International Journal of Research in Engineering, Science and Management
Volume-1, Issue-11, November-2018
www.ijresm.com | ISSN (Online): 2581-5792

Analysis of Distributed Multiagent System with JADE Platform

V. Banumathy[1], A. Mary Christina[2]
[1]*Lecturer, Dept. of Computer Engineering, A. D. J. Dharmambal Polytechnic College, Nagapattinam, India*
[2]*Lecturer, Department of Computer Engineering, Srinivasa Subbaraya Govt. Polytechnic College, Puthur, India*

***Abstract*: Agent- oriented software engineering (AOSE) is quickly emerging in response to urgent needs in both software engineering and agent-based computing. From past few years there is an improvement in Multi agent System. Many problems of distributed System are solved via multi agent system .The main aim of this work is to implement distributed multi agent system using JADE platform. This system should permit the parameterization of the characteristics of each agent running on the system in order to simulate a place for buying and selling. In this work a scenario is created to simulate a 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 numerous attributes such as price, delivery time, quality factor of the merchandise. Consist of an automation process between the buyer and potential sellers**

***Keywords*: Distributed MAS, AOSE, JADE, software, multi-agent, FIPA**

1. Introduction

Agent-Oriented Software Engineering (AOSE) is a new software engineering paradigm that arose to apply finest practice in the development of complex Multi-Agent Systems (MAS) by focusing on the use of agents, and organizations of agents as the main abstractions. The field of Software Product Lines (SPL) covers all the software development lifecycle required to develop a family of products where the derivation of concrete products is made systematically and rapidly.

A growing number of computer systems are being viewed in terms of autonomous agents. Agents are being espoused as a novel theoretical model of computation that more closely reflects current computing reality than Turing Machines [1]. Agents are being advocated as a next generation model for engineering complex, distributed systems [2, 3]. Agents are also being used as an overarching framework for bringing together the component AI sub disciplines that are essential to design and build intelligent entities [4, 5]. Yet despite this deep interest, a number of fundamental questions about the nature and the use of the agent-oriented approach remain unanswered.

Existing results in MAS research facilitate a developer to construct MAS easier than before. Among others, there are tools that can produce entire MAS from a specification, libraries of components that deal with concrete MAS issues (distributed planning, reasoning, learning), and theories that describe MAS behavior and properties. Knowing all of them needs a great effort. Existing surveys assist this task, but it is difficult to give an overall view of what software, theories, and methodologies exist, and how they are applied to MAS development.

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

Multi Agent Systems (MAS) are suitable and powerful tools adopted to model and simulate different types of complex systems [6]-[8]. They are mostly used because they enable intuitive modeling and rapid simulation of complex situations. To deal with realistic complex systems, large scale agent-based simulations are often required. However, considering the large number of the simulated entities and their complex behaviors, scalability becomes a challenging problem for such simulation environments. Distributed infrastructure such as clusters, grids and clouds are powerful computational environments that can be efficiently used to run large-scale agent-based simulations. However, the existing literature does not provide an overall study about the existing solution for distributing a given MAS. The main challenge, for MAS designers and developers is how to select the appropriate partitioning mechanism for a given MAS. There is no generic approach for guiding the designers and developers to select an appropriate approach for partitioning a given agent-based system. Therefore, in this work, we present a generic conceptual framework JADE for: 1) Analyzing existing partitioning methods and 2) It can also be used as a basis while designing a distributed architecture of new MAS based on relevant criteria. The major reason for this selection was the fact that JADE is one of the best modern agent environments. The selection 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 usual Java threads. JADE is updated frequently and has a large development crew and community. JADE works on any platform that supports a

Java Virtual Machine, or JVM. It has an excellent GUI with a lot of valuable features and tools. It supports multiple communication and transport protocols, such as socket, RMI and IIOP Communication.

2. Related work

Present a load management mechanism for distributed simulations of multi-agent systems. The mechanism reduces the cost of accessing the shared state in the distributed simulation by dynamically redistributing shared state variables according to the access pattern of the simulation model [14].

Present PDES-MAS, a distributed simulation engine for large scale multi-agent simulations. The system uses a space-time DSM approach to adaptively and dynamically divide the shared state of the MAS and addresses the issues of load balancing, synchronization and interest management in an integrated transparent manner [15].

Propose a reference model that specifies the usual launching and configuration infrastructure. The reference model is based on the ideas of agents and societies as constituting entities. For the reference model a FIPA-compliant service interface has been designed, which permits applications to be started on different hosts and possibly on different platforms [16].

Illustrate a detailed implementation of multi agents in an artificial stock market invoking the agent-based methodology on Java Agent Development (JADE) environment, a platform to extend multi-agent systems. The Extended Glosten and Milgrom Model, an agent based artificial stock market model, has been selected to depict the multi-agent environment model in JADE [17].

