MULEP-A Multi Level E-Procurement System with Distributed Agents

Ozgur Koray SAHINGOZ, Emre OZTAS

Abstract— Supply chain management system satisfies the customer demands through the most efficient use of resources, including distribution capacity, inventory and labor. E-procurement is expected to be integrated with the supply chain management systems, which are analyzed or processed by computers, in a large-scale and dynamic environment. In this type of systems there can be any number of buyers and suppliers at any time. Each supplier may provide necessary products from other suppliers, piece by piece or as a whole. This type of mechanism forms a multi level hierarchy between suppliers. This paper presents MULEP-a MUlti Level E-Procurement system which deploys the procurement process in this hierarchical supplier structure. System is managed by a set of distributed mobile agents which are developed over Java Agent DEvelopment Framework-JADE for platform independence.

Index Terms— **Multiagent System, E-procurement, Artificial Intelligence, Mobile Agent.**

I. INTRODUCTION

The number of businesses and individuals through the world who are discovering and exploring the Internet is growing dramatically. Through the past decades we have seen an increasing rate of globalization of the economy and, thereby, also of supply chains. Supply chain management (SCM) is the oversight of goods and services as they move in a process from supplier to consumer.

SCM is necessary cornerstone for any organization that wants to compete in the current worldwide economy. SCM contains the planning and management of all activities involved in purchasing of products, procurement, conversion, and logistics management activities. E-procurement is the business-to-business acquisition of goods and services over the Internet for managing corporate spending and expenditures. Recent trends in e-Procurement applications toward effectively reduce supply chain costs including spatial, temporal, and monetary resources in a computerized way.

E-procurement solutions automate and optimize many purchasing workflows, especially in interactions with suppliers, who are currently assuming responsibility for an increasing number of tasks. In an E-Procurement process it is important to get goods in the right quantity and quality, at the right time, in the right place and from the right source for the direct benefit or use of corporations, or individuals When you take a closer look at purchasing, you realize that often the greatest discrepancy between cost of procurement and actual item value can be found in low-value goods. This means that huge savings are waiting to be made in the purchasing department.

A. The problems

In many supply chain systems, a buyer (or the system) has a fixed number of suppliers which produce or procure a large variety of products and components. These suppliers are initialized into the system at start up and they do not always have products when it is needed. They could procure or compute the price of the interested product by getting the necessary information from sub-level suppliers. This forms a hierarchical structure between suppliers and almost all e-procurement system does not get into consider this hierarchical approach.

The wide geographic spread of the company’s operations, as well as the number of items it sells and different stocking policies for items based on demand, make internal operations exceedingly complex. Most of the supply chain system has a static supplier base. When a new supplier is to be added, it has to be registered manually by supplying its address and the necessary parameters. In a dynamically changing supply chain environment, a system should have the ability to adapt itself to this dynamic world.

B. Contributions

A fundamental characteristic of sourcing decisions is their uncertainty. A buyer never has complete information about all aspects of supplier performance and their future development. Contracts can never cover all possibilities of future developments. As a result, procurement system should try to reduce uncertainty by information seeking from suppliers. Suppliers may provide necessary products from their sub level suppliers as piece by piece or as a whole. This type of mechanism forms a multi level hierarchy, which could be located in a large scale environment, between suppliers.

Traditional approaches to software design are not sufficient when designing large scale distributed e-procurement system that exploits code mobility and dynamic reconfiguration Software agents are encapsulated computer systems situated

Manuscript received ICCSE in April 18, 2009.

Ozgur Koray SAHINGOZ, Turkish Air Force Academy Computer Engineering Department, Yesilyurt, Istanbul, TURKEY, (e-mail: sahingoz@hho.edu.tr).

Emre OZTAS, Turkish Air Force Academy Computer Engineering Department, Yesilyurt, Istanbul, TURKEY, (e-mail: emreoztas@gmail.com).

in some environments such as the Internet and are capable of flexible, autonomous actions in that environment to meet their design objectives. Therefore we designed MULEP-a MUlti Level E-Procurement system in which procurement process is fulfilled via distributed mobile agents developed over Java Agent DEvelopment Framework-JADE .

