Developing A Multi-Agent System in JADE for Information Management in Educational Competence Domains

Article *in* Procedia Economics and Finance · December 2015

DOI: 10.1016/S2212-5671(15)00404-9

2 authors:

Adrian Victor Sandita

University of Craiova

7 PUBLICATIONS **4** CITATIONS

SEE PROFILE

Claudiu Ionut Popirlan

University of Craiova

49 PUBLICATIONS **51** CITATIONS

SEE PROFILE

Available online at www.sciencedirect.com

ScienceDirect

Procedia Economics and Finance 23 (2015) 478 – 486

Procedia Economics and Finance

www.elsevier.com/locate/procedia

2nd GLOBAL CONFERENCE on BUSINESS, ECONOMICS, MANAGEMENT and TOURISM, 30-31 October 2014, Prague, Czech Republic

Developing A Multi-Agent System in JADE for Information Management in Educational Competence Domains

Adrian Victor Sandita[a]*, Claudiu Ionut Popirlan[a]

[a]*University of Craiova, 13 A.I. Cuza Street, Craiova 200585, Romania*

Abstract

Significant researches concerning multi-agent systems for information management have been conducted in recent years. Interest in multi-agent systems is rising because there are a lot of benefits that can be associated with their usage. Development of multi-agent system architectures, that facilitate information management, is an important research challenge that faces advancement in the management information systems area in the form of message passing technologies. The challenges associated with information management can be classified into three general categories: acquisition, organization and distribution. A management information system includes a series of processes that an individual (software agent) has to accomplish. Considering these processes, analysis of the information management has led to identification of several sets of such processes. By analyzing the previous research conducted in this area, it is determined that so far there haven't been many studies about information management development approaches based on multi-agent systems platform. The paper presents a multi-agent system that is designed to support the management of information about graduates of one or more educational institutes such as high school, university, masters and doctoral degree. We developed and implemented a multi-agent system for information management using the JADE platform. If the data is located in distributed execution environments, then the idea of using mobile agents allows the improvement of the system's performance. The multi-agents system presented in the paper is a distributed information management system based on the use of mobile agents, who receive queries from the system user and who, visiting the educational institutes databases, where this information can be found, transmit the interrogation results and then, a master agent processes these results based on rule sets implemented in the JADE platform. The mobility of the agent is defined by using an approach based on both the quality attributes specified for the multi-agent architecture and on the execution environments of the multi-agent system. This paper presents a distributed management model based on mobile agents in JADE. The model includes a distributed multi-agent system that is responsible for its management. Solving an information management problem requires a lot of tasks. The paper presents the results of a simulation of the multi-agent system operating on behalf of a user, management tasks and management information for some persons. Particularly, this work is focused on describing the designed multi-agent system

* Adrian Victor Sandita. Tel.: +4-0746-158-775; fax: +4-0251-418-215.
E-mail address: asandita@gmail.com

Selection and/ peer-review under responsibility of Academic World Research and Education Center
doi:10.1016/S2212-5671(15)00404-9

architecture in JADE. This multi-agent system can be used by an employer for an interview in order to find if a potential candidate has competence in your job description (domain of competence).

Selection and/ peer-review under responsibility of Academic World Research and Education Center
Keywords: multi-agents system; knowledge management; distributed knowledge; databases; JADE.

1. Introduction

Information management represents an important part of any organization, and so the main challenge of organizations is to manage this knowledge. The challenges associated with information management can be classified into three general categories: acquisition (which represents extracting the information from the databases, printed resources), organization (solving the problem of information storing) and distribution (solving the problem of extracting the right information at the right place and time).

An approach to support the information management is based on developing a corporate or organizational memory was presented by Popirlan and Stefanescu (2009). An organizational information in multi-agent systems is defined by Wooldridge and Jennings (1995) as an explicit, persistent representation of crucial information in an multi-agent system, in order to facilitate access, sharing, and reuse by members, for their individual or collective tasks. A corporate memory is appropriate to represent the personal information.

The requirements and aims of an organization have both explicitly and implicitly influenced the design and components of complex information systems. As long as these requirements assumed to be relatively stable this is a feasible approach to system design, since changes occur seldom and thus can be accounted by reengineering processes.

Modern forms of organizations are not as stable; they are more flexible, more dynamic, and more adaptable to changes that appear in the system.

