Implementation of Distributed Multi Agent System using JADE Platform

Sumit Prakash
PG Student,
BTKIT Dwarahat, Uttarakhand
India

Ajit Singh, Ph.D.
Associate Professor,
BTKIT Dwarahat, Uttarakhand
India

Pankaj Singh Sammal
PG Student,
BTKIT Dwarahat, Uttarakhand
India

ABSTRACT

Now a day's agents are more and more appears in computer science with different context and also with different meaning. The main acceptation of agents is due to the Artificial Intelligence and Distributed Artificial intelligence one, where the agents are essentially exploited as a one of the technique for developing special purpose systems that exhibit some kind of intelligent behavior. The main aim of this research is to implement distributed multi agent system using JADE platform. This system should allow the parameterization of the characteristics of each agent running on the system in order to simulate a virtual market buying and selling. In this research a scenario is created to simulate an electronic market place where multiple buyer and seller negotiate the purchase and sale of certain product to implement distributed multi agent system. The trading of the products will take into account several attributes such as price, delivery time, quality factor of the merchandise. Consist of an iterative process between the buyer and potential sellers. Each negotiation will begin with the publication of a need for a buyer, which meet all sellers the type of product ordered. The buyer will progressively select the best offer and the sellers tried to modify the conditions of sale to be chosen by the buyer. In addition to the sellers and buyers still exist a manager who will keep a simple record of reputation that each buyer / seller.

General Terms

Distributed multi agent system, Java Agent Development Framework, Foundation for Intelligent Physical Agents, E-Marketplace

Keywords

MAS, JADE, FIPA, ACL

1. INTRODUCTION

Multi-agent based simulation (MABS) has been receiving increasing interest in the recent years. One reason for this can be seen in the fact that the agent-paradigm allows the mapping of real world entities to autonomous software agents as a first approximation. Exploiting further skills of an agent, like the ability to communicate, learn or reason, can result in further benefits and new solutions. In fact, MABS as a sophisticated alternative to traditional simulation techniques attracts growing interest in a broad range of disciplines. Examples for their practical as well as scientific deployment are traffic simulations, crisis management, energy markets or scheduling problems.

A Multi-agent system is a system that consists of several agents that interact with each other. These Interactions are often handled by messages that are sent between the agents. These agents simulate intelligence by using methodical, procedural, functional or algorithmic search, for finding and processing approaches. Each of the agents can have different goals and behaviors, which together combines to a dynamic system.

A multi-agent system (MAS) is a network of loosely coupled software agents that interact with each other for solving problems which are beyond the knowledge or capacity of individual of each problem solver. The agents have some critical features that they are at least partially autonomous, no agent has a global view of the system or it cannot use this knowledge practically, there is no controlling agent. Multi-agent systems are very useful in solving problems that are impossible or difficult for monolithic system or an individual agent to solve. This could be problems like modeling social structures or simulating a trading market. A lot of work has already been done in the field of Multi-agent systems, as it is used for a wide variety of applications. In the research of artificial intelligence, the agent-based systems technology has been hailed with a new paradigm for designing, conceptualizing, and implementing software systems. Agents are nothing but sophisticated computer programs and acts autonomously on the behalf of their users, across the open and distributed environments, for solving problems which are very complex. But solving a complex problem however, the applications require multiple agents which are autonomous and interacting with each other and working together for. A Multi agent system works by distributing its capabilities and computational resources across the interconnected network of the agents. Whereas in centralized system, they are plagued by resource limitations, critical failure or performance bottlenecks, while a Multi agent System is a decentralized and that's why if a failure occur at a single point it will not affect the system and it keeps working but not like centralized system the whole system crashes.

A Multi Agent System is a system which allows for the interoperation and interconnection of number of existing legacy systems. By doing this like building an agent wrapper around such legacy systems, they could be incorporated into a society of agents. A Multi Agent System work out with problems in terms of autonomous agents component interacting with each other, which proves to be a more better way of representing the allocation of task, open environments, user preferences, team planning, and so on. A Multi Agent System is more efficient in retrieving, filtering, and globally coordinating information from the sources that are distributed spatially. A Multi agent system is a system which provides solutions in a situation like where expertise is temporally and spatially distributed. A Multi Agent System improves overall performance of the system, specifically in the field of reliability, computational efficiency, maintainability, flexibility, extensibility, responsiveness, robustness and reuse. But complex scenarios from different domains often bring their own complexity with them. It cannot be assumed that these domain-specific experts can understand, build or even control the execution of an agent-based simulation – simply due to its inherent complexity.

In this research work a new simulation framework that is based on JADE is created. On the one hand, JADE is extended by specific functionalities for simulation purposes like time and agent synchronization, agent/environment interaction, visualization and load balancing in order to simplify the work for an agent-based developer on such concerns as much as possible [6]. On the other hand, a main focus is on users, who are not familiar with multi-agent systems or distributed simulations. Here the framework provides a multi-language based GUI that can easily be customized to domain specific requirements. In this research paper the researcher focus on two important aspects of framework, the ability for extending the frameworks graphical user interface and the bidirectional interaction of agents with their environment by using our adapted service for simulations.

