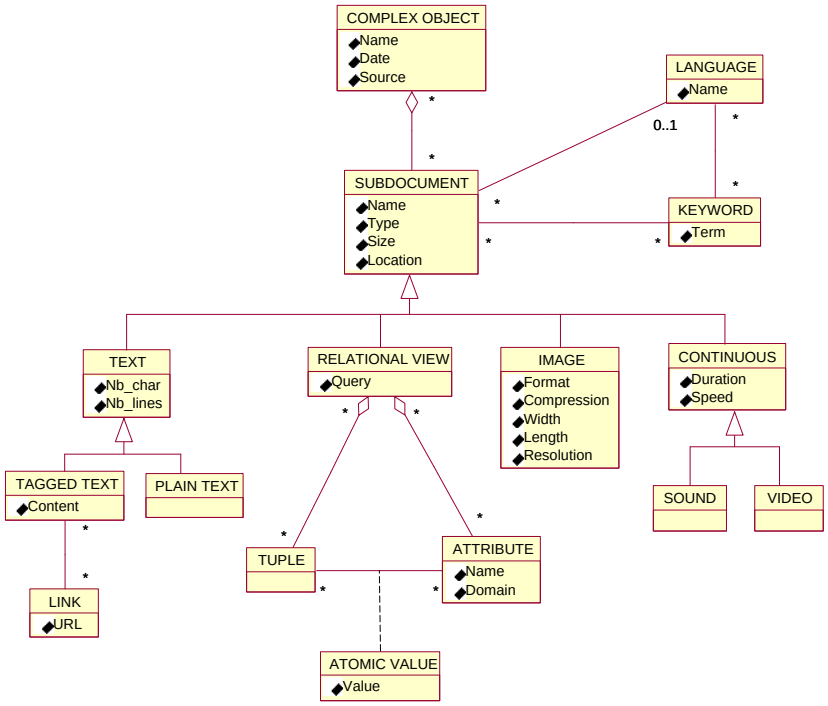

FIG. 1 – Modèle conceptuel d'un objet complexe

agents offrant des *services* permettant l'intégration des données complexes où chacun de ces services engendre des *produits*.

4.2.1 Les agents

- *MenuAgent :* C'est le pilote du système. Il sert d'interface entre ce dernier et l'utilisateur. Il a pour rôle de superviser les migrations d'agents. Il présente le répertoire des différents sites accessibles sur la plate-forme.
- *DataAgent :* Il a pour rôle de collecter les données relatives aux documents.
- *WrapperAgent :* Il instancie la structure UML sur la base des données transmises par le DataAgent.
- *XMLCreatorAgent :* Il construit les documents XML à partir d'une DTD et de la structure UML.
- *XML2RDBAgent :* Il stocke les documents XML dans une base de données relationnelle.