This multi-agent system architecture presented by Bellifemine et all. (2007) can be implemented on top of the FIPA (Foundation for Intelligent Physical Agents) architecture, which is a non-profit organization that deals with standardization issues with respect to agents and MAS. Agents act on behalf of their organization, and so they need to represent and follow the objectives of the organization they represent on the one hand and the objectives and aims of the virtual organization on the other hand.

We are developing a multi-agent system to support the management of information that belongs to the students of an educational institution. This is a distributed information management system that uses mobile agents, which are able to receive queries from the user and visit the stored databases in order to generate the required information. The system analyzes the user's query in order to establish which domains have the potential answer and sends the query to each domain until it finds a domain that has the answer. In addition to autonomy, reactivity, pro-activeness and social ability presented by Finin et all (1997), another software agent's property is mobility. Mobility refers to the ability of an agent to dynamically transfer its execution onto different sites implemented by Finin et all. (1997). Another feature of the mobile agent is its cloning ability.

2. Information Management using Multi-Agent Systems Approach

A detailed description of the Multi-Agent Systems (MAS) architecture is presented in Russell and Norvig (1995). The main focus of this article is on the knowledge management details using the recent standards, in particular on the Virtual Organization. The MAS architecture provides an agent framework and brings interoperability within and across agent based applications. The recent implementations of MAS are based on Java and Internet relevant technologies (JADE, JACK, XML, SMTP Active Objects, etc.). The interoperability involves relationships between agents and platforms and relationships between implementations of agent services.

The collaborative agents (MAS) can achieve competitiveness only through continuous optimization of the collaborative virtual organization with maximum synergy as criteria. If a partner doesn't perform according to expectations, for example it doesn't deliver on time or it doesn't bid well enough to compete with its outside

competitors, it will be replaced with a more suitable partner. This decision and appropriate negotiation will be performed by the multi-agent systems.

Security standards are an important issue of the inter-platform accessibility at this level that would enable a fair balance of the autonomy and cooperative forces by enabling enough access to the collaborative cluster's entities to each-other services while keeping secrets safe.

The main requirement of implementing the proposed architecture with software agents is to ensure inter-enterprise/inter-node/inter-platform communication amongst the participants in the collaboration. FIPA presented in FIPA (2009) has already developed strong services that enable each enterprise to become a Node in a Collaborative network of Agentcities. Each enterprise is supposed to implement its software to run on FIPA compliant agent platforms so that the agents on different platforms will be able to communicate with each other and access each other's services to create new value added services for the collaborative cluster.

The implementation mechanisms of the task distribution as well as the cloning procedures influence the decision-making. In Georgescu (2004) is suggested a method that uses fuzzy and allows to keep or to select a partner in the virtual organization. Grouping Policies such as obligation, constraining and authorization are used to keep the best cluster and influence the management structure of the virtual organization. Normative concepts enable interactive contractual design based on control mechanisms such as influence as a negotiation framework that configures the collaborative cluster.

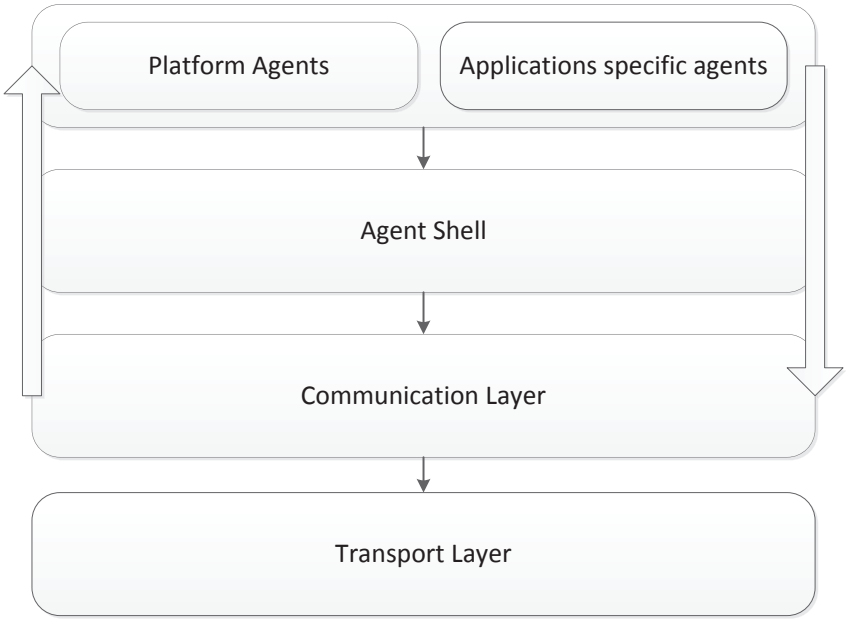

Fig. 1. FIPA-Enabled Multi-Agent System Architecture.