C. Outline of the paper

The rest of the paper is organized as follows. Section 2 makes a brief review of the literature. Section 3 introduces Agent concepts and Java Agent DEvelopment Framework-JADE. Our system framework is presented in Section 4 and its evaluation presented in Section 5. Finally, Section 6 concludes the paper.

II. Literature Review

A number of researches have focused on autonomous agent technology for B2B e-commerce applications [1]. For example, in [2] it was proposed as an agent-based intermediary architecture capable of supporting multiple suppliers and buyers within a multiple product domain. He et al. [3] surveyed various electronic commerce researches and pointed out that more advanced services like collaboration with other brokers should emerge in order to provide more support to buyers and suppliers involved in transactions. A number of organizations have recently adopted e-Procurement systems to purchase indirect materials for processes like operations, sales, maintenance, and administration [2].

MASCOT(‘Multi-Agent Supply Chain cOordination Tool’) is an architecture that aims at providing a framework for dynamic selection of supply chain partners and for coordinated development and manipulation of planning and scheduling solutions at multiple levels of abstraction across the supply chain[4]. There is no mobility of agents in the system, it uses a blackboard mechanism for communicating agents.

Wang and Fang designed an intelligent agent-based supply chain simulation system [5] in which a supply chain entity is represented by an agent. There are six layers in this system: raw material, providers, component manufacturers, product assemblers, product holders, retailers, and final buyers. Each agent has both suppliers and buyers except the raw material agents and buyer agents.

Kovalchuk’s aim in this study is to develop an integrated application for SCM [6], due to its complexity; it is difficult to address all the issues which can arise in the domain of SCM. To narrow the research scope, the project is mainly focused on the demand part of the supply chain.

A Two-Leveled Mobile Agent System for Electronic Commerce System is developed in [7]. System uses mobile agents as mediator but there is a two level mobility between buyers and suppliers. System uses publish-subscribe mechanism for registration process and registrations are done in a centralized structure to the main broker. Scalability of the system is not sufficient and there are some deficiencies for resembling the real world supply chain systems.

III. Multi Agent Systems and JADE

Software agents are encapsulated active codes situated in some distributed environments like Internet and are capable of flexible, autonomous actions in that environment to meet their design objectives. Multi agent systems are a new and attractive design paradigm and aims to introduce the required intelligence into distributed computation. Traditionally designed distributed systems have some problems and might have to handle work-load, the trend to open large numbers of buyers and suppliers, direct access to services/goods and user mobility [8]. Wooldridge and Jennings in [9] described intelligent mobile agents as having four main characteristics, Autonomy, Social Ability, Reactivity and Proactiveness, to overcome most of distributed system problems.

Java Agent DEvelopment Framework-JADE fully implemented in the Java language, which simplifies the implementation of Multi-Agent Systemsin a platform independent manner. The JADE platform can be seen as a middleware providing a set of useful tools that support the debugging and deployment phase [10, 11].

We select JADE as out multiagent system framework because;

- JADE simplifies the development of distributed and mobile applications which are composed of autonomous entities that need to communicate and collaborate in order to achieve the working of the entire system.
- JADE agents have their autonomy by controlling their own thread of execution. Consequently, they can be easily programmed to initiate the execution of actions without human intervention just on the basis of a goal and state changes.
- JADE is developed as an open-source project that involves the contributions and collaborations of the users.
- JADE provides a homogeneous set of APIs that are independent from the underlying operating system, network and Java version.
- JADE uses FIPA compliant communication language and has been designed to simplify the communication and message transport by making it transparent to the developers to send a message from an agent to another agent, and so allowing her/him to concentrate on the logic of the application.