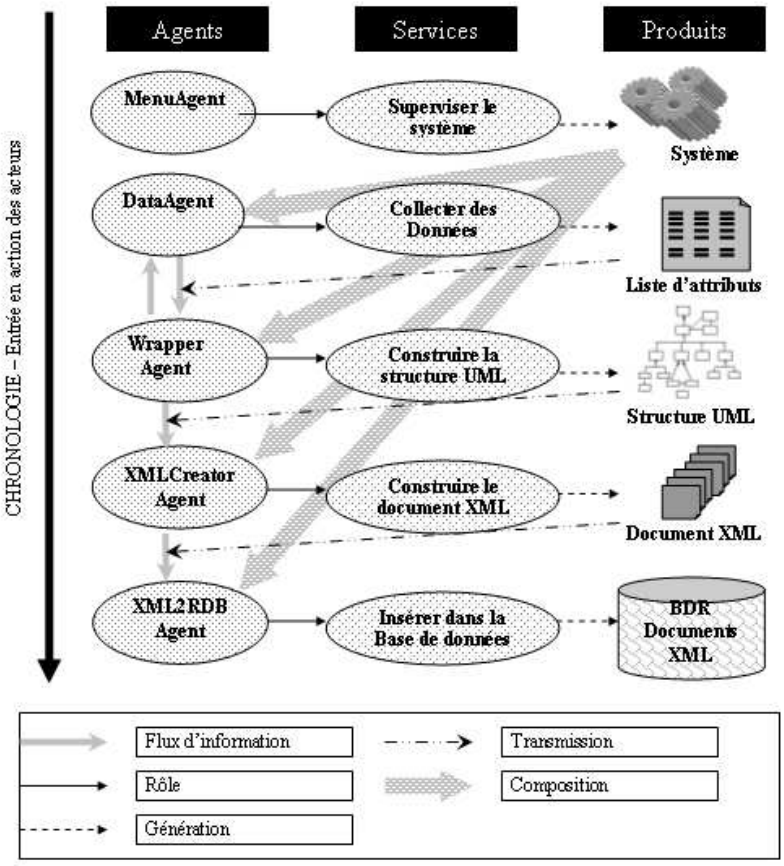

FIG. 2 – Architecture du système SMAIDoC

4.2.2 Les services

L'ensemble des tâches de la phase d'intégration des données complexes peut se dérouler en quatre grandes étapes.

- *Collecte* : A partir de documents contenant des données complexes, l'extraction des caractéristiques de celles-ci consiste en une collecte des méta données . Ces dernières sont formulées comme des attributs pouvant être, par exemple, les mots clés d'un texte, la durée d'une séquence vidéo ou le taux de compression d'une image.
- *Structuration* : A partir des données précédemment collectées, il faut instancier le modèle général prédéfini (Fig 1). Ceci permettra de générer un modèle conceptuel décrivant les données complexes.
- *Génération* : A partir du modèle instancié et à l'aide d'une DTD XML, les données

complexes sont traduites sous forme de documents XML bien formés.
– *Stockage* : A partir des documents XML bien formés, la dernière étape consiste à les stocker dans une base de données relationnelle.

4.2.3 Les produits

Lorsque le système SMAIDoC est activé, il commence par initialiser l'environnement dans lequel les agents vont évoluer (Fig 3). Les agents du système SMAIDoC opèrent sur des données complexes déjà collectées. L'extraction des caractéristiques de celles-ci est un résultat obtenu par un des agents et sert à alimenter les différentes classes du modèle UML. Les documents XML générés sont des produits servant à alimenter une base de données relationnelle. Cette dernière constitue le produit final fourni par SMAIDoC.

4.3 Structure fonctionnelle d'un agent

Pour décrire la structure et le fonctionnement d'un agent de SMAIDoC, nous proposons une grammaire exprimée par un formalisme basé sur les prédicats. Un agent est composé de deux parties : l'entête et le corps. L'entête identifie l'agent (ligne 2) par son nom ainsi que celui de la plate-forme à laquelle il appartient (ligne 3). Le corps de l'agent spécifie les scénarios possibles : tout d'abord l'enregistrement auprès de la plate-forme de l'agent (ligne 6), puis la définition du comportement (s'il possède un comportement spécifique) (ligne 7), la mobilité de l'agent le cas échéant et sa migration (lignes 9,10,11) et la communication inter-agent (lignes 12,13,14). Ces actions sont exécutées les unes après les autres de manière itérative.

Description de la structure et du fonctionnement d'un agent à l'aide d'une grammaire

1. < *Agent* >::=< *Entête_Agent* >< *Corps_Agent* >
2. < *Entête_Agent* >::=< *Agent_ID* >
3. < *Agent_ID* >::=< *Nom_Agent* >< *Nom_plateforme* >
4. < *Corps_Agent* >::=< *Configuration_Agent* >
5. < *Configuration_Agent* >::=< *Enregistrement* >< *Ajouter_Comportement* > < *Mobilité*>< *Communication* >< *Configuration_Agent* >
6. < *Enregistrement* >::= Si pas déjà fait Ajouter < *Agent_ID* > dans < *Registre_Plateforme* >
7. < *Ajouter_Comportement* >::=Ajouter < *Comportement* >à< *Agent_ID* > |*Action_Vide*
8. < *Comportement* >::=tâche spécifique accomplie de façon autonome
9. < *Mobilité*>::=< *Ontologie_Mobilité*>< *Migrer* >
10. < *Ontologie_Mobilité*>::=Ajouter < *Agent_ID* > dans < *Registre_Agents_Mobiles_Plateforme* >
11. < *Migrer* >::=Envoyer < *Agent_ID* > vers < *Adresse_hote* >
12. < *Communication* >::=< *Recevoir_Message* > | < *Envoyer_Message* > |*Attente_Active*
13. < *Recevoir_Message* >::=< *Attente_Message* >< *Lire_contenu_Message* >
14. < *Envoyer_Message* >::= Envoyer < *Message* >à< *Agent_ID_Destination* >
15. < *Lire_contenu_Message* >::= Lire < *Message* >
16. < *Attente_Message* >::= Attendre < *Message* >

4.4 Fonctionnement des différents agents de SMAIDoC

Pour illustrer le fonctionnement de SMAIDoC, considérons la situation suivante : soit un site S contenant différents types de données complexes.

