AN AGENT BASED INFORMATION SYSTEM FOR COMMUNITIES MEDIATION

Aluizio Haendchen Filho
Famesul – Uniasselvi

Hércules Antonio do Prado
Embrapa Food Technology, Catholic University of Brasília

Míriam Sayão
PUC-RS

Fénelon do Nascimento Neto
Embrapa Food Technology

Keywords: Enterprise Information Systems, Communities Mediation, Agent Based Systems.

Abstract: The adoption of the Multi-Agent System paradigm in the context of Enterprise Information Systems has been accelerated by the technology brought by Internet. The importance of MAS applications increases as the ubiquity of Internet, with its distributed and interconnected elements, becomes a *de facto* reality. However, the development of MAS is not trivial; agents-based systems are typically complex and difficult to develop due to the features required, some of them hard to implement. In this paper we briefly describe MIDAS, a service oriented (SOA) framework built on a reusable, adaptable and loosely coupled architecture, that aims to help in the development of MAS applications. An application in the domain of expert/customer mediation is presented to evidence the advantages of the framework. After that, the advantages of applying the SOA standard over the traditional message-based approach for MAS development are discussed.

1 INTRODUCTION

Due to the recent availability of Web-based development technology, the application domain of Multi-Agent Systems (MAS) has spread out. This kind of system has been applied in many areas in the Internet context: e-Commerce, Web Services (Finin, 1994), Knowledge Management (Kendall et al., 1999), Semantic Web (Decker et al., 2000), and Information Systems in general (Adam et al., 2004 and Jennings et al., 1996). The adoption of MAS in the application development over Internet has enabled interesting solutions to B2B, e-Business (Boughaci et al., 2005), and also applications that requires interoperability based on knowledge about applications and business processes. Klusch (1999) identifies *information intelligent agents* as one of the most promising areas for applying agents technology. Information agents acts in fields like collaborative systems over Internet, knowledge discovery from heterogeneous sources, systems for intelligent management of information for intranet of Internet, among others.

The importance of Internet as a common and ubiquitous systems environment has led to efforts like MIX (Mediation of Information using XML) (Baru et al., 1999), that uses agents to integrate information distributed in disparate sources. The Web also can be seen as a big distributed database having XML (and its extensions or modifications) as an underlying data model. Agents can naturally perform the roles required in a mediation of information process. In this paper, we apply MIDAS (Haendchen Filho, 2006), a platform to develop MAS, to build an

Haendchen Filho A., Antonio do Prado H., Sayão M. and do Nascimento Neto F. (2007).
AN AGENT BASED INFORMATION SYSTEM FOR COMMUNITIES MEDIATION.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - SAIC*, pages 295-298
DOI: 10.5220/0002388302950298

application to mediate the communication among a community of experts in food technology and customers in need for qualified information about food products in supermarkets. For this propose, we define a *mediator* agent that perform the information interchange between these communities, involving tasks like collecting questions, searching for a consent among experts, providing translations (with the support of communication specialists), and distributing the answers.

2 MEDIATION AMONG COMMUNITTIES

The dialogue among communities with different jargon can be seen as a sub problem of the cross lingual dialogue problem, a research field that looks for highly accurate communication of semantically complex content (Piwek and Power, 2006). The 'Saint Graal' for researchers in this field is to build an automatic process to translate one language to other, being able to cope with the different semantic levels of each specific language. A solution to this problem is much more important if you consider the challenge to human-to-human communication through a browser that has been set by the Internet. To build an effective automatic translator for this scenery is a dream far to be realized. However, the necessity of communication among communities is a reality that requires, if not the best solution, at least one viable solution. The solution we present uses the agents technology to facilitate the negotiation among the communities members.

3 MIDAS DESCRIPTION

MIDAS (Middleware for Intelligent and Distributed Agent-based Systems) (Haendchen Filho, 2006) is a platform that provides an environment to run agents and a WSA-compliant framework to ease its development. WSA aims to provide a common Web Service definition and its location inside a wider architecture in order to guide service implementers, authors of services specification and Web application developers. WSA represents the natural evolution from traditional applications to SOA ones (Odell, 2005).

Figure 1 shows the MIDAS generic architecture, which is based on the coexistence of several containers, each one executing a JVM (Java Virtual Machine). Each virtual machine provides a complete execution environment, where agents can execute concurrently in the same host. The architecture includes two different types of container: the Front-End Server (FES) and the Agent Container (AC). FES plays the integration rules of the platform, promoting the synchronization services and interoperability with external applications. It is similar to the front-end server used by JADE (Adam et al., 2004). AC is a Web container that can be inhabited by organizations, agents and components.