1.1 E-Marketplace

An Electronic Marketplace (E-Marketplace) or Digital Marketplace is a virtual marketplace where buying and selling is done over a communication network. The possibility of an E-marketplace is due to the advent of internet and World Wide Web. An E-Marketplace comprises entities like buyer, sellers, manager and product. The buyer and sellers buy and sell products respectively. Buyers look for information which they are interested in and sellers look for potential buyers who would buy their products. Many a times both the parties spend a lot of time in searching for information about each other. Then there is a negotiation between seller and buyer which is recorded by Manager Agent. A number of E-Marketplaces are ranging from CDs to Automobiles. An E-Marketplace has got the advantage e.g. it dramatically reduces transaction costs and on the other hand can enhance satisfaction of both buyer and seller. E-marketplace affects consumer purchase process. An E-Marketplace provides a mechanism for reducing the search costs (money, time and effort expended to gather product price, quality and feature information) for consumers. The combination of more information, electronic links and channels will give the buyers more choices, resulting in the shift of bargaining power to the Buyers. The product price, search costs, marketing (advertising) costs, overload costs, inventory costs and production costs are lower.

1.2 Distributed Multi Agent

A multi-agent system is a computerized system composed of multiple autonomous intelligent agents interacting with each other within an environment. Using MAS (multi agent system) can solve problems which are so much complex that they are difficult or not possible for monolithic system or an individual agent to solve. Intelligence might include some procedural, functional, methodic or algorithmic search, for finding and processing approach. Although there is a considerable overlap between a multi-agent system and an agent-based model but they are not always the same [10]. The main goals of an agent-based model is that it searches for explanatory insights into collective behavior of the agents those are obeying simple rules, that are typically in a natural systems, rather than in solving engineering problems or specific practical. This type of terminology of agent-based model tends to be more often used in the sciences and multi-agent system in engineering and technology. The topics where multi-agent systems research can deliver an appropriate approach include modeling social structures, disaster response and online trading. Multi agent System has been brought in research and development of distributed system. Many new multi-agent applications are currently being developed and launched. Developing distributed multi-agent applications is a difficult task. Main generic categories of domains where MAS find real applications are:

1.2.1 Distributed Situation Assessment: network diagnosis, information gathering on the Internet, distributed sensor networks, etc. Agents with different level of understanding and control, share local interpretation to achieve a common and consistent level of understanding and response.

1.2.2 Distributed Resource Scheduling and Planning: factory scheduling, network management, etc. – agents coordinate their schedules to resolve conflicts over resources and to maximize system output.

1.2.3 Distributed Expert Systems: concurrent engineering, network service restoral, etc. - agents are sharing information and work together (negotiation) to a common solution; each agent participate with different expertise and solution criteria [11].

1.3 JADE

The proposed framework has been developed on a widely used agent platform (JADE). JADE is a framework for developing multi-agent applications. It is provided as a FIPA-compliant agent framework with a package to develop Java agents. JADE is fully implemented in Java and it is totally open-source, it is FIPA compliant and runs on a variety of operating systems including Windows and Linux (and, as illustrated below, it is also possible to run JADE in a mixed environment).

In JADE framework, there is an agent container in each host to hold its local agents. Each agent is an active thread that has its own behaviors. Agents in JADE are executed concurrently. JADE provides a virtual agent platform, by which all agents can interact with each other, regardless of their containers or hosts. The main reason for this selection was the fact that JADE is one of the best modern agent environments. The choice of this platform for the implementation of the Multi-agent system was based on some advantages and criteria. JADE uses Java and each agent is run in a separate thread, which is faster than conventional Java threads. JADE is updated regularly and has a large development crew and community [6]. JADE works on any platform that supports a Java Virtual Machine, or JVM. It has an excellent GUI with a lot of useful features and tools.

It has already been used in a lot of development and research projects and has a high acceptance rate in the community. It supports the FIPA specification standard for Multi-agent system messaging. There are very good security features, such as SSL support for inter platform communications, permission grants and added security possibilities [7]. The platform is easy to distribute on multiple hosts. There is a wide range of different extensions and libraries for additional features, such as added security, web service integration and embedded JADE for small devices. It supports multiple communication and transport protocols, such as socket, RMI and IIOP Communication.

2. RELATED WORK AND APPROACHES

In Recent years there is much work has been done in Multi Agent System to implement distributed environment. A literature search shows that most of the related researches have been done Distributed Multi Agent System using Jade platform by following this:

According to [1] J. Bryan Logan they showed a framework for the distributed simulation of multi-agent systems which aims to overcome some of the deficiencies of the ad-hoc, centralized, time driven simulation approaches typically employed for agent simulation. Their framework uses the notion of 'spheres of influence' as a basis for dynamically partitioning the shared state of the simulation model into logical processes, and sketched an algorithm for dynamically partitioning the simulation to perform load balancing.