- *Étape 1 :* Sélection du site S par l'utilisateur et ordre de migration des agents *DataAgent* et *WrapperAgent*.
- *Étape 2 :* *MenuAgent* ordonne à ces deux agents de migrer. La migration s'effectue.
- *Étape 3 :* Collecte séquentielle des méta données par *DataAgent* et instanciation de la structure UML par *WrapperAgent*. Suivant le type des données, *DataAgent* collecte les attributs et les transmet séquentiellement à *WrapperAgent* en vue de la construction progressive de la structure UML.
- *Étape 4 :* Transmission de la structure à l'agent *XMLCreator*.
- *Étape 5 :* *XMLCreator* construit le document XML à partir de la structure UML et de la DTD qui lui est associée. Il parcourt et empile les éléments. Suivant la forme de la structure, il compare les éléments empilés avec les composantes de la structure. Il génère le code XML correspondant.
- *Étape 6 :* Transmission du document XML à *XML2RDBAgent*.
- *Étape 7 :* *XML2RDBAgent* ordonne le stockage du document XML dans la base de données.
- *Étape 8 :* Retour à l'étape 1.

5 Réalisation

Bien plus qu'un choix, l'utilisation du langage JAVA (java 2002) est une évidence compte-tenu des spécificités de notre application. Il est portable, la présence d'une machine virtuelle étant la seule obligation. De nombreuses plate-formes de programmation d'agents sont disponibles sous JAVA. Nous avons utilisé la version 2.61 de JADE (jade 2002, F. Bellifemine et Al 1999) qui respecte les normes définies par la FIPA. JADE propose de construire des structures d'accueil des agents : les *containers*. Ce sont des éléments abstraits qui peuvent se situer sur l'une ou l'autre machine, mais toujours sur une même instance de la plate-forme JADE. Par ailleurs, on note la présence de deux agents pilotes par défaut. D'une part, l'agent *AMS* (*Agent Management System*) a pour rôle de superviser la gestion des agents présents sur la plate-forme : leur état, leur emplacement, leurs migrations, les sites d'accueil disponibles... D'autre part, l'agent *RMA* (*Remote Management Agent*) a pour rôle de fournir une interface pour JADE. De plus, nous avons défini un nouveau pilote, *MenuAgent*, qui permet à l'utilisateur de sélectionner un site -parmi ceux disponibles- contenant des données complexes destinées à l'intégration. Une fois le site sélectionné, le pilote *MenuAgent* ordonne la migration des agents *DataAgent* et *WrapperAgent* vers ce site et ainsi le traitement peut commencer.

Pour communiquer, les agents de JADE utilisent un protocole nommé ACLMessage[1]

[1] standardisé par la FIPA

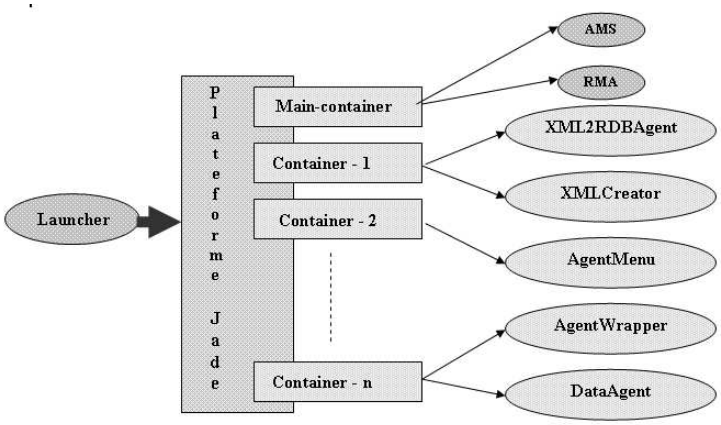

FIG. 3 – Vue d'ensemble de la plate-forme JADE

(J. Pitt et Al 1999). Les messages échangés par les agents de JADE utilisent une ontologie en format sérialisable (texte). Dans JADE, une ontologie est représentée sous forme de classe JAVA générée automatiquement suite à la demande d'envoi de message. Ce mode de communication peut être apparenté à l'association de SOAP (*Simple Object Access Protocol*) et de RDF (A. Andjomshoa et Al 2001).