Because the access to web services is standardized, the agent can directly do the search. The use of a mediator agent may still have its advantages since the mediator agent can store information about the success of a cluster of agents in performing a collaborative, complex task on the one hand and in the performance of specific service providers (agents) on the other hand. Since a mediator may gather this experience from a huge number of tasks it may get a much better overview about the performance of agents than an individual agent. Therefore, it may be much better able to define and establish highly successful clusters of agents if a complex task is to be solved.

Considering the characteristics and the architectural aspects, agent technology and web services have the capability to form a perfect symbiosis. Agents can provide their services, can compose complex services from

simpler ones, can deal with the high dynamicity of the Internet and can optimize existing clusters/workflows of highly complex services.

Any enterprise that uses basic Dynamic Systems Environment services can build a virtual organization using FIPA-virtual standards to add the services and policies, standards that include specification of core competencies, process specifications, wireless access to information, financial services among partners and coordination mechanisms. In Fig. 1 the virtual organization architecture (an extension of o classical Dynamic System Environment using FIPA) is presented.

3. JADE Implementation

Multi-agent systems can be made by using any kind of programming language. In particular, object-oriented languages are considered a suitable means because the concept of agents is similar with the concept of objects. In fact, agents share many properties with objects such as: inheritance, message passing, encapsulation, etc.

Agent-oriented programming languages are a new class of programming languages that focus on taking into account the main characteristics of multi-agent systems. Minimally, an agent-oriented programming language must include some structure corresponding to an agent, but many also provide mechanisms for supporting additional attributes of agencies such as beliefs, goals, plans, roles and norms. A list of several agent-oriented languages can be found in (Popirlan, 2010). The software platforms and frameworks represent key instruments in the development of multi-agent systems. Most of them provide a means to deploy multi-agent systems on different types of hardware and operating systems, usually providing a middleware to support their execution and essential operations such as communication and coordination. Some of these platforms and frameworks have the common goal of providing FIPA-compliant functionalities to support interoperation between different multi-agent systems.

JADE presented in Bellifemine et all (2007) is a software platform that provides basic middleware-layer functionalities which are independent of the specific application and which simplify the realization of distributed applications that exploit the software agent abstraction defined in Wooldridge and Jennings (1995). A significant merit of JADE is that it implements this abstraction over a well-known object-oriented language, Java, providing a simple and friendly API.

One of the goals of JADE is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents. During the development of the system with JADE, the following types of classes are created and implemented:

- Agent classes to describe various types of agents.
- User Interface classes for customer interaction.
- Agent activity classes for behaviors.
- Database classes to handle the database of the system.
- Communication classes to manage the negotiation between agents.
- Ontology classes to define concepts, predicates and agent actions for the domain.

The system must be able to ensure the agents communication, including the communication protocols. Agent's communication is based on:

- sending (action);
- receiving (perception) messages.

The degree of coherence and coordination comes from the extent to which the system avoids redundant actions, competition on resources, bottlenecks and the unsafe operating conditions. The goal is to maintain an overall coherence, without always having a global control in place.

The coordination between mobile agents not entering the competition is based on cooperation. For the agents entering in the competition, or those having reciprocal dependence, the coordination is based on negotiation.

Communication between agents is done via messages. An agent can communicate with other agent by message passing. An agent that wants to communicate with another agent first has to create a message object, and then send it to the target agent. A text message has a type of optional argument object. The receiver agent determines the course of action by checking the type of message received and gets parameters as the argument object. For system

implementation one can use a subset of standard indicators, presented by Finin et all (1997) of the Knowledge Query and Manipulation Language (KQML):

tell
:content <expression>
:language <word>
:ontology <word>
:in-reply-to <expression>
:force <word>
:sender <word>
:receiver <word>

A message has two sections:

- message header;
- message content (also known as message body).

The header contains the information regarding:

- sender;
- receiver(s);
- subject;
- date;
- time;
- priority.

There is also a slot that contains the agent identification. Using the agent identification, the information about an agent, such as its name, can be retrieved from a repository or agent directory. Message content has the following attributes:

- an action verb;
- an object;
- preconditions;
- constraints.