Jei Lars Braubach proposed a article on "Deployment of Distributed Multi-agent Systems" [2]. In this article, they have argued that deployment techniques are important for the wide-spread and industrial adoption of multi-agent system technology. They have investigated the general requirements and the extent to which existing deployment techniques can be adapted to support the launching and configuration of distributed multi-agent systems.

In year 2011 more research has been done on Distributed Multi agent system and a research paper on "A Framework for Monitoring the Execution of Distributed Multi-agent Programs" [3] was published. In this paper, the researcher has designed and implemented a framework for monitoring the execution of distributed multi-agent programs, detecting the occurrence of certain events of interest and reacting to these events accordingly

Marcel Koster shows his research in the field of Distributed Multi Agent System in year 2012. A proposed work was done in distributed Multi agent System through JADE in award of Master thesis on research paper "Reliable Multi-agent System for a large scale distributed energy trading network"[4]. The research group of this master thesis is Distributed Systems. The Distributed Systems group performs research and delivers education in all aspects of distributed information systems with particular emphasis on Service-Oriented Computing.

In 2014 a new architecture was designed on Multi agent system for solving distributed services and application and research paper was published on "A multi agent architecture on distributed services and applications"[5]. This paper describes a flexible user services oriented multi agent Architecture (FUSION@) and explains how this architecture has been designed and applied to real time scenarios. FUSION@ is a novel architecture which integrates SOA approach with intelligent agents for building systems based on the ambient Intelligence paradigm. The architecture focus on distributing the majority of system functionality into remote and local services and application.

From previous literature works, it appears that there are many research studies exploiting various techniques blended with Distributed multi-agent technology and JADE platform. Consequently, in order to success on e-Market Simulation, agent should have abilities to perform as a behalf of user to handle with business tasks such as planning, reasoning and learning.

3. AGENT DEVELOPMENT FRAMEWORK

3.1 Agents

An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. A generic agent has a set of goals (or intentions), certain capabilities to perform actions, and some knowledge (or beliefs) about its environment. To achieve its goals, an agent needs to reason about its environment (as well as behaviors of other agents), to generate plans and to execute these plans. On the other hand, an agent is a self-contained software program capable of controlling its own decision making and acting based on its perception of its environment, in order to gain one or more goals. The agent has the following main behavior attributes: autonomous, cooperative (social), reactive, proactive, and learning. Being autonomous means that agent can act without intervention by other entities (humans or computer systems), and exercising control over one's own action/algorithms. Agents can interact with other agents (or possibly humans) via some kind of agent communication language (social). To be reactive means that agents can perceive the environment around itself and are able to respond in a timely fashion to changes in the environment (reactive). In additions, agents do not simply act in response to the environment but are parts of a more complex goal-oriented behavior (proactive). Also, agents can change their behavior based on their previous experience (learning).

On the other hand, an agent is a software entity that has a set of protocols which govern the operations of the manufacturing entity, a knowledge base, an inference mechanism and an explicit model of the problem to solve. Agents communicate and negotiate with the other agents, perform the operations based on the local available information and may pursue their local goals. This definition has both technical and organizational aspects. Technically, agents possess sufficient knowledge and inferential capability to behave in a manner that would be classified as "intelligent" if performed by a person. Organizationally, agents are entrusted with sufficient authority to make commitments for users. This enables them to represent their principals and adhere to the same corporate rules, policies and procedures required to be followed by people in the organization. The common characteristics possessed by an agent are:

3.1.1 Autonomy: The agent is able to do at least part of its functionality independently and follow goals autonomously

3.1.2 Intelligence: The agent has some specialized knowledge in one or more application fields.

3.1.3 Interaction: The agent is able to collect information or to react on conditions of its environment.

3.1.4 Reactivity: An agent must be capable of reacting appropriately to inputs from its environment.

3.1.5 Pro-activity/Goal-Orientation: An agent does not just react to changes to its environment but it takes the initiative.

3.1.6 Learning: An agent has to change its behavior based on its previous experience.

3.1.7 Mobility: Mobility enables an agent to transport itself from one node of a network to another.

3.1.8 Communication/cooperation: An agent can use the communication capability to make contact with its environment.

3.2 Multi-Agent Systems

A multi-agent system is a computer program with problem solvers situated in interactive environments, which are each capable of flexible, autonomous, yet socially organized actions that can be, but need not be, directed towards predetermined objectives or goals. Thus, the four criteria for an intelligent agent system include software problem solvers that are:

3.2.1 The situatedness of an intelligent agent means that the agent receives input from the environment in which it is active and can also effect changes within that environment. Examples of environments for situated agents include the internet, game playing, or a robotics situation.

3.2.2 An Autonomous System is one that can interact with its environment without the direct intervention of other agents. To do this it must have control over its own actions and internal state. Some autonomous agents can also learn from their experience to improve their performance over time. For example, on the internet, an autonomous agent could do a credit card authentication check independent of other issues in the purchasing transaction.

3.2.3 A Flexible Agent is both intelligently responsive as well as proactive depending on its current situation. A responsive agent receives stimuli from its environment and responds to them in an appropriate and timely fashion. A proactive agent does not simply respond to situations in its environment but is also able to be opportunistic, goal directed, and have appropriate alternatives for various situations. A credit agent, for example, would be able to go back to the user with ambiguous results or find another credit agency if one alternative is not sufficient.