Middleware and application agents are the basic elements of the platform. The middleware agents abstract completely generic characteristics, such as communication, concurrency, lifecycle management, services discovery and interoperability. They enable the developer to focus only in the application details. The introduction of the agent concept to play these roles makes easier to satisfy important non-functional requirements for Web architectures, like flexibility, dynamic behavior, and adaptability. The following middleware agents have been used to perform the roles defined by the WSA reference model,: (i) a Broker agent, playing the MOM roles; (ii) a Proxy agent, playing the SOM roles; (iii) a Catalog agent, playing the ROM roles and (iv) a Manager agent, playing the MGM roles.

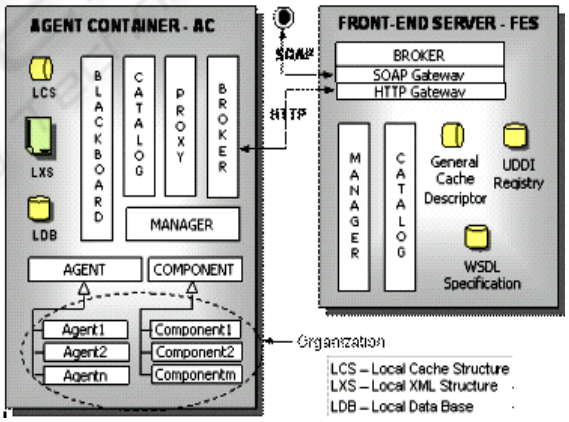

Figure 1: The main concepts of the MIDAS.

Application agents and components are only located in the AC containers. AC provides a structure composed by abstract classes (Agent and Component) and a blackboard. The introduction of the abstract agent concept extends the WSA reference architecture specification, providing a way to group in a super-class the common properties to all the agents. The application agents (or components) are implemented by extending the abstract classes, which provide the hot-spots from which specific

applications behavior can be implemented. Application agents and components are instantiated in Organizations, and the Blackboard agent offers a powerful mechanism to support for the agents communication model.

Besides agents and components, the platform carry resources representations, which can be stored in XML files, relational databases or documents in general. The General Cache Descriptor element located in the FES represents the platform-consolidated structure of resources, kept in cache memory. In the AC container, the LXS element represents the services specification described in XML. The resources structure is also kept in cache memory, being represented by the LCS element. LDB represents local databases, which can be handled and/or accessed by agents and components. The Web Services specifications and registries are represented by the WSDL and UDDI elements.

4 EXPERT/CUSTOMER MEDIATION

The problem approached in this paper can be visualized in Figure 2. C_1, C_2, .., C_n represent the customers requiring specialized information in non-technical language. E_1, E_2, …, E_m represent the experts that can provide technical answers to the customers questions. The triangle represents a set of communication specialists that are in charge of translating the technical language from the experts to the customer language. In order to enable its reuse, the answers are stored in an *answers server*. Notice that the information required by customers are those not supplied by the product labels. For instance, a customer may be interested in knowing about the effects on human health of a cereal produced from GMO (Genetic Modified Organism) grain, or the difference from organic to hydroponics lettuce. In this case, a *mediator* agent will be responsible for building a consensus among the specialists addressed to provide a sound technical answer to the customer. This answer will be translated by the communication team and sent back to the customer and stored in the answers server. The application developed is named BeyondTheLabel (BL).

Figure 3 shows a partial of BL structure, as provided by one of the GUIs wizards in MIDAS. The right side panel shows the resource representation, in the *structure* mode. The BeyondTheLabel element is the root node, from what the agents and components are instantiated. It can be seen, in the panel, the components wrapper AreaData, InstitutionData, ResearcherData, that encapsulate Web Services required by the application. The component InstitutionSearch encapsulate two Web services: *searchInstitutionByName* and *searchInstitutionByRegion*, that retrieve institutions by name of region. The left side panel shows specification details of *searchInstitutionByName* service: service name, address, parameters and description.

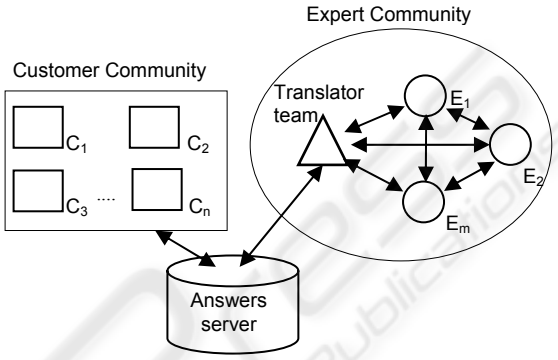

Figure 2: Interaction among communities.

