Multi-Agent System for Distributed Data Retrieval using PQR Approach

M. Murali
Department of CSE
SRM University – 603 203
Chennai, India.

Dr. R. Srinivasan
Dean and PG Head
RNS Institute of Technology
Bangalore, India.

ABSTRACT
The paper describes the process involving distributed data access from a mobile device, employing mobile agents. To answer any query in the distributed environment the search is conducted to answer the query only in the databases, which are known to the systems. The transfer of database to the system, where the query is originated will involve high communication cost, response time and increase the network traffic. In order to reduce the values of these parameters and incidentally the network traffic, mobile agents are used to fetch the result from various sites. . A mobile agent is a software program that migrates from one node to another where the data is located instead of transmitting data across the network. This paper presents a study of deployment of Multi-agent system for retrieval of data and also the management of distributed resources. Experiments conducted reveal the performance of Mobile agents, using Parallel Query Retrieval (PQR) approach.

General Terms: Multi-Agent, Distributed Data, Query Processing, Mobile Computing

Keywords: Mobile Agent, Data Retrieval, JADE

1. INTRODUCTION
In mobile computing environment [1,2,3,4] users can access information independent of their location through wireless connections. However, accessing information should not restrict the mobile users to specific location. From data management point of view, often mobile users are likely to bring only the fraction of data they need to access, because the resources available to them are very limited. So the data sharing among mobile and fixed hosts, need to employ distributed computing technologies. In mobile computing, users are mobile, but the point of attachment to the network changes as they are moving. The emergence of low-cost and portable computing devices such as laptops and personal digital assistants (PDA) have made possible for users to work from anywhere at any time. As the technology is growing, millions of people carry portable computer that uses wireless connection to access global network. As shown in Fig.1, each Mobile Unit (MU) that is connected to a wireless network can be connected, in turn, to a global information network thereby providing unrestricted user mobility.

A Mobile Unit may change its location while computations are being processed. While moving , a mobile device can retain its network connections through the support of Base Stations(BS). The BSs and Fixed hosts (FHs) perform the transaction and data management functions with the help of Database Server (DBS) without affecting any aspect of the generic mobile network. The DB Server can either be at BSs or it can be a part of FHs. Within the mobile computing environment, shared data are stored and controlled by a number of DB Servers. The BS is used as a switch and some specific tasks are assigned. However to include the database functionality the entire architecture of a BS is to be modified. For this reason DBSs, as separate nodes on the wired network, can be reached by any Base Station.

Mobile agent technology [5,6,7,8] is identified as an efficient tool for searching and retrieving information from the remote databases. The main advantage of this tool is that a mobile agent can search information instead of the users. Moreover the mobile agent can migrate to the node where the data is located instead of transmitting data across the network. It performs its task there, and returns to the original node along with the results taken from the nodes that have been visited. When data transmission causes any problem, mobile agent is one of the alternative technology to be used to reduce the execution time. Thus, a mobile agent can utilize the bandwidth of a network much more efficiently than a direct connection when it is accessing remote databases. In this paper, we implement the parallel query retrieval approach in Multi-agent system for off-line queries.

2. RELATED WORK
In [9] an experimental mobile-agent system searching technical reports distributed across multiple machines is described. The application was implemented on the D'Agents system [10]. It utilizes the statistical information retrieval system, Smart, that uses the vector-space model to measure the textual similarity between documents and is wrapped inside a stationary agent on each machine. Kawamura et al. [11] have analyzed three basic agent paradigms applied to the process of accessing data from distributed databases such as direct access, stationery agent access, and mobile agent access. Also, Papastavrou et al. [12] and Ismail et al. [13] have analyzed the comparisons between a Java applet-based approach and a mobile agent technology in accessing distributed databases from the WWW. Menczer [14] designed and implemented Myspiders, a multi-agent system for information discovery in the Internet. The usage of mobile agents for information filtering has been analyzed by Theilmann et al. [15]. Nguyen with his colleagues [16] has developed an agent system that helps information retrieval from the Internet.