JADE includes both the libraries of Java classes required to develop application agents and the run-time environment that provides the basic services and that must be active on the device before agents can be executed. Each instance of the JADE run-time is called container. The set of all containers is called platform and provides a homogeneous layer that hides to agents the complexity and the diversity of the underlying tiers (hardware, operating system, type of network, JVM). Figure 1 draws the architecture of a JADE agent system deployed on a set of heterogeneous computing nodes [12].

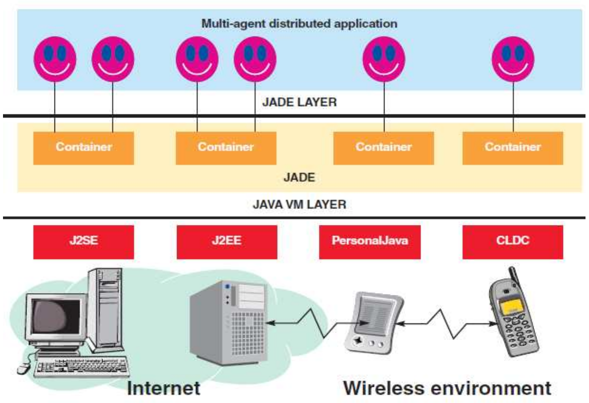


Figure 1. JADE Architecture

IV. System Development

In many electronic commerce and supply chain systems, a buyer (or the system) has a limited number of suppliers, which are registered into the system at start up. There might have many suppliers, spread all over the world, who supply a large variety of parts and components. These suppliers does not always have the requested products, But they could get the price from the sub level suppliers with a hierarchical manner. When a new supplier is to be added, it has to be entered manually by supplying its address and the necessary parameters. In a dynamically changing supply chain environment, a system should have the ability to adapt itself to this dynamic perfect market world.

In our system we developed a new e-procurement model for multi level supply chain management system which has a large number of buyers and suppliers who spread all over the world and interacting via the Internet. Our system hierarchical framework is shown in Figure 2.

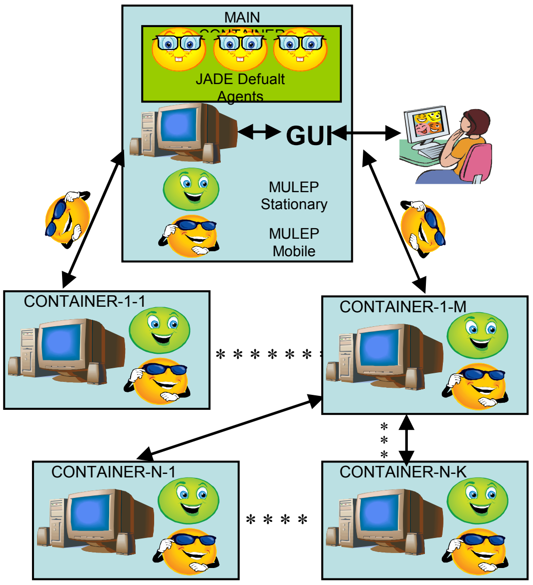


Figure 2. Hierarchical Framework of MULEP

The system is composed by a Main Controller running on a central host and several remote Containers running on remote host. Both Main Controller and remote containers have its own execution flow and stationary agents. In the system mobile agents are used as mediator. The term mobility is used to indicate a change of location performed by the entities of a system. Starting from simple data, the mobility has had an evolution that has led to the movement of execution control, code and execution environment.

The execution flow of the procurement process is as the follows;

a. There are some buyers and suppliers, which have subscribed to system. Buyers subscribe to main container system and suppliers are subscribed to other subscribers or the main container. The number of buyers and suppliers in the system can increase or decrease at any time.

b. When a buyer wants to buy a product, he has to make a request from the main container via its Graphical User Interface.

c. MULEP Stationary Agent generates a mobile agent who will search the best price and return the result to main container. This mobile agent generates clones of itself and sends these clones to each of the registered suppliers.