3.2.4 Finally, an Agent is Social that can interact, as appropriate, with other software or human agents.

On the other hand, a multi-agent system is one that consists of a number of agents that take specific roles and interact with one-another to solve problems that are beyond the capabilities or knowledge of any individual agent. These interactions can vary from simple information interchanges, to request for particular actions, and on to cooperation, coordination and negotiation in order to manage interdependent activities. The interaction and coordination are the core process of a multi-agent system. Agent technology and multi-agent system can also be used to develop highly complex systems. "Agent-based modeling is most appropriate for domains characterized by a high degree of localization and distribution and dominated by discrete decision". Supply chain system is a large-scale complex system. Decentralization, collaboration and intelligence are its essential characteristics. Multi-agent system is a fast developing information technology, where a number of intelligent agents, representing the real world parties, co-operate or compete to reach the desired objectives designed by their owners. The increasing interest in Multi agent system is because of its ability to provide robustness and efficiency; to allow inter-operation of existing legacy systems and to solve problems in which data, expertise, or control is distributed. The general goal of MAS is to create systems that interconnect separately developed agents, thus enabling the ensemble to function beyond the capabilities of any singular agent in the set-up. MASs try to solve the entire problem by collaboration with each other. In this way, MAS can help to solve complex problems and make decisions or support humans to make decisions. Therefore, agents are especially suitable for coordination of supply chains due to the following characteristics- Data, resources and control over data and resources are inherently distributed. And supply chain is adaptive and changes over time.

Agents can serve as wrappers for the supply chain management components owned by a particular supply chain entity. The advantage of Multi agent System is they increase efficiency and speed of simulation, due to asynchronous functioning. The agents are famous for their robustness and liability; if one agent fails other agents can perform the same roles. Scalability and flexibility, it is possible to adapt the system according to the problem. More cost effective, because implementation can be simpler than using mathematical methods. Reusability of agents that can be developed by experts and innovation to develop new technological application. Useful, when information is scarce.

MAS can also be the most suitable method to distributed problems. These problems are complex and multifaceted (e.g., vehicle production) or only solvable if decomposed, or that means an important cost reduction (e.g., monitoring of a wide geographic area) or lead to more efficacy (e.g., product delivery). According to this definition, it is hard to identify a problem that is inherently distributed; many problems are solvable in both a distributed or centralized way, the choice depends on specific characteristics. This concept appears as a mean to obtain solutions to different problems or to those that can be solved using fewer resources. Mechanisms to model the organizational structure. Despite these advantages and disadvantages, if there are some modules that are clearly generic that can be reused in other applications, we clearly gain when developing new applications using agent-based technologies. Additionally, there is a problem of complexity and characteristics of the problem to solve. Moreover, MAS can also have an important role when there is no analytical solution or when the problems are mainly distributed, and because of that, MAS are the most natural and understandable solution for users [9].

Multi-agent systems are ideal for representing problems that include many problem-solving methods, multiple viewpoints, and multiple entities. In these domains, multi-agent systems offer the advantages of distributed and concurrent problem solving along with the advantages of sophisticated schemes for interaction. Examples of interactions include cooperation in working towards a common goal, coordination in organizing problem-solving activity so that harmful interactions are avoided and beneficial possibilities exploited, and negotiation of sub problem constrains so that acceptable performance ensues. It is the flexibility of these social interactions that distinguishes multi-agent systems from more traditional software and which provides the power and excitement to the agent paradigm [8].

3.3 Multi-Agent Negotiation

Negotiation is a discussion among conflicting parties with the aim of reaching agreement about a divergence of interests. Negotiation is used as a coordination mechanism to find an acceptable agreement between partners or to collectively search for a coordination solution. Negotiation may involve two parties (bilateral negotiation) or more than two parties (multilateral negotiation) and one issue (single-issue negotiation) or many issues (multi-issue negotiation). Negotiation represents a key form of interaction in agent mediated electronic markets that transcend the sale of uniform goods. Though negotiation, suppliers and consumers can reach complex agreements in an iterative way, which better match the needs and capabilities of different parties. Negotiation may end with either agreement or no agreement. Failure to agree can occur in two ways the first one is either party decides to opt out unilaterally, or the two do not agree to any proposal. The resistance points or limits play a key role in reaching agreement when the parties have the ability to unilaterally opt out of the negotiation − they define the worst agreement for a given party which is still better than opting out.

In a system that composes of multiple autonomous agents, the negotiation is a key that form an interaction that enables the groups of agents in arriving at a mutual agreement regarding some plan, goal or belief, for example. Particularly as the agents are autonomous and they cannot be assumed to be benevolent, the agents influence others to convince them to act in certain ways, and negotiation is thus critical for managing such inter-agent dependencies. There are three broad topics in research for negotiation that serves for organizing the issues under its consideration. *First*, the negotiation protocols that are the set of rules that governs the interaction between agents. *Second*, the negotiation objects those are the range of issues over which agreement must be reached between the parties. Finally, the agents' reasoning models provide the decision making apparatus by which participants attempt to achieve their objectives.

