



AGENT BASED ACCESS TO HETEROGENEOUS DATABASES

Vyacheslav Y. Pysmennyi, Maksym Y. Ternovoy

National Technical University of Ukraine "Kyiv Polytechnic Institute"
Industrialnyy Al., 2, Kyiv, 03056, Ukraine,
e-mail: vpysmennyi@gmail.com, maximter@mail.ru

Abstract: *The paper presents the implementation of system for access to the information in distributed information-telecommunication environment with heterogeneous databases using multi-agent approach. Different ways for multi-agent system construction were considered. It was developed a scheme of interaction between agents in the proposed system. The approach aims to improve the efficiency of data access in heterogeneous information systems without the necessity to change them.*

Keywords: *databases, distributed system, multi-agent system, FIPA, JADE.*

1. INTRODUCTION

The integration of information and computing resources in a single environment is one of the most important areas of modern information technologies. Rapid development of global information and computer networks leads to a paradigm shift in the data accessing, collecting, storing and processing. This trend has been seen to exclusively distributed schema of designing, supporting and storing resources [1]. At the same time there is a desire to virtually merge information resources at the level of access [2, 3].

Information resources often stores in relational databases (DB) of various heterogeneous systems [4]. It causes difficulties and in some cases impossibility to merge such databases into a single logical database using standard tools provided by database management systems (DBMS) [5]. On the other hand, existing approaches to the information integration of such systems require modifying or changing existing software [5, 6], which, at the same time, requires considerable financial costs. Thus, the union of databases, which are parts of the existing automated systems, at the level of access to data is actual issue.

2. MAIN PART

This issue in general cannot be solved by using a monolithic system. The solution of this problem, which is proposed in this paper, is to develop an approach to the design of a multi-agent system (MAS), which is formed by a group of interacting

intelligent agents. The main functionality of the system being designed is to provide an access to information from different databases in the distributed environment. Multi-agent system has the ability to self-repair and resistance to failure due to sufficient supply of components and self-organization. The main characteristics of the components of this system can be identified as [8]: autonomy, limited notion of agents about the structure of the whole system and decentralization of control. As the operation environment of the MAS is used an agent based platform, which is a basic tool for developing MAS and allows user to create, destroy, interpret, execute and transfer agents. Agents who are just registered with a platform can interact with agents that are already registered on the same platform. There is the possibility of interacting between agents that are registered on different platforms.

In process of developing of proposed MAS we should consider the requirements, which are defined by the task:

- Geographically distributed nature of the corporate environment;
- The need to interact with different types of DBMS, including desktop DBMS;
- The need to support scalability and flexibility in case the structure is changed.

In the process of solution of the problem there were offered two architectural approaches to building MAS: multi-platform (Fig. 1) and single-platform (Fig. 2).

To realize the task the second approach has been chosen. It uses a single platform, because it is less

resource-intensive and easier to implement comparing to the first way. Reliability of the system was improved by running a backup platform, to which all agents will switch automatically in the event of unavailability of the main platform.

To build the multi-agent system for information exchange it was proposed to use two types of agents, which named as `MainClientAgent` and `ListenerAgent`. The main objectives of `MainClientAgent` are to provide the interaction with user, to transfer requests to a group of `ListenerAgent` agents and to summarize data received from these agents. The main objectives of `ListenerAgent` are to interact with the database on the nodes of the system and to transfer obtained data to `MainClientAgent` in response to its request. At each node, where the database is located, it should be installed one `ListenerAgent`. `MainClientAgent` should be installed at those nodes where the access to distributed data needs to be provided.

After turning on agents, all `ListenerAgent` agents, which are in the system, register in yellow pages service (DF). This is a guarantee for `MainClientAgent` that it will have the relevant information about the currently active agents in the system at the moment when it has to send requests to all `ListenerAgent` agents.

The scheme of agents' interaction is shown in Figure 3 and described below.

1. When `ListenerAgent` has been started it registers with a yellow pages service.
2. Using the GUI user enters search parameters. `MainClientAgent` uses search parameters to form a request.
3. `MainClientAgent` requests DF for a list of active agents (in this case `ListenerAgent` agents list), that matches specific search criteria.
4. DF creates and sends a response to `MainClientAgent` with a list of agents which are registered in the DF and meet specific criteria.
5. `MainClientAgent` generates a request message and sends it to all agents listed in the list received from the DF.
6. Each `ListenerAgent` receives a request and passes the query to the database it is connected to.
7. DBMS handles the request and sends the result to `ListenerAgent`: a set of data (7a) or an empty set (7b), if no data was found.
8. `ListenerAgent` creates and sends to `MainClientAgent` a message with an attached data set (8a) or with the text "No data" (8b), if the data was not found in the database.
9. `MainClientAgent` handles messages received from each `ListenerAgent` and forms the result data set as a single table and displays it on a screen.