d. The main role of the supplier is to compose of an execution platform for incoming mobile agents. When a supplier gets a mobile agent it activates this mobile agent and this agent communicates with the stationary supplier agent for deciding what to do. At this level a mobile agent can do three things;

- It gets a price from the supplier agent and returns result to one up level.
- Supplier agent returns it does not have the necessary products right know, but they can be obtained from one sub-level supplier, if mobile agent interested in with these suppliers, SA gives the necessary address information for getting price piece by piece or as a whole.
- SA does not have the products and cannot obtain from any where, so there is no need to do anything.

e. After all results are collected from the sub level suppliers (or the specified timeout period has expired), Mobile agent selects the best dealing one and sends this message to up level mobile agent.

f. At last the agent at the main container gets the all necessary prices and select best dealing one. After that it sends the necessary approval message to suppliers directly or over the hierarchical mobile agents.

Because our system is designed for a large scale platform, there can be always an error in some computers or at connections. For preventing from locking we define a timeout period in which all answers should be got. Every mobile agent waits answers from the sub level mobile agents in this period of time which is set when the mobile agent created. If in this time no answer is get, then mobile agent makes its selection between incoming answers and return result to the up-level agent.

A. Life Cycle of an Agent

A Mobile Agent has a three-stage execution life cycle on different nodes and suppliers on our E-Procurement System. Flow of transition between stages, depicted in Figure 3, is as the following;

(1) being created on the main container,
(2) migrate/clone itself to related suppliers
(3) disposal upon completion of mission.

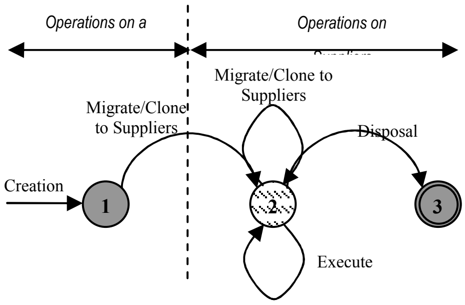


Figure 3. Life Cycle of a Mobile Agent

Agents communicate by asynchronous message exchange, a communication model almost universally accepted for distributed and loosely coupled interactions, i.e. between heterogeneous entities. In order to communicate, an agent just sends a message to a destination. Agents are identified by a name, therefore the send operation does not need the destination object reference and, as a direct consequence, there is no temporal dependency between communicating agents. The sender and the receiver could be available at different times.

B. Agent Communication

Agents in the supplier side should communicate between them for cooperation. The mobile agent does not have authorization at the supplier side. It can collect necessary information from the supplier agent by sending messages. For standardized communication between heterogeneous agents we use the FIPA Agent Communication Language (ACL) [12] messages as shown in Figure 4. The FIPA ACL specifies a standard message language by setting out the encoding, semantics and pragmatics of the messages.

```
// Create a new message
ACLMessage  msg = new ACLMessage (
ACLMessage.QUERY_IF );

// Ask for price of ProductA (and also check if
available)
msg.setContent( "price ProductA" );

// Retrive Id of a stock server called supplier1
AID supplierId = new AID ( "supplier1",
AID.ISLOCALNAME);

// Add the supplierId to the message
msg.addReceiver ( supplierId );


// Send message
send(msg);

// Now wait for an answer...
msg = blockingReceive();
```

Figure 4. FIPA ACL Messagein MULEP

C. Supplier Subsystem

A supplier has to initialize a Supplier Subsystem on its machine to join the system. A supplier can join to the system over another supplier or directly over the main container.

When a supplier system is created for the first time, it subscribes to the system providing its address and the names of its products. If a supplier starts or ends delivering a product, it should also give necessary information to up-level suppliers or main container.

Every supplier agent has to know the address of the up-level suppliers so that it can make a connection. Supplier agent registers to the up-level suppliers by sending its product definitions and waits requests (as mobile agents) from them.