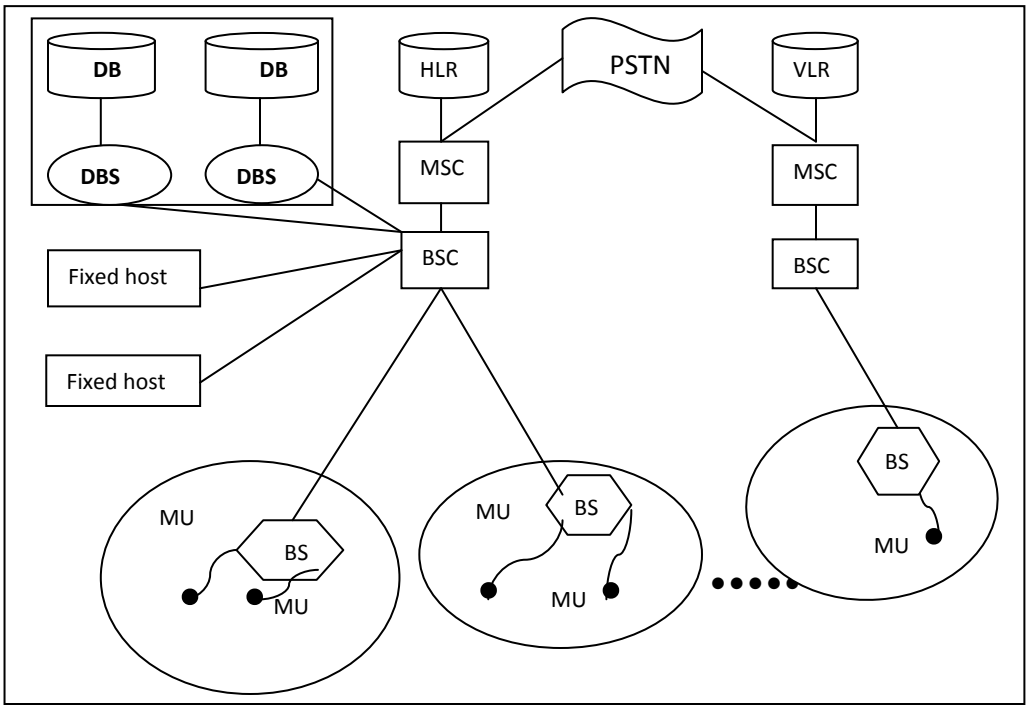


Fig-1 : Mobile Computing Reference Architecture

3. MOTIVATION

Centralized solutions are generally more efficient. Anything that can be computed in a distributed system can be moved to a single computer and optimized to be at least as efficient. However , distributed computations are sometimes easier to understand and easier to develop especially when the problem being solved is itself distributed. Multi-agent systems are the best way to design distributed computing systems.

4. TOOL USED IN IMPLEMENTATION

JADE (Java Agent Development Framework) is a middle-ware [17,18,19] which simplifies the implementation of multi-agent systems by providing a set of graphical tools that support the debugging and deployment phases. In JADE, the agent platform can be distributed across machines and the configuration can be controlled via a remote Graphical User Interface (GUI). JADE has several interesting features that at least make the process of implementation easier. They are distributed agent platform, GUI , debugging tools, intra-platform agent mobility and agent sniffer.

JADE classes easily support the development of multi-agent systems. It warrants syntactical compliance and, where possible, semantic compliance with Foundation for Intelligent Physical Agents (FIPA) [17] specifications. The Figure 2 is showing the architecture of FIPA. The Agent Management System (AMS) is the agent who exerts supervisory control over access to and use of the agent platform. Only one AMS will exist in a single platform. The AMS provides white-page and life-cycle service, maintaining a directory of Agent Identifiers (AID) and agent state. Each agent must register with an AMS in order to get a valid AID. The Directory Facilitator (DF) is the agent which provides the default yellow page service in the platform. The Message Transport System, also called Agent Communication Channel (ACC), is the software component controlling all the exchange of messages within the platform, including messages to/from remote platforms.