3.3.1 Pre-Negotiation is the process of preparing and planning for negotiation and involves mainly the creation of a well-laid plan specifying the activities that negotiators should attend to before actually starting to negotiate. Effective pre-negotiation requires that negotiators prioritize the issues and define the targets. Priorities are set by ranking-order the issues, i.e., by defining the most important, the second most important, and so on. Additionally, effective pre-negotiation requires that negotiators agree on an appropriate protocol that defines the rules governing the interaction. The negotiation literature describes several protocols that vary significantly depending on the type and amount of information exchanged between agents. Simple protocols allow agents to exchange only proposals, i.e., solutions to the problem they face.

Richer protocols allow agents to provide feedback on the proposals they receive. This feedback often takes the form of critiques, i.e., comments on which parts of proposals are acceptable or unacceptable. Sophisticated protocols allow agents to provide arguments to support their negotiation stance.

3.3.2 Actual Negotiation Actual negotiation is the process of moving toward agreement (usually by an iterative exchange of offers and counter-offers). The negotiation protocol defines the states (e.g., accepting a proposal), the valid actions of the agents in particular states (e.g., which messages can be sent by whom, to whom, at what stage), and the events that cause states to change (e.g., proposal accepted). It marks branching points at which agents have to make decisions according to their strategies. Thus, at each step of negotiation, agents often need to follow their strategies to choose among different possible actions to execute.

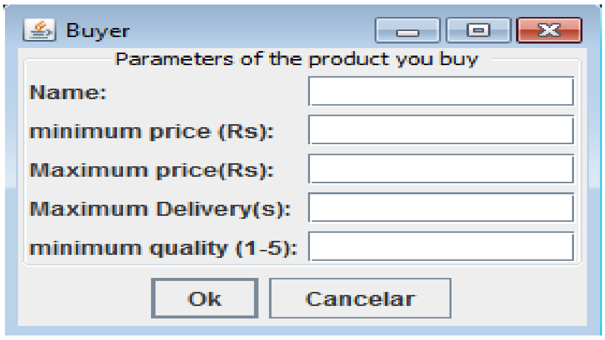
Negotiation techniques are used to overcome conflicts and coalitions, and to come to an agreement among agents, instead of persuading them to accept a ready solution. In fact, negotiation is the core of many agent interactions because it is often unavoidable between different project participants with their particular tasks and domain knowledge whilst they interact to achieve their individual objective as well as the group goals [12]. The importance of negotiation in MAS is likely to increase due to the growth of fast and inexpensive standardized communication infrastructures, which allow separately, designed agents to interact in an open and real-time environment and carry out transactions safely.

4. THE PROPOSED SYSTEM

4.1 Specification

4.1.1 Identification and Characterization of Agents

There are two main types of agents, buyer and seller as shown in the Figure 1 and 2. Both agents have a type of product that trade as well as a minimum and maximum amount you are willing to transact. Also have an associated term delivery, and the buyer this is the maximum period for which the delivery will have to be satisfied in the case of the seller and the fastest time this can make the product available. Finally there is another common attribute that represents the quality of the product transaction represented a scale of 1 (low) - 5 (very high). The seller will have an associated product quality and buyer will require a minimum for this quality. This field can influence the decision of a buyer who can pick a product of superior quality over price.



Buyer

Parameters of the product you buy

Name:

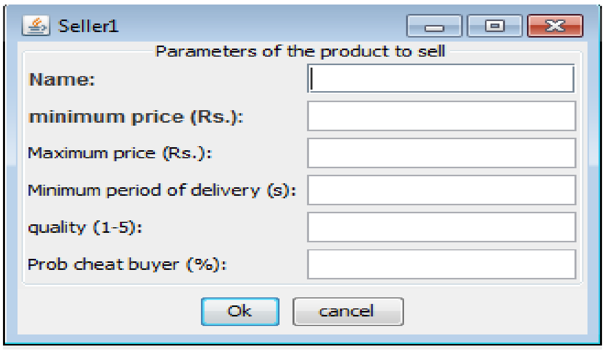
minimum price (Rs):

Maximum price(Rs):

Maximum Delivery(s):

minimum quality (1-5):

Figure 1: Initial parameter of Buyer Agent



Seller1

Parameters of the product to sell

Name:

minimum price (Rs.):

Maximum price (Rs.):

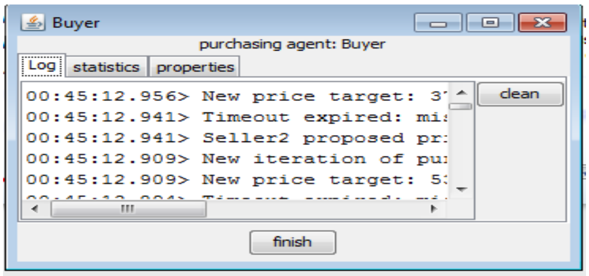
Minimum period of delivery (s):

quality (1-5):