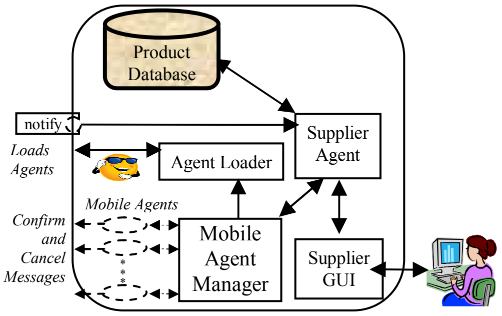


Figure 5. Supplier System Architecture

The Supplier Subsystem includes five main components as shown in Figure 5.

a. Product Database contains the services and product details of the supplier.
b. Graphical User Interface (GUI) which is used for interaction (i.e. inquiring about past and current transactions from the agent, or a user may also specify selling strategies through this module) with human users.
c. Mobile Agent Manager controls and coordinates the incoming mobile agents.
d. Agent Loader downloads the necessary classes from the up-level suppliers.
e. Supplier Agent acts on behalf of the user.

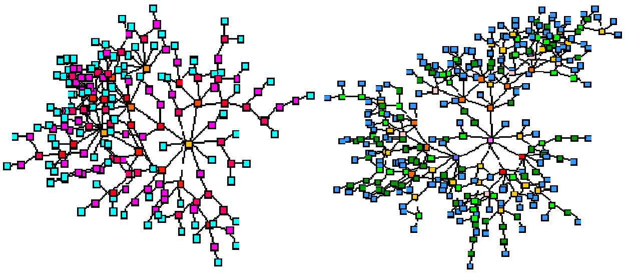
Supplier Agent is a stationary agent, which is responsible for offering an interface to end users to enter information, control the operations, and input query tasks. Supplier Agent processes purchase orders from buyer agents and decides how to execute transactions according to selling strategies specified by the user. Since organizations differ in the products they sell, a supplier agent should be customized before it is placed online.

Banking: In order for an agent-based marketplace to become anything more than an experimental platform, it has to be able to communicate with existing banking and financial services. When an agent needs to make a payment, it sends a request to its bank to withdraw funds, and receive a secure wrapper, called a check. Before sending out a check, the bank verifies the existence of sufficient funds in the agent's account.

V. Test Results

We ran a simple benchmark with a single buyer, one main container and a multiple number of suppliers, which were distributed over the system. The computational experiments of the system were performed by the same machines, Intel Core 2 Duo 2.66 GHz with 1 GB of DDR RAM and 4MB L2 Cache PCs running the Windows XP Pro operating system.

The experiment used a single order source whose order generation rate was slow enough to allow all the listeners to have processed a given order before the next was generated.



a) 200 Suppliers b) 300 Suppliers

Figure 6. Sample Topologies for E-Procurement System

System contains different numbers of suppliers which change between 100 and 500. Each supplier could have 0-8 sub-level randomly generated suppliers. These topologies can be generated by Boston University Representative Internet Topology gEnerator [14] as shown in Figure 6 and total dispatching time of mobile agent is shown in Figure 7. This figure shows the flexibility and adaptability of the system for large number of suppliers.

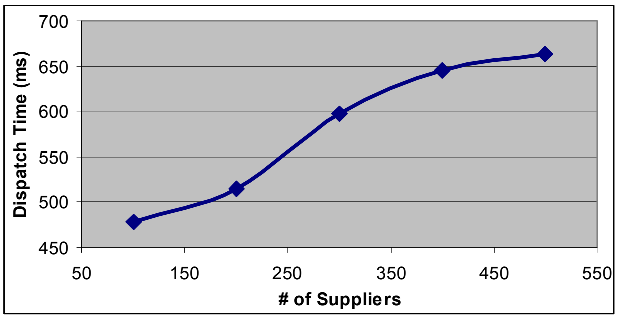


Figure 7. Dispathcing Time for Different Number of Suppliers

VI. CONCLUSION AND FUTURE WORKS

Electronic procurement is currently one of the most discussed topics in supply management. Without doubt, it will dramatically change the way purchasing is done in the near future. By means of this electronic marketplaces are approaching the old economic idea of a "perfect market world". We come close to this main target one more by developing MULEP-A Multi Level E-Procurement System with Distributed Agents which has a multi level agent architecture that supports the requirements of electronic procurement. Our system's communication infrastructure is based on the publish/subscribe paradigm, which allows participants to join and leave the system dynamically, extending the flexibility and adaptability of the system. System also facilitates the parallel computation by running mobile agents on suppliers concurrently to enhance performance.