The action verb is used to indicate the type of action to be taken by the receiver, such as

- request;
- propose;
- query.

An object is the result or expectation.

Agents must have intelligence to process messages. A body of knowledge supports the intelligence of an agent.

4. Results

In order to illustrate the proposed solution for searching data in competence domains with distributed databases, we will explain the setting and describe a case study.

The application exploits collaborative agents to reach remote distributed databases, in competence domains context, and locally access data of interest, analyzing them and extracting the required information without any need to transfer the data over the network.

The database diagram for case study is shown in Fig 2.

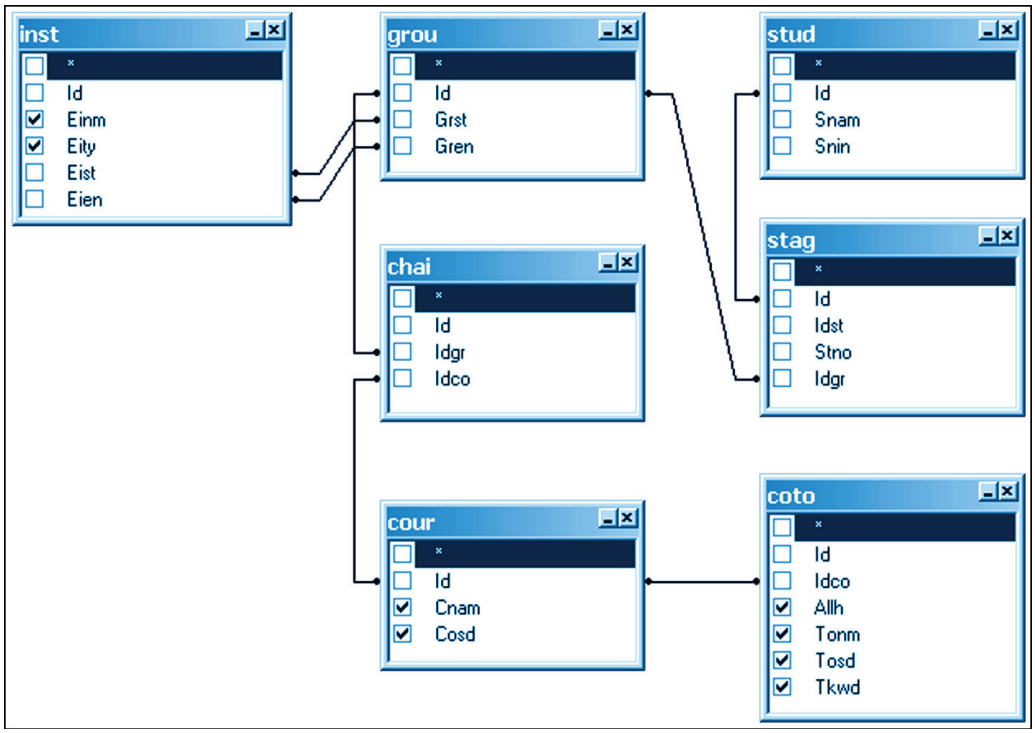

Fig.2. Database Diagram.

Tablo1. The structure of the database tables used in case study is shown below

		Educational Institution			Groups
edin	Id	Educational Institution Id	grou	Id	Group Id
	Einm	Educational Institution Name		Grst	Start date
	Eity	Educational Institution Type		Gren	End date
	Eist	Start date			
	Eien	End date			Students
			stud	Id	Student Id
		Courses		Snam	Student Name
cour	Id	Course ID		Snin	National Identification Number
	Cnam	Course Name			
	Cosd	Course short description			Stages
			stag	Id	Stage Id
		Course Topics		Idst	Student Id
coto	Id	Topic Id		Stno	Stage No
	Idco	Course Id		Idgr	Group Id
	Allh	Allocated hours			
	Tonm	Topic name			Chains
	Tosd	Topic short description	chai	Id	Chain Id
	Tkwd	Keywords		Idgr	Group Id
				Idco	Course Id

Polling was conducted with the following query:

```
SET @Nin = 'nin01',
    @Kwd = '%kwd13%';
SELECT DISTINCT inst.Einm, inst.Eity, cour.Cnam,
                coto.Tonm, coto.Tosd, coto.Allh, coto.Tkwd
FROM  chai, coto, cour, grou, inst, stag, stud
WHERE stud.Snin = @Nin AND coto.Tkwd LIKE  @kwd AND
      grou.Grst >= inst.Eist AND grou.Gren <= inst.Eien AND
      grou.Id = chai.Idgr AND chai.Idco = cour.Id AND
      coto.Idco = cour. Id AND stud.Id = stag.Idst
GROUP BY inst.Einm, inst.Eity, coto.Tkwd
```

For instance, an agent that is sent to a remote database can analyze the local data and return the attributes that contain a specific keyword. To speed up the research, the application can be shaped after a tree of collaborative agents. An agent lives in a particular environment, and therefore JADE provides one, as shown in Fig 3.