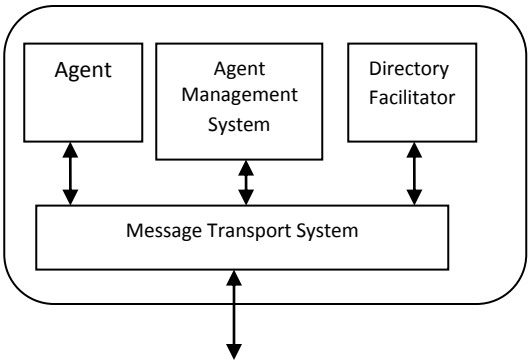


Fig.2 The reference architecture of a FIPA agent platform

JADE fully complies with the reference architecture and when a JADE platform is launched, the AMS and DF are immediately created. Furthermore the Messaging Service is always activated to allow message-based communication. The agent platform can be split to be on several hosts. Each JVM is a basic container of agents that provides a complete run time environment for agent execution and allows several

agents to concurrently execute on the same host. The main-container is the one where the AMS and DF live. The other containers, instead, connect to the main container and provide a complete run-time environment for the execution of any set of JADE agents.

5. OVERVIEW OF MOBILE AGENT

In many cases, data is distributed in several databases and a search engine cannot access it, because it may not have direct access to the content of one or more databases. This is true because the databases are private and access is limited. The field of distributed information retrieval focuses on searching a set of distributed databases. The results from searching each of these databases must then be combined into a single list.

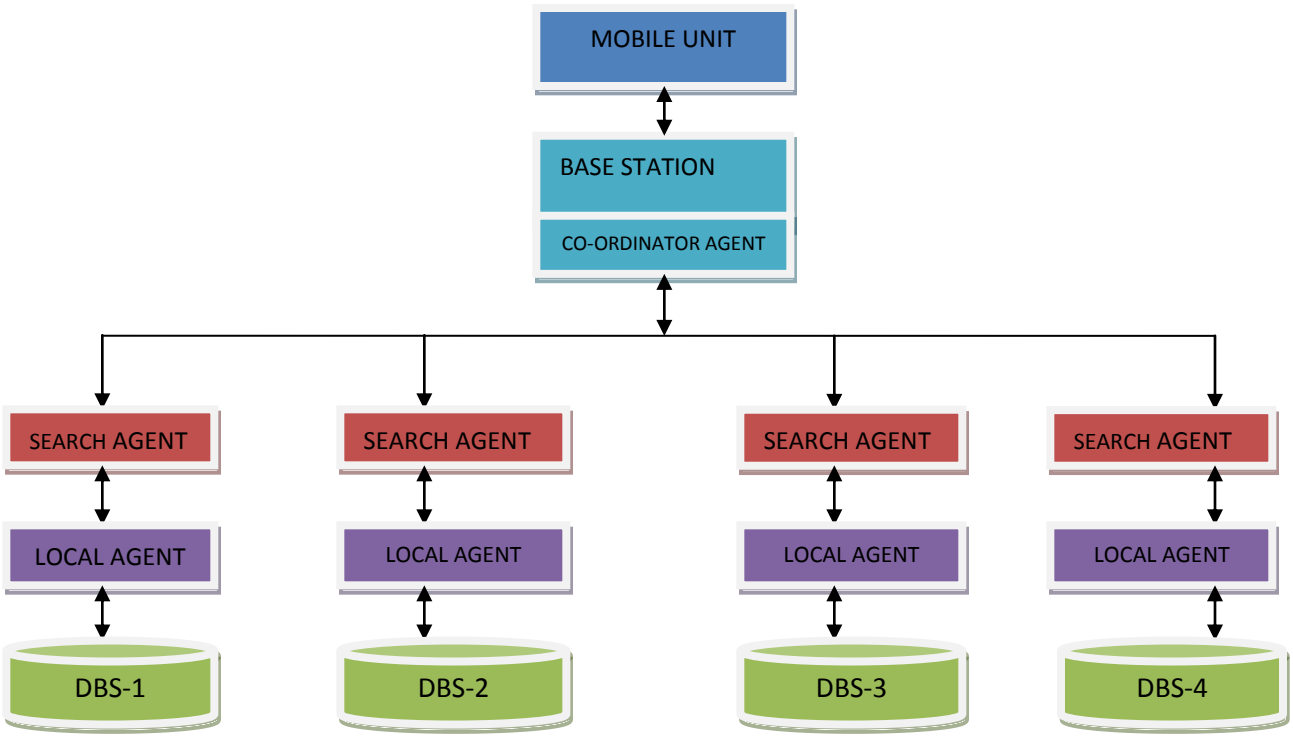
A mobile agent is an autonomous program that can move from machine to machine in the network under its own control [5,6,7]. It can suspend its execution at any point, transport itself to a new machine, and resume execution from the point it stopped execution. An agent carries both the code and the application state. Actually, the mobile agent paradigm is an extension of the client/server architecture with code mobility.