In the proposed system, mobile agents are used as mediators between buyers and suppliers. The main advantage of using an agent-based approach is that the agents interact autonomously and power is devolved to the agents, i.e. there is no need for a super-agent to oversee all communication and interaction. The benefits of the proposed model are reduced network load, higher adaptability by allowing dynamic changes in system configuration, information hiding, asynchronous communication and flexibility of agent based execution. Implementation of a prototype system in Java and JADE is completed. As future work, we plan to apply the system in different application domains and focus on new design decisions to improve its performance and scalability.

REFERENCES

[1] Jennings, N.R., Norman, T.J., Faratin, P., O'Brian, P. and Odgers, B.: Autonomous agents for business process management, *Journal of Applied Artificial Intelligence* , Vol. 14, No. 2, pp. 145–189 (2000).

[2] Tewari, G. and Maes, P.: Design and Implementation of an Agent-Based Intermediary Infrastructure for Electronic Markets, *Proceedings of the 2nd ACM conference on Electronic commerce*, pp. 86–94 (2000).

[3] He, M., Jennings, N.R. and Leung, H.F.: On Agent-Mediated Electronic Commerce, IEEE Transactions on knowledge and data engineering, Vol. 15, No. 4, pp. 985–1003 (2003).

[4] N. M. Sadeh, D. W. Hildum, D. Kjenstad, and A. Tseng. MASCOT: An agent-based architecture for dynamic supply chain creation and coordination in the internet economy. Production Planning and Control, 12(3), 2001.

[5] Design of an intelligent agent-based supply chain simulation system, Yucheng Wang; Liping Fang, Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on Volume , Issue , Page(s):1836 – 1841, 7-10 Oct. 2007.

[6] Thierry Moyaux, Brahim Chaib-draa, and Sophie D'Amours,Supply Chain Management and Multiagent Systems: An Overview. In MultiAgent-Based Supply Chain Management, ISBN: 3-540-33875-6 (Springer). pp 1-27, 2006.

[7] Şahingöz O. K., Erdogan N., "A Two-Leveled Mobile Agent System for Electronic Commerce", Journal of Aeronautics and Space Technologies, Vol.1 No. 2, pp.21-32, July 2003.

[8] Lange, D.B. and Oshima, M.: Seven Good Reasons for Mobile Agents: Communications of the ACM, Vol. 42, March 1999, No. 3, pp. 88-89.

[9] Wooldridge, M. and Jennings, N. R.: Intelligent agents: Theory and practice: The Knowledge Engineering Review, Vol. 10, 1995, No. 2, pp. 115-152.

[10] Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa, JADE—A FIPA-compliant agent framework CSELT internal technical report. Part of this report has been also published in Proceedings of PAAM'99, London, April 1999, pp.97–108.

[11] F. Bellifemine, A. Poggi, G. Rimassa, Developing multi-agent systems with JADE, Proc. of the 7th International Workshop on Intelligent Agent. Theories, Architectures and Languages (ATAL-2000), LNCS, vol.1986, pp. 89-103, ISBN 3-540-42422-9, Springer.

[12] JADE white paper, http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf, accessed in 2009.

[13] FIPA Agent Communication Language Specifications, http://www.fipa.org/repository/ aclspecs.html, accessed in 2009.

[14] Medina, A., Lakhina, A., Matta, I., ve Byers, J., (2001). BRITE: An Approach to Universal Topology Generation. In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, Ohio, 346-356. t5fr