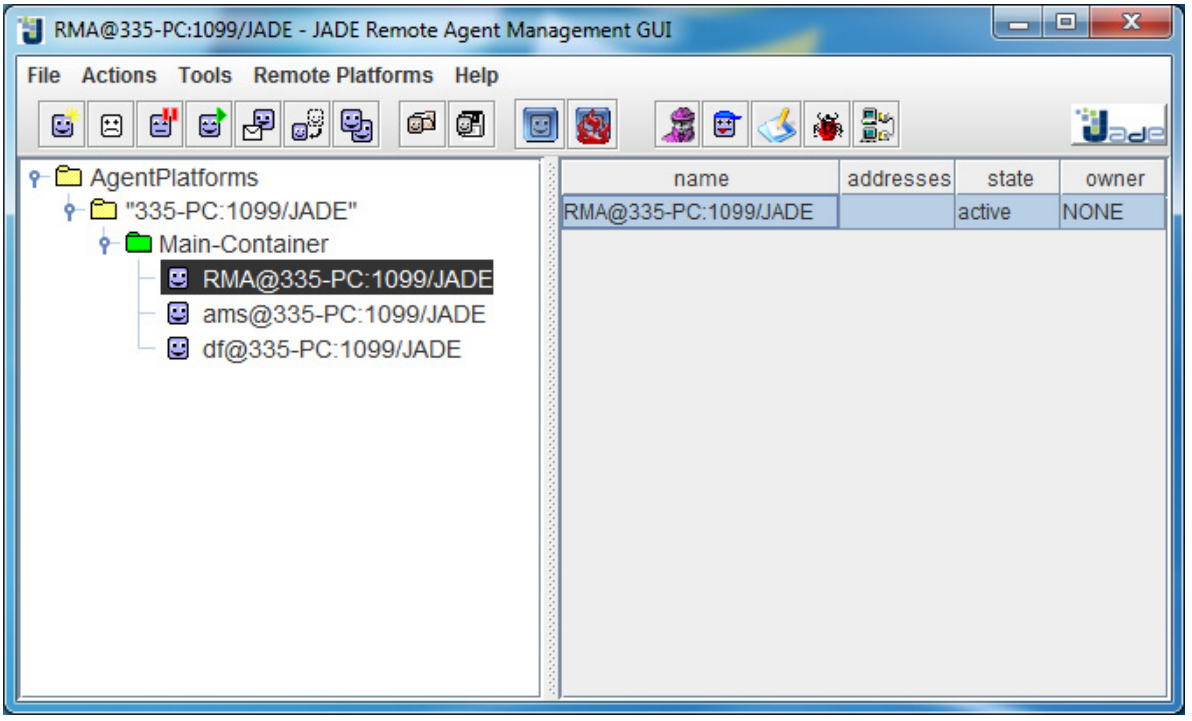

Fig. 3. Intelligent Agent Visualization in JADE Environment.

In Fig. 4 is a sample of the “Introsppector“ Agent created in JADE and in this application.