6. MULTI-AGENT RETRIEVAL SYSTEM

A user can enter query from the mobile device. The query is migrated to BS, since no more user interaction is needed. The BS / FH comprises of the co-ordinator agent and the search agent. The Co-ordinator agent is a stationery agent. Upon receiving request from the mobile device through Base Station, the co-ordinator agent [20,21,22] will assign the tasks to the search agents. When receiving a query request from the co-ordinator agent, the search agent will start to move to a collection of database servers. Then it will communicate with the local agent at the respective database server. Since the Local Agent having knowledge about the structured data in the database, the query is applied to database server. Then search agent will ask the local agent about the query that is requested by the co-ordinator agent. The query results will be given to the search agent by the local agent. The same process is continued in all the database servers. Then the search agent will return results to the co-ordinator agent. After receiving the search results from the search agent, the search results will be stored into the server database. The simple Join operation is applied before returning them to the mobile device.

7. THE PARALLEL QUERY RETRIEVAL (PQR) APPROACH

The PQR approach [23] is shown in Figure. 3. The co-ordinator agent will dispatch Several search agents to all database servers. The search agents are sent to the database servers one after the other in a sequence. After collecting the results from the database servers the search agents will return the results to the co-ordinator agent. Since multiple agents are dispatched at a time, it takes less amount of time to collect the results from different servers. The main task of the Co-ordinator agent is generation of search agents and notifying the task. Since several agents are dispatched in parallel , time taken to retrieve the data is minimized.



DBS – Database Server

Fig-3: Multi-Agent system using PQR approach.

8. AGENT COMMUNICATION

Traditional inter-process communication is followed in Multi-Agent applications[24,25,26,27]. These mechanisms are too low level for supporting the communication requirements of interacting mobile agents. The reason is that the traditional inter-process communication mechanisms [28,29] do not consider mobility of interacting entities. Since mobility is a distinguishing feature of mobile agents, the locations where agents are executing at the time of communication, and the location where actual communication takes place plays an important role.

In the inter-agent communication, mobile agents communicate with one another at a specific location in the system. Either the sender agent, or the receiver agent, or both of the agents need to visit the location to communicate. The inter-agent communication technique as shown Fig. 4 involves a Co-ordinator agent (CA) running on Base Station , a searcher agent (SA) running on database server, for data to be communicated. But in this work of inter-agent communication, agents communicate [30,31] with one another independent of the location in the network. A coordinator agent is responsible for a large work, subcontracts various tasks of the work to one or more searcher agents by negotiation. To implement this efficiently, the coordinator agent and one or more searcher agent(s) communicate with one another.

The protocol used for the communication between the Co-ordinator agent and the search agents, is the FIPA-Request protocol. It is well suited for the process, since the co-ordinator agent has to notify message to all search agents and the required parameters. The co-ordinator agent initiates a FIPA-Request protocol with each search agent and it sends a request message to the search agent asking a service. The searcher agent receives, executes and returns message to the co-ordinator agent along with the result of the action.