3. Existing methods

In this we analyze the existing relevant partitioning approaches that were devoted to be used in the case of MAS.

A. Cluster-based partitioning methods

The comparative study of the algorithm is based on three metrics which are Number of messages to measure the communication, CPU load and Number of migrations. Four scenarios were defined in order to study the influence of the movement of agents as well as the situation. While the k-means algorithm shows more efficiency in a few scenarios, it becomes worse when agents are moving individually. Therefore clustering agents into partitions based on the position may not be effectual in the cases where agents are acting individually.

B. Grid-based partitioning methods

Partitioning the grid for parallel processing has three main goals:

- Make partitions with equal numbers of cells.
- Reduce the number of partition interfaces--i.e., decrease partition boundary surface area.
- Lessen the number of partition neighbors.

Balancing the partitions guarantees that each processor has an equal load and that the partitions will be ready to communicate at about the same time. Since communication between partitions can be a relatively time-consuming process, minimizing the number of interfaces can decrease the time associated with this data interchange. Minimizing the number of partition neighbors minimize the chances for network and routing contentions. In addition, minimizing partition neighbors is significant on machines where the cost of initiating message passing is expensive compared to the cost of sending longer messages. This is particularly true for workstations connected in a network.

An approach to distribute shared spatial environment in MAS consisting of considering the spatial environment as a grid that can be separated into multiple spatial regions. The formed region is allocated to diverse Logical Processes (LPs). A situated agent is hosted to the consequent LP. Grid-based clustering algorithm dedicated to partition and load-balance crowd simulations among multiple processors but it is less capable to balance the load during runtime.

4. Proposed Framework

The proposed framework has been developed on a broadly used agent platform (JADE). JADE is a framework for developing multi-agent applications. It is provided as a FIPA (The Foundation for Intelligent Physical Agents)-compliant agent framework with a package to extend Java agents. JADE is fully implemented in Java and it is fully open-source, it is FIPA compliant and runs on a variety of operating systems including Windows and Linux.

JADE is a middleware which facilitates the development of multi-agent systems under the standard FIPA for which intention it creates multiple containers for agents, each of them can run on one or more systems. Is understood that a set of containers constitutes a platform.

JADE offers an environment where JADE agents are executed, the Class Libraries to create agents using heritage and redefinition of behaviors and a graphical toolkit to monitoring and managing the platform of Intelligent Agent agents.

Agent is a computer program that acts for a user or other program in a relationship of agency. The term "agent" describes a software abstraction, an idea, or a concept, similar to OOP terms such as methods, functions, and objects. The concept of an agent offers a convenient and powerful way to illustrate a complex software entity that is capable of acting with a certain degree of autonomy in order to accomplish tasks on behalf of its host. But unlike objects, which are defined in provision of methods and attributes, an agent is defined in terms of its behavior.

By an agent-based system, we signify one in which the key abstraction used is that of an agent. Agent-based systems may contain a single agent [11], but arguably the greatest potential lies in the application of multiagent systems [12].

The following are the properties of agents:

Autonomy: agents encapsulate a few state and make decisions

about what to do based on this state, without the direct involvement of humans or others;
Reactivity: agents are located in an environment, which are capable to perceive this environment and are capable to respond in a timely fashion to changes that occur in it;
Pro-Activeness: agents do not simply act in response to their environment, they are able to demonstrate goal-directed behavior by taking the initiative;
Social Ability: agents interact with other agents via some kind of agent-communication language [13], and normally have the ability to engage in social activities in order to achieve their goals.

The agent is a key element in the design of multi-agent systems. Ferber in [9], defines it as a physical or virtual entity that can act, perceive its environment and communicate with others, is autonomous and has skills to attain its goals and tendencies. Also, Wooldridge [10] presents the agent as an autonomous entity situated in some environment and having the following features: reactivity, proactivity and social ability. For Purely communicative MAS, in addition the common attributes such as, goals and internal state, the agents have more social abilities and they use direct contact to communicate and cooperate. Also, agent's goal is a criterion that can be used to distribute agents into partitions.