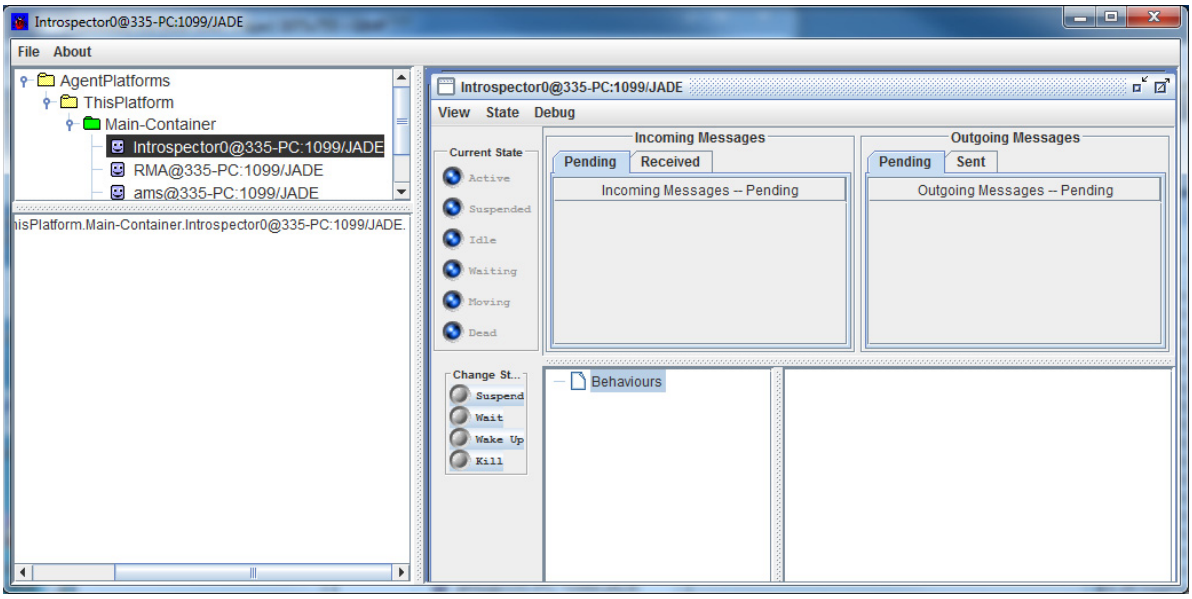

Fig. 4. "Introspector" Agent in JADE Environment.

If an agent on a local database finds links to other possibly interesting data in a different location (distributed context), it clones itself and has the clones follow these links, to recursively continue the search in different data bases, as shown in Fig. 5.

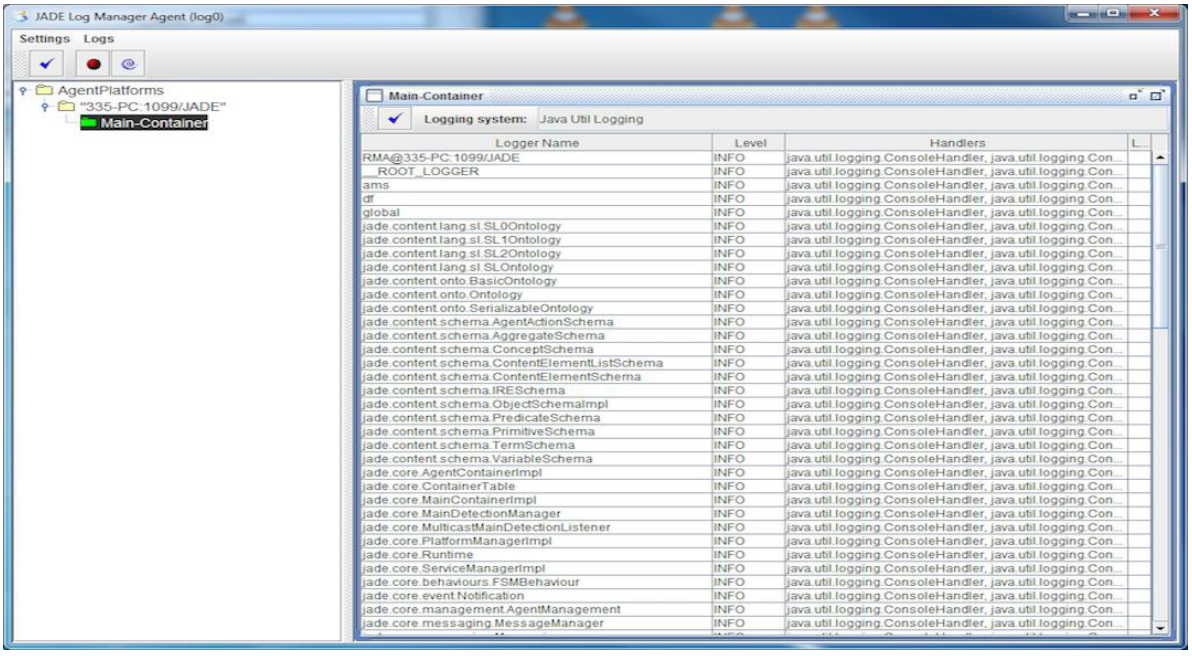

Fig. 5. Agent search work in databases on JADE Environment.