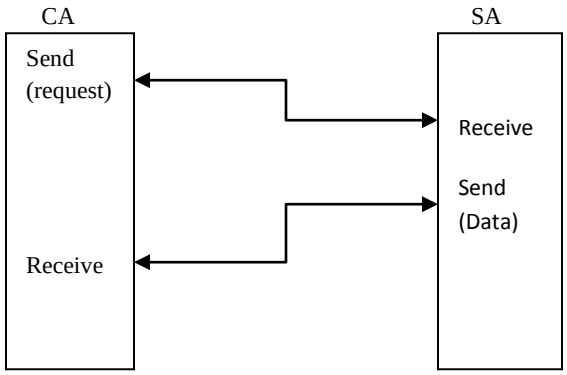


Fig. 4 Agent Communication

9. SYSTEM OPERATION

9.1 Starting the agent execution

The JADE framework controls the creation of a new agent. For this purpose, the agent constructor is executed. The agent is given an identifier and it is registered with the AMS, it is put in the ACTIVE state, and finally the setup() method is executed.

9.2 Stopping agent execution

Any behaviour can call the Agent.doDelete() method to stop agent execution.

9.3 Launching an agent

A list of agents can be specified on the command line, by specifying arguments, embedded within parenthesis, and can be passed to each agent.

9.4 The agent tasks and behaviours

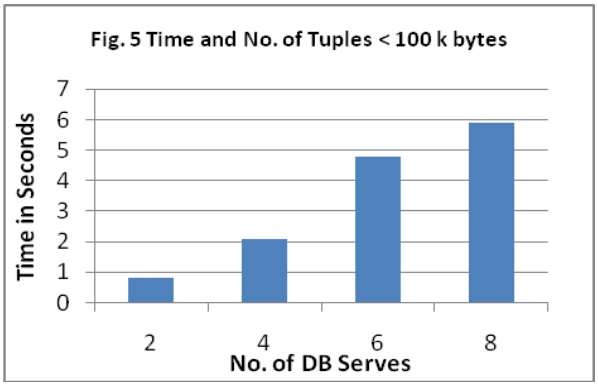
An agent must be able to carry out several concurrent tasks, every JADE agent is composed of a single execution thread and all its tasks are modeled and can be implemented as Behaviour objects. The developer who wants to implement an agent-specific task should define one or more Behaviour subclasses. The addBehaviour(Behaviour) and removeBehaviour(Behaviour), allow to manage the ready-tasks- queue of a specific agent.

9.5 Agent mobility

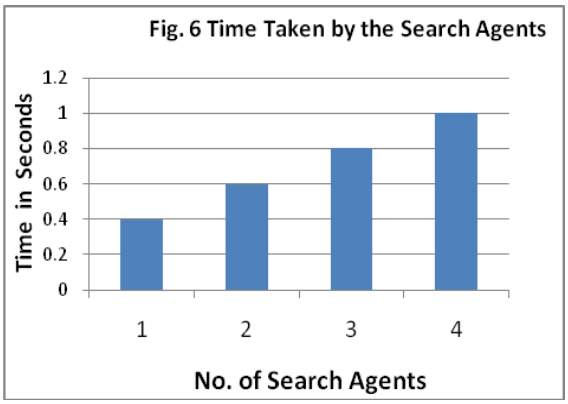
The two public methods doMove() and doClone() of the Agent class allow a JADE agent to migrate elsewhere or to spawn a remote copy of itself under a different name. Method doMove() takes a jade.core.Location as its single parameter, which represents the intended destination for the migrating agent. Method doClone() also takes a jade.core.Location as parameter, but adds a String containing the name of the new agent that will be created as a copy of the current one.