Dans l'objectif de faciliter la mise en place du système, son démarrage et son initialisation, nous avons créé un lanceur (*"launcher"*) qui a pour rôle de remplacer l'utilisateur dans cette tâche fastidieuse. Le lanceur crée la plate-forme JADE et les containers virtuels d'accueil des agents, puis initialise les agents pilotes du système (RMA et DMS). On peut avoir une vue d'ensemble du système la Fig 3.

Concrètement, notre système SMAIDoC permet à l'utilisateur de sélectionner les documents qu'il souhaite intégrer dans une base de données. Les types de documents suivants sont proposés : texte simple, texte balisé (HTML, XML...), image, son, vidéo, vue relationnelle. Pour chacun des types, des attributs par défaut sont proposés à l'utilisateur comme les mots-clés, le nombre de mots et de lignes pour un texte, la durée pour un son ou une vidéo, la résolution pour une image...

Les données complexes recueillies à partir d'un quelconque site sont intégrées dans une base de données selon le processus décrit auparavant. Les diverses tâches sont accomplies par les agents de SMAIDoC. Ces derniers évoluent dans les différents containers de la plate-forme JADE que nous avons définis sur une même machine hôte. Ces containers peuvent être implantés sur différentes machines hôtes engendrant ainsi une mobilité inter-machine des agents de SMAIDoC.

6 Conclusion et perspectives

Dans cet article, nous avons proposé une nouvelle approche d'intégration de données complexes basée à la fois sur l'entreposage de données et sur les systèmes multi-agents.

Cette approche repose sur une architecture évolutive dans laquelle on peut ajouter, modifier ou supprimer des services, voire créer de nouveaux agents.

Le système SMAIDoC aborde l'intégration de données complexes déjà recueillies. SMAIDoC, grâce aux agents *DataAgent* et *WrapperAgent*, modélise ces données sous forme d'objets complexes. Ensuite, l'agent *XMLCreator* les traduit en documents XML pour que l'agent *XML2RDBAgent* les stocke dans la base de données. Ces différents services effectués par cet ensemble d'agents constituent les différentes tâches du processus d'intégration des données complexes dans une base de donnés (ETL). Notons que les agents, en plus de leur flexibilité, présentent l'avantage d'être mobiles.

Notre objectif consiste à étendre les possibilités de SMAIDoC aux tâches de recueil et d'analyse des données complexes. Cet objectif peut être atteint grâce à l'architecture évolutive de SMAIDoC. Il est possible de donner à l'agent de collecte de données (*DataAgent*) la capacité de recueillir des données en conversant avec des moteurs de recherche et d'exploiter les réponses de ces derniers. D'autre part, nous pouvons créer de nouveaux agents dont les services respectifs peuvent être la modélisation multidimensionnelle des données complexes ou encore l'analyse à l'aide de techniques OLAP (S. Chaudhuri et Al 1997) ou de fouille de données.

Références

A. Andjomshoa and M.H. Shafazand and S.Bahonar, (2002), "The application of software agent technology to project management infrastructure", 4th international conference on Information Integration and Web-based Applicaions and Services (iiWAS'02), Bandung, Indonesia.

BDD-ERIC (2003), SMAIDoC download page, *http ://bdd.univ-lyon2.fr/logiciels.php ?-id=4*.

F. Bellifemine and A. Poggi and G. Rimassa, (1998), "JADE : A FIPA compliant agent framework", PAAM'99, London pp 97-108.

B. Burg, (2000), "Towards the deployment of an open Agent World", Journées Francophones d'Intelligence Artificielle Distribuée et de Systèmes Multi-Agents (JFIAD-SMA), pp 1-17.

S. Chaudhuri and U. Dayal, (1997), "An overview of data Warehousing and OLAP Technology", ACM-SIGMOD, Record 26(1).

S. Cluet, (1999), "Intégration de données hétérogènes", *http ://www-rocq.inria.fr/-cluet/*.

E.F Codd, (1993), "Providing OLAP (on-line analytical processing) to user-analysts : an IT mandate", Rapport, E.F. Codd and Associates.