5. Distributed Multiagent System

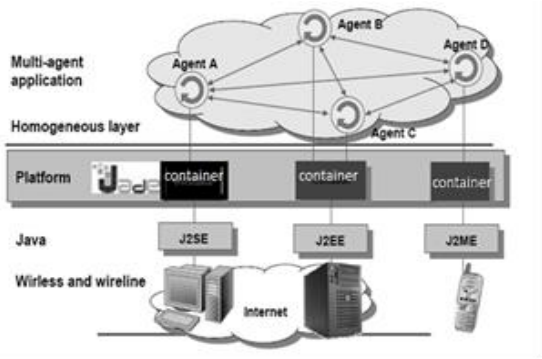

Fig. 1. Framework architecture

A multi-agent system is a computerized system composed of multiple interacting intelligent agents in an environment. Multi-agent systems can be used to crack problems that are difficult or impossible for an individual agent or a monolithic system to solve. Intelligence may comprise some methodic, functional, procedural approach, algorithmic search or reinforcement learning. Although there is significant overlap, a multi-agent system is not always the same as an agent-based model (ABM). The aim of an ABM is to search for explanatory insight into the collective behavior of agents obeying simple rules, typically in natural systems, rather than in solving specific practical or engineering problems. A Multi Agent System is more capable in retrieving, filtering, and globally coordinating information from the sources that are distributed spatially. A Multi agent system is a system which offers solutions in a situation like where expertise is temporally and spatially distributed. A Multi Agent System improves overall performance of the system, particularly in the field of reliability, computational efficiency, maintainability, flexibility, extensibility, responsiveness, robustness and reuse.

In our proposed method, there are three main types of agents, buyer, seller and JADE. Both seller and buyer agents have a type of product that trade as well as a minimum and maximum amount you are eager 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 obtainable. Finally there is another general attribute that represents the quality of the product transaction represented a scale of 1 (low) - 5 (very high). The seller will have a related product quality and buyer will require a minimum for this quality. This field can influence the decision of a buyer who can select a product of superior quality over price.

A. *Sellers*

The sellers register into the JADE with their unique name and their address. After registration they send the information about the products they contains into the JADE. They add and update the products into JADE. Whenever the buyers request for products the JADE agent verifies the sellers and connect the sellers to buyers. Here we take sellers as book sellers. The sellers add the book titles and the cost of books. The sellers connect to the buyers directly or indirectly whenever requires.

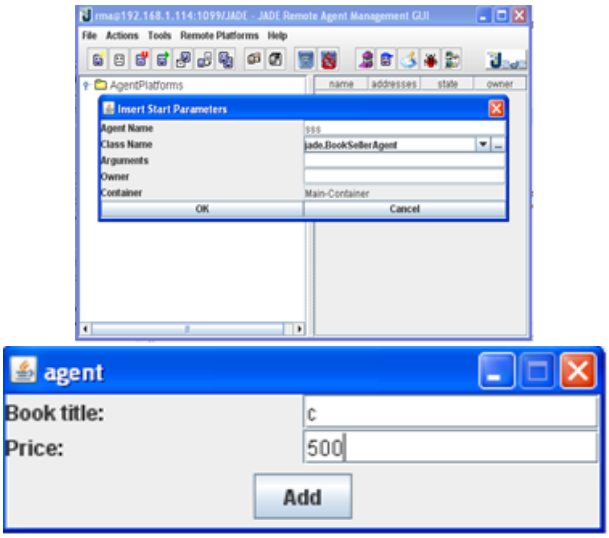

Fig. 2. Progress window of sellers for add and update the books

B. *Buyers*

The buyers send the request to the JADE agent for various products they require before that they want to register into JADE. Immediately the agent sends an automatic reply to the buyer after receiving the request from them. The agent reply contains the available products related to the request and the price of products. If the buyer satisfied with the price they can

buy the products. Here we take buyers as book buyers. The book buyers request the books with titles. The buyers connect to the sellers directly or indirectly.

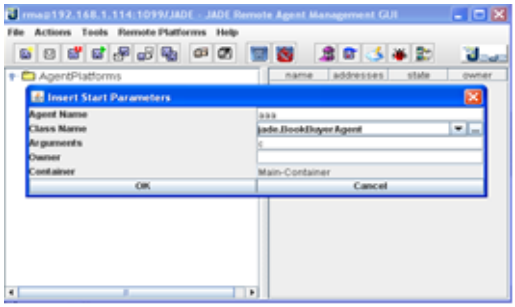

Fig. 3. Progress window of buyers searching for books JADE Agent

This agent collects information about the different transactions that are being made and who made them. Permits to correlate 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 which are used for both actions. One for registration of new agents, registering your Agent ID and all relevant information to the product which they trade, another one for potential partners that have the product that you are looking at a purchase order from a buyer's agent. After receive the requests from the buyers it automatically starts searching the agents contains the requested books and send the reply message to the buyers regarding the agents.