10. EXPERIMENT RESULTS

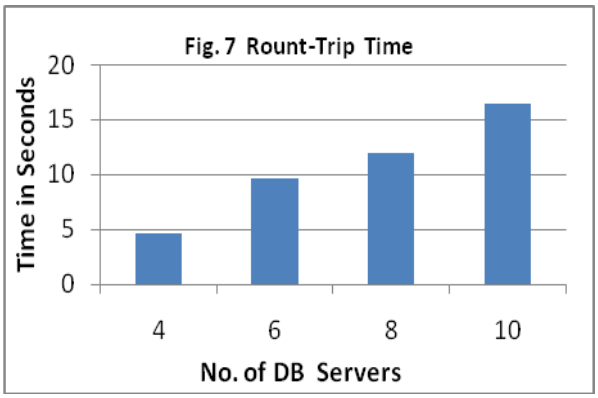
Experiment was conducted in JADE platform to retrieve the data from distributed environment. Mobile Agents can move to the data source, where it locally retrieves the data. Performance is good when the mobile agent's code to be transferred. In this experiment the minimum no. of search agents used, to test the performance in the Local Area Network. We did not evaluate the quality of the results obtained. Further we assumed that only one query is received from the mobile unit at a time. The experiments were conducted in order to evaluate the performance of query retrieval using the mobile agent technology and the results are as depicted in Figures 4,5 and 6.



The size of the tuples retrieved by the search agents is less than 100 Kb. The no. of database servers considered in this experiment is minimum. The time is calculated in the program itself as time taken to send the query to Co-ordinator agent, further to search agents and return time to the user .



The turnaround time of individual search agents is calculated. The time taken to tranfer the data , depends on the no. of search agents. If we send more number of search agents, the time taken to retrieve the data is slightly increased. The time is calculated from the query received till the result is returned to co-ordinator agent.



The toal time is calculated to process the request. The response time is varying, depends on the no. of database servers. The time is calculated from the query is received till the result is returned. In this experiment no waiting time was assumed at any point. Further the experiment was conducted during non-peak hours to avoid the traffic and data loss in the network.

11. CONCLUSION

This paper presents a study of Multi-agent system usage for the management of distributed resources. Multi-agent technology is an alternative approach to the client-server traditional systems. Mobile agent based approach offers some advantages, such as scalability of the system, load balancing and low traffic in the network. The mobile agents require a proper environment for implementation and execution. For the implementation of the mobile agents, the JADE (Java Agent Development Framework) is chosen. Further work will be related to the security of the mobile agent.

12. REFERENCES

[1] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, Broadcast disks: data management for asymmetric communication environments, in: Proceedings of the ACM SIGMOD Conference, CA, 1995.

[2] R. Alonso, H.F. Korth, Database system issues in nomadic computing, in: Proceedings of ACM SIGMOD Conference on Management of Data, 1993, pp. 388–392.

[3] R. Alonso, S. Ganguly, Query optimisation for energy efficiency in mobile environments, in: Proceedings of the 1993 International Workshop on Foundations of Models and Languages for Data and Objects, Aigen, Austria, 1993.

[4] Y. Fu, S. Madria, Multi-layered databases for intelligent query answering in mobile environments, in: International Workshop on Reliable and Secure Applications in Mobile Environments, New Orleans (also invited paper in NSF workshop), October, 2001.

[5] S. Fricke, K. Bsufka, J. Keiser, T. Schmidt, R. Sesseler, and S. Albayrak, "Agent-Based Telematic Services and Telecom Applications," Comm. ACM, vol. 44, no. 4, pp. 43-48, Apr. 2001.

[6] M.L. Griss and G. Pour, "Accelerating Development with Agent Components," Computer, vol. 34, no. 5, pp. 37-43, May 2001.

[7] D Gavalas, G. Tsekouras, C. Anagnostopoulos, A mobile agent platform for distributed network and systems management, Journal of Systems and Software 82 (2) (2009) 355–371.