5. Conclusion and future work

In this paper a multi-agent approach for distributed data searching in the context of educational competence domains was presented. We provide a description of java-based agents, developed in JADE environment, and we presented a simple illustration to show how the proposed system might work.

In the future, we will try to implement agents in order to adopt a control-oriented point of view, for example to request services while they require files. With the adoption of a blackboard or a tuple space on each database server, data can be accessed without requiring the presence of peculiar services and in a more natural data-oriented style. We intend to develop a prototype of this multi-agent system, which will successfully address even more relevant and practical problems.

Acknowledgements

Adrian Sandita's work was cofinaced from the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013, project number POSDRU/159/1.5/S/140863, Competitive Researchers in Europe in the Field of Humanities and Socio-Economic Sciences. A Multi-regional Research Network.

References

Bellifemine, F.L., Caire, G., Greenwood, D., (2007). Developing Multi-Agent Systems with JADE, Wiley.

Finin, T. Labrou Y.and J. Mayfield, (1997). Software Agents: KQML as an agent communication language, invited chapter in Jeff Bradshaw (Ed.), MIT Press, Cambridge.

Finin, T., Labrou, Y., Mayfield, J., (1997). Software Agents: KQML as an agent communication language, invited chapter in Jeff Bradshaw (Ed.). MIT Press, Cambridge.

FIPA, 2011. Foundation for Intelligent Physical Agents. website: http://www.fipa.org.

Genesereth, M.R., Ketchpel, S.P., (1994). Software Agents. In Communications of the ACM.

Georgescu, V., (2004),. "A Generalization of Symbolic Data Analysis Allowing the Processing of Fuzzy Granules", Lecture Notes in Computer Science (sub-series: Lecture Notes in Artificial Intelligence), LNAI 3131, Springer-Verlag, Berlin-Heidelberg-New York, pp. 216-226, ISSN 0302-9743, ISBN: 3-540-22555-2.

Georgescu, V., (2006). "Learning Under Bounded Rationality: Conventional versus Computational Intelligence Based Approaches", Lecture Notes in Computer Science, accepted for publication (forthcoming), Springer-Verlag, Berlin-Heidelberg-New York.

JADE, (2011). Java Agent Development Environment. website: http://jade.tilab.com

Karnik, N.M., Tripathi, A.R., (1998). Design Issues in Mobile-Agent Programming Systems, In IEEE Concurrency.

Popirlan, C.I., 2010. Knowledge Processing in Contact Centers using a Multi-Agent Architecture, In WSEAS Transactions on Computers.

Popirlan, C.I., Dupac, M., (2009). An Optimal Path Algorithm for Autonomous Searching Robots, In Annals of University of Craiova, Mathematics and Computer Science Series.

Popirlan, C.I., Stefanescu, L., (2009). Mobile Agents System for Intelligent Data Analysis, In Proceedings of WSEAS Applied Computing Conference.

R.H. Bordini, L. Braubach, M. Dastani, A.E.F. Seghrouchni, J.J. Gomez-Sanz, J. Leite, G. O'Hare, A. Pokahr and A. Ricci, (2006). A Survey of Programming Languages and Platforms for Multi-agent Systems, *Informatica, 30*(1), 33–44.

Russell, S., Norvig, P., (1995). Artificial Intelligence: A Modern Approach, Prentice Hall.

Smith R. and Davis, R. (1981). Frameworks for Cooperation in Distributed Problem Solving, IEEE Transactions on Systems, *Man and Cybernetics*, *Vol*. SMC-11 1981, 61-70.

Stoian G., Popirlan, C.I., (2010). A proposal for an enhanced mobile agent architecture (EMA). In Annals of the University of Craiova, Mathematics and Computer Science Series.

Tandareanu, N., Popirlan, C.I., (2009). A Mobile Agents Approach for Knowledge Bases *Processing, In Proceedings of the Twelfth IASTED International Conference on Intelligent Systems and Control* (ISC 2009).

Wooldridge, M., Jennings, N.R., Kinny, D., (2000). The Gaia Methodology for Agent-Oriented Analysis and Design, *In Journal of Autonomous Agents and Multi-Agent Systems.*

Wooldridge, M.J., Jennings, N.R., (1995). Intelligent Agents: Theory and Practice. In Knowledge Engineering Review.