Prob cheat buyer (%):

Figure 2: Initial parameter of Seller Agent

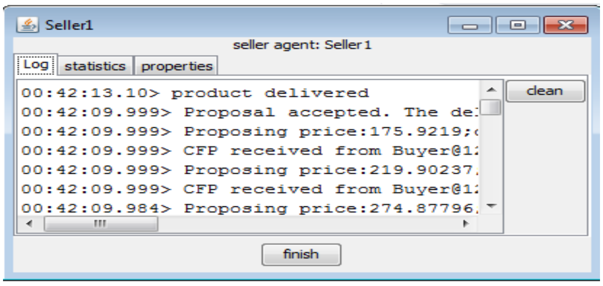
4.1.1.1 Buyer Agent - This agent is responsible for starting the communication in the transaction process by publicizing their need to all Sellers on the type of product that it wants to acquire as shown in figure 3. Go after receiving bids in an iterative process of selling various vendors and select the best in terms of price, quality and delivery time until you find one that pleases you.



```
00:45:12.956> New price target: 3'
00:45:12.941> Timeout expired: mi:
00:45:12.941> Seller2 proposed pr:
00:45:12.909> New iteration of pu:
00:45:12.909> New price target: 5:
```

Figure 3: Progress window of Buyer searching for Sellers

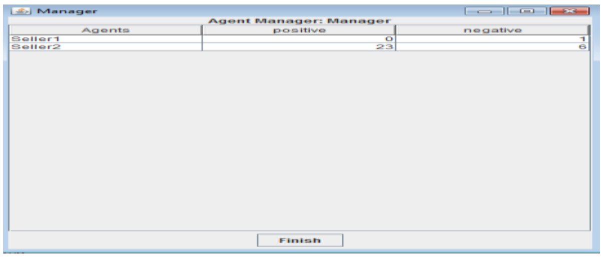
4.1.1.2 Selling Agent - This agent provides a type of product to buyer’s agents, having a minimum selling price as shown in figure 4. After receiving a request from a buyer by sending your proposal, if it is rejected then you may (if their parameters still allow leeway) revised its proposal and submit a new, more attractive to the buyer. This agent has a margin of error associated so not always meet the contracts that they propose.



```
00:42:13.10> product delivered
00:42:09.999> Proposal accepted. The de:
00:42:09.999> Proposing price:175.9219;
00:42:09.999> CFP received from Buyer@1:
00:42:09.999> Proposing price:219.90237.
00:42:09.999> CFP received from Buyer@1:
00:42:09.984> Proposing price:274.87796.
```

Figure 4: Progress window of a transaction from a Seller

4.1.1.3 Manager Agent – as shown in the figure 5 this agent collects information about the different transactions that are being made and who made them. Permits to associate a reputation to buyers / sellers agents based on their historical fulfillment of contracts established. Jade DF - is a central register of entries linking agent’s services? This service is used for both actions. One for registration of new agents (buyers and sellers), registering your Agent ID and all relevant information to the product which they trade. Another search for potential partners that have the product that you are looking at a purchase order from a buyer's agent.



Agents	positive	negative
Seller1	0	1
Seller2	23	6

Figure5: Manager Agent collecting information about the Reputation of sellers

4.1.2 Interaction protocols

The buyer / seller agents will use a protocol extension to FIPA-ContractNet in its iterative aspect. In this protocol, two entities are distinguished, the Initiator and the Participant. The buyer agent implements like behavior Initiator using the class of Jade ContractNetInitiator. The seller's agent implements like behavior using the class Participant Jade SSIteratedContractNetResponder. The only exception to this protocol will be FIPA-Cancel-Meta-Protocol, common to many other protocols.

4.2 Development

4.2.1 Platform / Tool used

In this research platforms used NetBeans and Eclipse, running on Ubuntu and Windows7 respectively. These platforms have a development interface for very long java, easy to use, simply add the jar file JADE so that you have a complete and very powerful development environment JADE. They also come also with automatic graphical interface editors, as well as automatic query generation and documentation, making it the preferred choice for this job.

4.2.2 Structure of the Application

The application is divided into 4 basic packages: buyer, seller, manager and useful as shown in figure 6. The buyer and seller package implementing both the FrmGUI and FrmParam classes. These two classes provide a graphical interface to the main window and the agent respectively Creation window. Both have the Agent class that implements the core processes creating the corresponding agent JADE and its behavior. Once the manager does not have this parameter implements only FrmGUI. The manager also has the reputation classes corresponding to a reputation attributed to an agent buyer / seller. This reputation is based on compliance or non-compliance with contracts established by the buyer / seller and data stored on 2 agents, the number of positive votes and the number of negative vote’s .The package includes 3 useful helper classes with functions of DF, the graphical user interface and ranking functions of agents to their reputation and use by the manager.