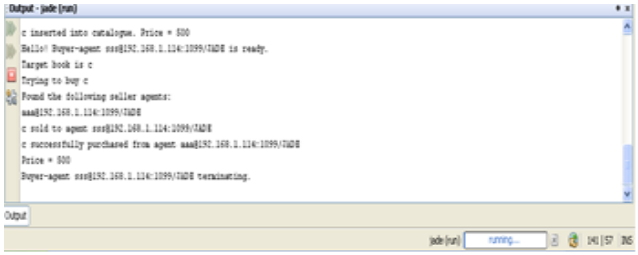

Fig. 4. Collecting information about the sellers contain the requested books

In this platforms used NetBeans, running on 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 an absolute and very powerful development environment JADE. 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.

6. Details regarding implementation

All agents are recorded in the Directory Facilitator (DF), allowing buyer's agents, seller's agents to easily find the desired product. DF keeps track of all advertised services offered by all the agents in the same Jade platform. After registered into JADE if the buyers send the request to JADE it automatically search the products required by the buyers and connect buyers to the respected sellers.

7. Results

The basic experience successfully concluded, it was potential to see the transaction in progress and all this iterative steps. The basic experience with JADE also concluded positively, where it can be seen that the JADE kept the transactions and reputations of the agents properly. The experience of burden had apparent success.

8. Conclusion

In this work we introduced a framework called JADE for Multi-agent based simulations. Based on our frameworks base-GUI it allows the programmer to realize a domain specific end user application for Multi-Agent based simulations. Using the agent platform and implementing and testing the simulating technique through Java agent development platform show how agent behave rationally, autonomously and automatically 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 about agent distribution. Because agents in the implemented system were representing the amount of work they were distributed.

References

[1] P. Wegner, Why interaction is more powerful than algorithms, Comm. ACM 40 (5) (1997) 80–91.
[2] N.R. Jennings, M. Wooldridge, Agent-oriented software engineering, in: J. Bradshaw (Ed.), Handbook of Agent Technology, AAAI/MIT Press, 2000, to appear.
[3] M. Wooldridge, Agent-based software engineering, IEE Proc. Software Engineering 144 (1) (1997) 26–37.
[4] N.J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, San Mateo, CA, 1998.
[5] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, Englewood Cliffs, NJ, 1995.
[6] P. Egan, J. Cagan, C. Schunn and P. LeDuc, 'Design of Complex Biologically Based Nanoscale Systems Using Multi-Agent Simulations and Structure Behavior Function Representations', J. Mech. Des., vol. 135,no. 6, p. 061005, 2013.
[7] A. Lopez-Paredes, B. Edmonds and F. Klugl, 'Special Issue: Agent Based Simulation of Complex Social Systems', SIMULATION, vol. 88, no. 1,pp. 4-6, 2012.
[8] K. Mustapha, H. Mcheick and S. Mellouli, 'Modeling and Simulation Agent-based of Natural Disaster Complex Systems', Procedia Computer Science, vol. 21, pp. 148-155, 2013.
[9] J. Ferber, an Introduction to Distributed Artificial Intelligence. Harlow, Eng.: Addison-Wesley, 1999.
[10] N. Jennings, K. Sycara and M. Wooldridge, 'A Roadmap of Agent Research and Development.' Autonomous Agents and Multi-Agent Systems, pp. 7-38, 1998.
[11] P. Maes. Agents that reduce work and information overload. Communications of the ACM, 37(7):31–40, July 1994
[12] A. H. Bond and L. Gasser, editors. Readings in Distributed Artificial Intelligence. Morgan Kaufmann Publishers: San Mateo, CA, 1988.

[13] M. R. Genesereth and S. P. Ketchpel. Software agents. Communications of the ACM, 37(7):48–53, July 1994.
[14] Ton Oguara, Dan Chen, Georgios Theodoropoulos, Brian Logan, Michael Lees, "An Adaptive Load Management Mechanism for Distributed Simulation of Multi-agent Systems", DS-RT, 2015, IEEE/ACM International Symposium on, Distributed Simulation and Real Time Applications, IEEE/ACM International Symposium on 2015, pp. 179-186.
[15] Vinoth Suryanarayanan, Georgios Theodoropoulos, Michael Lees, "PDES-MAS: Distributed Simulation of Multi-agent Systems", 2013 International Conference on Computational Science, Volume 18, 2017, Pages 671–681.
[16] Lars Braubach, Alexander Pokahr, Dirk Bade, Karl-Heinz Krempels, and Winfried Lamersdorf, "Deployment of Distributed Multi-agent Systems".
[17] Kumar, P. N.; Rupika, R.; Vennilla, S.; Abinaya, K.; Mohandas, V. P., "Implementing an Agent Based Artificial Stock Market Model in JADE - An Illustration", International Journal of Engineering & Technology, Jun/Jul2018, Vol. 5 Issue 3, p. 2636.