J. Darmont and O. Boussaid and F. Bentayeb and S. Rabaseda and Y. Zellouf, (2002a), "Multimedia Systems and Applications",Web multiform data structuring for warehousing, Kluwer Academic Publishers, Vol 22, pp 9-27.

J. Darmont and O. Boussaid and F. Bentayeb, (2002b), "Warehousing Web Data", 4th International Conference on Information Integration and Web-based Applications and Services (iiWAS'02), Bandung, Indonesia, pp 148-152.

Le site officiel de la FIPA, *http ://www.fipa.org/*.

F. Goasdoué and V. Lattès and M.C. Rousset, (2000), "The use of CARIN language and algorithms for information integration :the PICSEL system", International Journal of Cooperative Information Systems IJCIS, vol 9, n°4, pp 383-401.

J.F. Goglin, (1998), "La construction du Data Warehouse du data mart au data web", Hermès.

V. Harinarayan and A. Rajaraman and J.D. Ullman, (1996), "Implementing Data Cubes Effciently", IEEE on Data Engineering.

S. Hassas and S. Fenet, (2000), "ANT : a distributed problem solving framework based on mobile agents", Advances in Mobile Agents System Research, vol 1, n°4, pp 39-44.

W.H. Inmon, (1996), "Building the data Data Warehouse", John Wiley & Sons, USA, 2ième édition.

Le site officiel de JADE, *http ://sharon.cselt.it/projects/jade/*.

Le site officiel de JAVA, *http ://www.sun.java.com/*.

R. Kimball,(1996), "The data warehouse toolkit", John Wiley.

R. Kimball and R. Merz, (2000), "The data webhousing", Eyrolles.

M. Klusch,(2001), "Information Agent technology for the Internet : A Survey", Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration, D. Fensel (Ed.), vol 36, n°3.

Le site officiel de LEAP, *http ://leap.crm-paris.com/*.

S. Miniaoui and J. Darmont and O. Boussaid, (2001), "Web data modeling for integration in data warehouses", First International Workshop on Multimedia Data and Document Engineering (MDDE 01), Lyon, pp 88-97.

S. Pesty and C. Webber and N. Balacheff, (2001), "Baghera : une architecture multi-agents pour l'apprentissage humain", Agents Logiciels, Coopération Apprentissage et Activité Humaine (ALCAA), pp 204-214.

J. Pitt and F. Bellifemine, (1999), "A Protocol-Based Semantics for FIPA 97 ACL and its Implementation in JADE", AI*IA'99.

C. Reynaud and G. Giraldo, (2002), "Vers l'automatisation de la construction de systèmes de médiation pour le commerce électronique", Journées de l'action spécifique 'Web Sémantique', Paris.

M.C. Rousset, (2002), "Knowledge Representation for Information Integration", International Symposium on Methodologies for Intelligent Systems (ISMIS).

K. Sykara and D. Zeng, (1995), "Cooperative Intelligent Software Agents", Rappot n°CMU-RI-TR-95-14, Robotics Institute Technical Report.

K. Sykara and D. Zeng and K. Decker and A. Pannu and M. Williamson, (1997), "Distributed Intelligent Agents", IEEE Expert, pp 1-32.

M. Wooldridge and N.R. Jennings, (1995), "Intelligent Agents : Theory and Practice", Knowledge Engineering Review, vol 10, n°2, pp 115-152.

Summary

The expansion of the World Wide Web and the multiplication of data sources lead to the proliferation of heterogeneous data (texts, images, videos, sounds, databases) that are called complex data. In order to explore them, we need to carry out their integration into a unified format. This integration tasks highlights several issues when collecting, structuring and storing complex data. Many methods for integration exist in the litterature, such as mediated schemes with wrappers and warehousing. In this paper, we propose a new approach for complex data integration that exploits both classical warehousing and multi-agents system. The multi-agents technology provides an evolutive architecture to our system. We consider the different tasks of the data integration process as services that are offered by actors called agents. To validate this approach, we have designed a multi-agents system for complex data integration called SMAIDoC.

Keywords : data integration, complex data, services, agents, Multi-Agent System