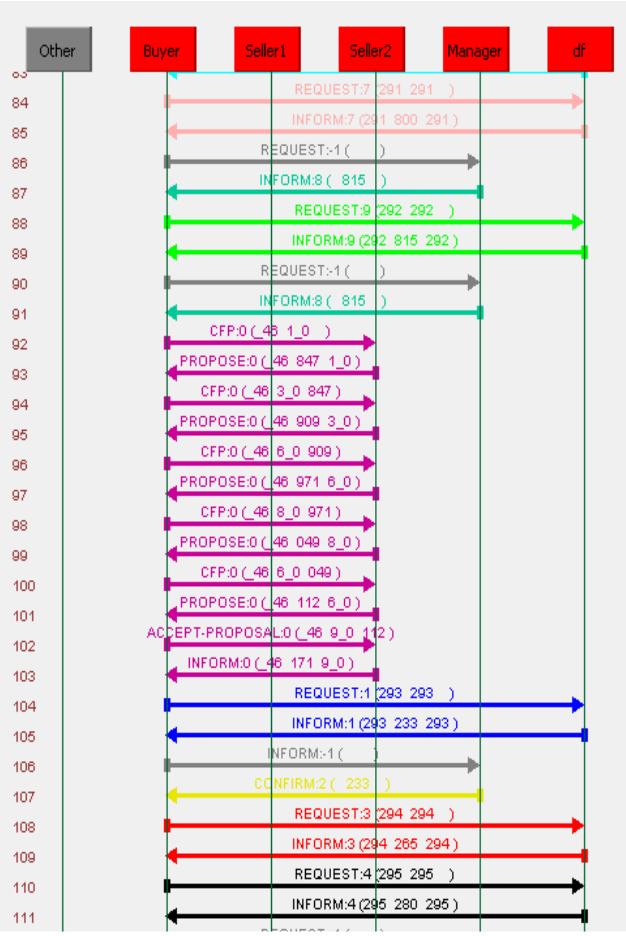


Figure 6: The operation of JADE in negotiation between Buyer and Sellers and communication between Agents

4.2.3 Relevant details of the Implementation

All agents are recorded in the DF, allowing buyer's agent's seller's agents to easily find the desired product. Buyer's agents and sellers have minimum and maximum initial rates respectively. These prices will be "objective "that agents try to achieve initially, following the guide. These values will be increased or decreased (but never exceed the limits) of negotiations with various events, events such as accepting a proposal or forward to another round of negotiations. The seller has a probability of deceiving the buyer when responding to a "Call for Proposal ". When this happens , the seller advises that sells a quality product 5 (maximum) and can deliver the product in a second .When the buyer accepts an offer , the seller waits some time (the deadline) before informing the buyer of success, simulating the delivery time , and to be sometime between negotiations. The buyer, to evaluate the different proposals and decide which best takes into account the various attributes. The base score is the price, and then multiplied by less than 1 value according to the other attributes (the better the proposal of the attribute with respect to the limit in the same buyer, the greater the reduction). The best score is the lowest. The classification of attributes is made by the buyer in relation to the attributes of the best proposal for the term and for the quality and for the price, is made as to the minimum price (objective). The possible ratings are "good", "sufficient " and "bad".

4.3 Experiences

4.3.1 Purpose of each Experiment

Basic experiment - In this experiment two agents buyer and seller are placed to make a transaction in a single product, first ensuring that were able to complete the transaction to verify a transaction successfully, then adverse conditions to test transactions impossible. This is the simplest test drive the transaction process. Basic experience with manager similar to previous experience but with a manager to see if it successfully recorded the transactions and votes on reputation. Experience the load - this experiment multiple sellers and buyers agents have been created in order to try to overload the system and see how it behaves when multiple agents buy and sell the same product simultaneously. Different products as well as a second step. Experience and reputation of deception - Forcing the reputation of selling agents by authorizing several buyers were designed to test how they reacted the reputations of the Sellers in the system were also forced the likelihood of deception to see how he reacted the reputation system promised when the contract was not fulfilled.

4.3.2 Results

The basic experience successfully concluded, it was possible to see the transaction in progress and all this iterative steps in the figure 7 shown below. The basic experience with manager also concluded positively, where it can be seen that the manager kept the transactions and reputations of the agents properly. The experience of burden had apparent success but it was hard to visualize the results and transactions taking place in the midst of so much data. The experience and reputation of cheating agents responded well to forced change, responding negatively when the contract was not fulfilled and not negotiating with Sellers of low repute.

