



# INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume3, Issue3)

Available online at [www.ijariit.com](http://www.ijariit.com)

## Software Agent Modelling Of Web Application

Arti Verma  
GITM, Lucknow

Yogendra Singh  
GITM, Lucknow

---

**Abstract:** *Businesses are becoming automated nowadays, with almost every business getting an online representation in the form of Web Application. The different kinds of businesses require dynamic nature of Web Applications which require less dependency on a human controlled software interface. With the applications catering to the needs of hundreds to thousands of users simultaneously, it's hard to manage such kind of applications using a combination of human interface. This particular need has led to the study and research in the field of software agents or intelligent agents, which use some kind of intelligence to aid the various operations of the application in use. Several studies and research have been made in this area which has resulted in development of some high-quality software applications over the web, which boast of having complete functionality driven by these intelligent agents and requiring minimum to no human interference. In this research work, software modeling of Web applications using intelligent agents has been envisaged. The various fundamentals of agent research have been discussed in detail with special heed to implementation framework. Finally, an agent-based model has been developed using the latest tools and techniques. This research work provides comprehensive analysis and implementation of an agent-based software application.*

**Keywords:** *Software Engineering, Software Applications, Multi-Agents, Negotiation, Interaction.*

---

### I. INTRODUCTION

Recent advances in software engineering, business process management, and computational intelligence resulted in methods and techniques for developing advanced e-commerce applications as well as supporting automating e-commerce business processes. Despite this fact, up to now, the most successful e-commerce systems are still based on humans to make the most important decisions in various activities within an e-business transaction. In this context, development of automatic negotiations is one of the most important research issues. While, depending on the type of the transaction, different negotiation procedures could be utilized, only a few proposed frameworks are generic and flexible enough to handle multiple scenarios. On the other hand, agent technology is often claimed to be the best approach for automating e-commerce business processes (including price negotiations). However, it is difficult to find successful large-scale agent-based e-commerce applications to confirm this claim. This paper presents negotiating agents that change their negotiation protocol and strategy through dynamic loading of negotiation modules.

Automated negotiation is achieving steadily growing attention as a mechanism for coordinating interaction among computational autonomous agents which are in a consumer-provider or buyer-seller relationship and thus typically have different interests over possible joint agreements. The thesis realizes the modeling and design of Adaptive Negotiation Agent System for Software Application, in which every agent has a cognitive part, composed of a knowledge base and an inference engine. The agents negotiate, based on a negotiation language, which contains a set of primitives, and also based on negotiation criteria, represented as rules. The negotiation process is a complex feature of traditional buying and selling. This process can be examined in the context of automated negotiation, as applied in the multi-agent based e-commerce. Real-world negotiations do not require the parties to reach a negotiated agreement; similarly, the automated negotiation covered here has the same option. An agent can choose "no deal" if it cannot negotiate a satisfactory agreement. Furthermore, we distinguish between "open" and "closed" marketplaces. A closed marketplace is based upon a predefined set of users, who "enroll" in the marketplace and agree to a certain set of rules. An open marketplace has no such agreement; agents are welcome to enter and exit at any time, and are required to agree to no rules.