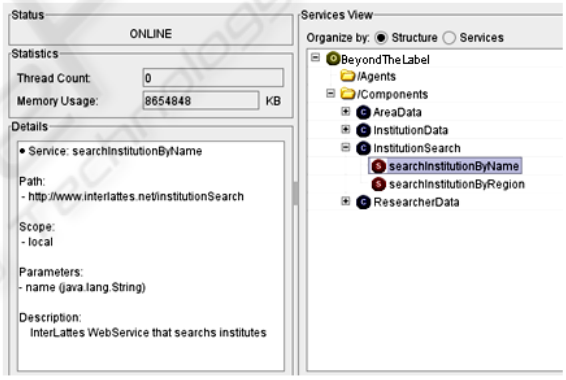

Figure 3: BL container partial view provided by the Manager GUI wizard.

Agents can request Web Services transparently, as they were located inside the container. ConcreteAgent is an agent that requests a remote Web Service. For this purpose, it invokes the *require()* method, that is inherited from the abstract class Agent, informing as parameter the service name. The method returns a service wrapper object from the Manager agent. In this moment, it adds the data in the *wrapper* and invokes its *run()* method, for a synchronous call, or *submit(),*for an asynchronous call.

Whenever a remote request is executed, the Proxy agent detects what protocol is used (HTTP or SOAP), and redirects the request to the Broker instead of asking for the class Factory instance of a

native entity (in case of a local call). Once the request is redirected to the FES Broker, it is forwarded to Adapter, that converts it to the SOAP format. The processing results are captured and send back to the client agent in a List.

This short description shows how the agents can perform a remote call for Web Services, in the BL application.

5 DISCUSSION

In this paper MIDAS, a platform to support MAS development, has been applied to the communities mediation problem. An application to build consensus among specialists over customers questions were presented. Considering that, even for simple domains, the automatic translation of different jargons is a far to be achieved reality, the solution involving human actors sounds to be an interesting one. The next step for this work is to stress the BL application in an experimental plot of three supermarkets in order to have a robust application to be scaled for any supermarket that desires to join the BeyondTheLabel network.

REFERENCES

Adam, E.; Mandiau, R. Design of a MAS into a Human Organization: Application to an Information Multi-Agent System. *Proceedings of the 5th Agent-Oriented Information Systems*, October 13, 2003 Chicago, IL, USA. Springer Verlag, Lectures in Artificial Intelligence LNAI 3030, 2004. pp. 1-15.

Baru, C.; Gupta, A.; Ludaescher, B.; Marciano, R..; Papakonstantinou, Y.; Velikhov, P. XML-based Information Medidation with MIX. *Proceedings of the Exhibitions Program of ACM SIGMOD*, 1999. pp. 25-34.

Boughaci, D.; Drias, H. An Agent-Based Approach using the ebXML Specifications for E-business. Proceedings of the *5th International Conference on Information Technology: Coding and Computing (ITCC'05)* - Volume II, 2005, pp. 766-767.

Decker, S.; Melnik, S.; Harmelen, F. V.; Fensel D.; Klein M.; Broekstra, J.; Erdmann, M.; Horrocks, I.. The Semantic Web: the Roles of XML and RDF. Proceedings of the IEEE Internet Computing, September 2000. pp.71-80.

Finin, T. KQML as an agent communication language. Third International Conference on Information and Knowledge Management. CIKM-94, ACM Press, nov. 1994. pp. 17-29.

Haendchen Filho, A. A Middleware Framework for Buiding Multi-Agent Systems in the Internet. Ph.D. Thesis, Pontifícia Universidade Católica do Rio de Janeiro PUC-Rio, Rio de Janeiro, RJ, Brasil, 2005.

Jennings, N. R.; Wooldridge M. J. Applying Agent Technology. Int. Journal of Applied Artificial Intelligence, 1996. 9 (4) 351-369.

Kendall, E. et al. A Framework for Agent Systems. In: Implementing Applications Frameworks – Object-Oriented Framewoks at Work. M.Fayadd et al. John Wiley & Soons, 1999.

Klusch, M. (Ed.). Preface of "Intelligent Information Agents." Springer Verlag, Berlin, 1999.

Odell, J. Agent-based Process Management for SOA and WS Applications. *OMG Workshop on MDA, SOA and Web Services* March 21-24, 2005, Orlando, FL, USA.

Piwek, P.; Power, R. CROCODIAL: Crosslingual Computer-mediated Dialogue. In *Proceedings of the 8th International Conference on Enterprise Information Systems*, Paphos, Cyprus, 2006, 75-84.