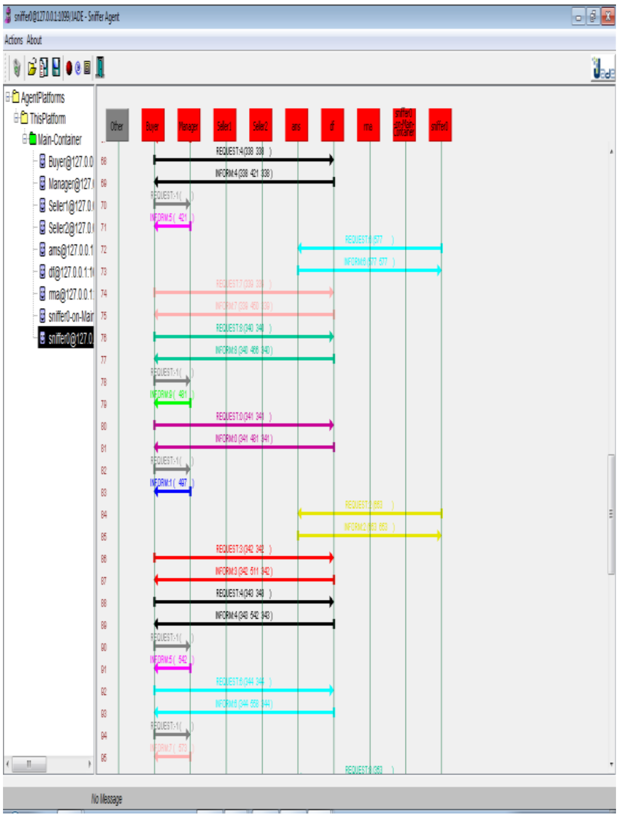


Figure 7: The Operation of JADE in E-Marketplace System

5. CONCLUSION

In this paper we introduced a framework for Multi-agent based simulations called Agent.GUI that is built on top of the JADE platform. Based on our frameworks base-GUI it allows the programmer to realize a domain specific end user application for Multi-Agent based simulations. For this purpose Agent.GUI provides open and adaptive interfaces.

The experiments carried out were generally all completed successfully. The program meets the objectives it had set and there is room for improvement. This program was made with generic attributes to be easily used for any type of market. The program has a graphical interface has been presented as a difficulty viewing this process, even this may be difficult to see the transaction when the system is overloaded with requests, so for testing purposes was sometimes locked code to test module by module to ensure proper operation. After implementing the scenario of proposed system shows intelligent behavior of the agents through distributed knowledge and platform. Using the agent platform and implementing and testing the simulating technique through Java agent development (Netbeans IDE) platform show how agent behave rationally and autonomously interact with each other to provide an optimal solution as well as reduction of domain complexity through distributed platforms. Agent coordination, competition, concession and negotiation with each other show an artificial intelligence environment. Very important conclusion is drawn from the full experiments report about agent distribution. Because agents in the implemented system were representing the amount of work they were distributed.

6. FUTURE WORK

The upgrades to this system could be connected with these three key points. First there should be Automatic creation of agents, allowing agents to launch randomly running. Every time the Agents can be generated automatically. Second there should be inclusion of a greater number of possible attributes in the description of the proposal, so the system become more flexible and more understandable. Third there should be improved algorithm of choice of tenders.

7. ACKNOWLEDGMENTS

The author would like to acknowledge all those people who helped in realizing in this research work and Dr. Ajit Singh**,** Associate Professor, Department of Computer Science and Engineering, Bipin Tripathi Kumaon Institute of Technology, Dwarahat for his expert guidance, valuable suggestion and constant supervision throughout this research work.

8. REFERENCES

[1] J. Brian Logan, Georgios Theodoropoulos, "The Distributed Simulation of Multi-agent Systems", 2000.

[2] Jei Lars Braubach[1], Alexander Pokahr[1], Dirk Bade[1], Karl-Heinz Krempels[2], and Winfried Lamersdorf[1,] "Deployment of Distributed Multi-agent Systems", ESAW 2004, LNAI 3451, pp. 261–276, 2005._c Springer-Verlag Berlin Heidelberg 2005

[3] Eslam Al Maghayreh, Samer Samarah, Faisal Alkhateeb, Iyad Abu Doush, Izzat Alsmadi and Ahmad Saifan, "A Framework for Monitoring the Execution of Distributed Multi-agent Programs", 2011

[4] Marcel Koster, "Reliable Multi-agent System for a large scale distributed energy trading network", Master Degree Thesis, 2010/2011

[5] Juan M.Corchado[1], Dante I.Tapia[2], Javier Bajo[1],"A multi agent architecture on distributed services and applications", Volume 8,Number 4,April 2014

[6] JADE, Java Agent Development Environment, 2006, "http://jade.tilab.com" accessed March 17, 2014

[7] FIPA: the Foundation for Intelligent Physical Agents, "http://www.fipa.org" accessed March 17, 2014

[8] Fabio L. Bellifemine, Giovanni Caire, and Dominic Greenwood."Developing Multi-Agent Systems with JADE"**.** Wiley, April 2007

[9] Phillip J. Turner and Nicholas R. Jennings." Improving the scalability of Multi-Agent Systems in Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems". Lecture Notes in Computer Science, Vol. 1887/2001,

[10] H. Mehl and S. Hammes, "Shared variables in distributed simulation," in Proceedings of the 7th Workshop on Parallel and Distributed Simulation (PADS93)

[11] D. Conklin, J. Cleary, and B. Unger, "The Sharks World: A study in distributed simulation design," in Proceedings of the Multiconference on Distributed Simulation,, San Diego

[12] Bartolini C., Preist C. and Jennings N.R. (2002)." A Generic Software Framework for Automated Negotiation"**.** In Proc. 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Bologna Italy, 2002.