[8] Tainchi Lu_, Chinghao Hsu , Mobile agents for information retrieval in hybrid simulation environment , Journal of Network and Computer Applications 30 (2007) 244–264

[9]. Brewington, B., et al.: Mobile agents in distributed information retrieval. In: Klusch, M.(ed.) Intelligent Information Agents, pp. 355–395. Springer, Heidelberg (1999)

[10]. Gray, R.S., Kotz, D., Cybenko, G., Rus, D.: 'D'Agents: Security in a multiple-language, mobile-agent system. In: Vigna, G. (ed.) Mobile Agents and Security. LNCS, vol. 1419, pp. 154–187. Springer, Heidelberg (1998)

[11] T. Kawamura, S. Joseph, A. Ohsuga, S. Honiden, Designing multi-agent systems based on pairwise agent interactions, IEICE Transactions on Information Systems E 84-D (8) (2001) 968–980.

[12] S. Papastavrou, G. Samaras, E. Pitoura, Mobile agents for World Wide Web distributed database access, IEEE Transaction on Knowledge Engineering 12 (5) (2000) 802–820.

[13] L. Ismail, D. Hagimont, A performance evaluation of the mobile agent paradigm, in: Fourteenth Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), Denver, November 1999, pp. 306–313.

[14] Menczer, F.: Complementing Search Engines with Online Web Mining Agents. Decision Support Systems 35, 195–212 (2003)

[15] W. Theilmann, K. Rothermel, Optimizing the dissemination of mobile agents for distributed information filtering, IEEE Concurrency 8 (2) (2000) 53–61.

[16] Nguyen, N.T., Ganzha, M., Paprzycki, M.: A Consensus-based Multi-agent Approach for Information Retrieval in Internet. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A.,Dongarra, J.J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 208–215. Springer, Heidelberg (2006)

[17] FIPA: Agent Management. Available, http://www.fipa.org

[18] JADE, 2006, http://jade.tilab.com/.

[19] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE: a software framework for developing multi-agent applications. Lessons learned, Information and Software Technology 50 (1–2) (2008) 10–21.

[20] J. Rosenschein and G. Zlotkin, "Designing Conventions for Automated Negotiation," Readings in Agents, San Francisco: Morgan Kaufmann, M. Huhns and M. Singh, eds., 1997.

[21] T. Sandholm and V. Lesser, "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework," Proc. First Int'l Conf. Multi-Agent Systems, 1995.

[22] Jin-Wook Baek *, Heon Y. Yeom, A timed mobile agent planning approach for distributed information retrieval in dynamic network environments, Information Sciences 176 (2006) 3347–3378

[23] Ali Selamat *, Md. Hafiz Selamat, Analysis on the performance of mobile agents for query retrieval, Information Sciences 172 (2005) 281–307

[24] N.R. Jennings and M.J. Wooldridge, Agent Technology: Foundations, Applications, and Markets, Berlin, Germany: Springer-Verlag, 1998.

[25] M.P. Papazoglou, "Agent-Oriented Technology in Support of EBusiness," Comm. ACM, vol. 44, no. 4, pp. 71-78, Apr. 2001.

[26] B. Hayes-Roth, L. Brownston, and R. van Gent, "Multiagent Collaboration in Directed Improvisation," Readings in Agents, San Francisco: Morgan Kaufmann, M. Huhns and M. Singh, eds., 1997.

[27] A. Murphy, G. Pietro Picco, and G. Roman, LIME: A Middleware for Logical and Physical Mobility, Proc. 21th Int'l Conf. Distributed Computing Systems, Apr. 2001.

[28] S. Mishra and Y. Huang, "Fault Tolerance in Agent-Based Computing Systems, Proc. 13th ISCA Int'l Conf. Parallel and Distributed Computing Systems, Aug. 2000.

[29] S. Mishra, Y. Huang, and H. Kuntur, DaAgent: A Dependable Mobile Agent System, Proc. 29th IEEE Int'l Symp. Fault-Tolerant Computing (Fast Abstract), June 1999.

[30] S. Mishra, Y. Huang, and H. Kuntur, "Programming Environment of DaAgent," Proc. 12th IASTED Int'l Conf. Parallel and Distributed Computing Systems, Nov. 2000.

[31] S. Mishra and P. Xie, "Interagent Communication and Synchronization in DaAgent, Proc. 21st Int'l Conf. Distributed Computing Systems (poster paper), Apr. 2001.