3. IMPLEMENTATION AND APPLICATION

To implement the proposed multi-agent system Java Agent Development Framework (JADE) platform has been chosen. It claims to comply with the FIPA specifications [10, 11] and fully implemented in Java language [12]. JADE contains with a set of containers that are distributed in the network. JADE as a tool for agent development and as a environment for running agents is suitable for implementing the proposed approach because we can use JDBC drivers for database connection, which allows the agent, that was built on JADE, to be independent from the type of database used in the system.

Implementation of the proposed approach was considered on the example of developing a system that handles an access to distributed databases of Main Department of the executive agency of the Kyiv city council (Kyiv City State Administration) for population protection against the Chernobyl Catastrophe after-effects. Existing information environment of the department is a set of legacy unrelated information systems of the department's units that are distributed geographically in the city and use MS Access databases. The information system of the Main Department consolidates information in Oracle DBMS. Now all operations of information consolidation are made manually. To do this it is needed to hand held the following: unload the information from the regional department's databases, transmit it to the main department and download to the central database unit. It causes significant delays in updating the information.

To improve accessing the information from databases in regional departments there was proposed to use the system described above.

4. CONCLUSIONS

Proposed approach allows solving the problems of access to and works with information from heterogeneous databases, and removes the problem of interaction with the data in the outdated information systems without the need for their modification. System based on this approach is built using Java technologies, which allow developer to use such system on any platform on which can run a Java Virtual Machine. To connect agents with DBMS the JDBC drivers are used, that give the possibility to connect an agent to any database for which JDBC drivers are implemented and which supports the SQL.

5. REFERENCES

- [1] Working out of fundamental bases of creation of the distributed Information resources [Digital source]. – Digital text data (205 131 bytes). – Access mode: <http://www.ict.nsc.ru/sitepage.php?PageID=14> Thursday, 31.03.2011 17:49 (in Russian).
- [2] Mbarka B. Control access policies for distributed resources. [Digital source]. – Digital text data (130 853 bytes). – Access mode: http://www.fisoft1.com/sources/policies_report.pdf Thursday, 31.03.2011 17:53.
- [3] Nunes C., Kulesza U., Sant'Anna C., Nunes I., Garcia A., Lucena C., Comparing Stability of Implementation Techniques for Multi-agent System Product Lines. *European Conference on Software Maintenance and Reengineering*, 2009. – p.229-232.
- [4] Connolly T., Begg C., Strachan A. *Database systems: a practical approach to design, implementation and management*. 2-nd edition. Moscow: Williams, 2000. –1120 p. (in Russian)
- [5] Yakovlev Yu. S. About the concept of construction and a choice of the distributed databases of information retrieval systems [Digital source]. – Digital text data (230 675 bytes). – Access mode: <http://dspace.nbu.gov.ua:8080/dspace/bitstream/handle/123456789/727/6-Yakovlev.pdf?sequence=1> Thursday, 31.03.2011 17:55. (in Russian)
- [6] Silin A. V. *Research and working out of models of the distributed databases in corporate information systems*: The dissertation of a Cand.Tech.Sci. : 05.13.11 Moscow, 2002 134 pages. (in Russian)
- [7] Neruda R. Cooperation of Computational Intelligence Agents. *International Symposium on Collaborative Technologies and Systems (CTS'06)*, 2006. – pages. 256-263
- [8] Distributed intellectual systems based on agents. [Digital source]. – Digital text data (659 077 bytes). – Access mode: <http://www.intuit.ru/department/algorithms/distrsa/10/> Thursday, 31.03.2011 18:01. (in Russian)
- [9] FIPA Agent Management Specification. [Digital source]. – Digital text data (1 457 286 bytes). – Access mode: <http://fipa.org/specs/fipa00023/SC00023K.html> Thursday, 31.03.2011 18:11.
- [10] FIPA Specifications. [Digital source]. – Digital text data (85 946 bytes). – Access mode: <http://fipa.org/specifications/index.html> Thursday, 31.03.2011 18:16.
- [11] Bellifemine F., Poggi A., Rimassa G. JADE – a FIPA-compliant agent framework. [Digital source]. – Digital text data (154 953 bytes). – Access mode: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=8FDC78AFCC8CAC3AC365050D78617C59?doi=10.1.1.86.6371&rep=rep1&type=pdf> Thursday, 31.03.2011 18:06.
- [12] Introduction. [Digital source]. – Digital text data (5 950 bytes). – Access mode: <http://jade.tilab.com/doc/html/intro.htm> Thursday, 31.03.2011 18:14.
- [13] Gorelikov M., Ivanov A. Architecture and implementation of JADE platform. [Digital source]. – Digital text data (1 470 431 bytes). – Access mode: <http://agentlab.ru/confluence/pages/viewpage.action?pageId=17367045> Monday, 23.05.2011 18:32.