### II. LITERATURE REVIEW

The design and implementation of an environment for automated negotiations, offering support for the use of various existing bargaining models, together with their respective negotiation strategies, is presented in [3]. It is possible to integrate different models, approximating the automated negotiations to the way the real world works. An open architecture for the negotiation environment is proposed, where the number of buyers and sellers, or the offer and demand for services or products, can be changed during execution time. In a flexible way, the negotiator agent can increase the set of negotiation strategies and also the number of business domains. Standard negotiation protocols such as on-line auctions already exist, and more are being developed all the time. In principle, it should be possible to develop some kind of uniform protocol theory, which embodies all kinds of protocols — ranging from auctions to protocols for co-operation to coalition formation. However, to expect every agent (irrespective of the domain) to be able to conform to any of these protocols would result in seriously heavyweight agents. Fortunately, though, it does not necessarily follow that every application will need a different protocol. There are many different tasks requiring coalition formation, and Preist expects that a (possibly small) set of such protocols would apply to most of them, just as a small set of auction protocols satisfies most applications involving one-to-many negotiation over a commodity. Beer argues that the mechanisms by which negotiation can be undertaken are fairly clearly defined (for example by FIPA). It is the negotiation currency that is very much application dependent. Most work reported is based on monetary values of some form, but we can easily conceive of situations where the speed of response or some measure of ‘closeness’ would be more appropriate. Schroeder puts forward the alternative view that it is unlikely that any such standard will emerge for two reasons. First, the applications involving argumentation/negotiation are very different in their nature ranging from transactions in auctions to argumentation in the philosophical sense. Second, if argumentation/negotiation is of importance to the participants, they will want to use the best protocol available making a standard solution unlikely [5]. These characteristics were implemented in the proposed environment with ontology concepts and production rules. The use of ontology-enabled the negotiator agents to be implemented, a priori, for any type of business domain. The fact is that the environment supplies a protocol for the agents to interact on the same ontology, allowing an agent to be detached from the various business domains it may be possible to negotiate. The individuality of each negotiator agent is therefore tied to the rules it possesses. Price negotiations are one of the important aspects of e-commerce transactions [6]. A rule-based implementation of automated price negotiations, used in a multi-agent system that models an e-commerce environment, is presented. A brief description of the conceptual architecture of the system and a simplified scenario that involves multiple buyer agents participating in multiple English auctions performed in parallel are described. Dominant-strategy mechanisms in allocation domains, where agents have one-dimensional types and quasilinear utilities, are studied [7]. Considering as input an allocation function, an algorithmic technique for finding optimal payments is presented, for a class of mechanism design problems. Optimality of payment functions is linked to a geometric condition. When the condition is true, an optimal payment function that is piecewise linear in agent types is described. Mechanism design problems that have no objective functions, but seek payments fulfilling a combination of constraints, are reduced at solving a system of linear inequalities. These reductions give solutions of mechanism design problems that are otherwise difficult to solve.

Regardless of advances in agent-oriented methods and technologies, due to lack of systematic methodologies to guide the development process, they have a limited rate of adoption in large-scale systems. So, agent-oriented methodologies play a significant role in this area. Currently accepted methodologies provide different modeling concepts and analysis techniques together with some part of supporting tools. Maturity and scope of coverage are different in these methodologies [6]. Below three famous methodologies are reviewed:

#### Prometheus

Prometheus methodology is an Agent-Oriented Software Engineering methodology (AOSE) with a focus on the development of intelligent agents rather than black boxes [7]. It has three phases of (1) System Specification, (2) Architectural Design, and (3) Detailed Design [3]. All software engineering activities are included in Prometheus, from requirements specification to detailed design and implementation [7]. Each phase has several models with a focus on dynamics of the system and whole system structure or structure of system components. Textual descriptor forms that provide the details for individual entities are also included in each phase [3]. During system specification, functionalities are defined and by using goals as scenarios the system will be specified; its interface with the specified environment is described in terms of actions, percepts, and external data. In architectural design, the types of the agent will be identified and the system structure will be captured. In detailed design, internal development of each agent will be developed and defined in detail [3].

#### Tropos

Tropos is a methodology with a focus on the analysis of requirements at its early stage. It supports all software development process activities, from system analysis to the system design and system implementation [4]. This methodology provides a general interface to different development process activities, documentation and evolution of the software rather than an incremental refined and extended model. This analysis process allows the reason for developing the software to be captured. The development process in Tropos consists of five phases: (1) Early Requirements, (2) Late Requirements, (3) Architectural Design, (4) Detailed Design and (5) Implementation [4]. Requirements analysis in Tropos is split into two main phases of Early Requirements and Late Requirements analysis. In the Architectural Design and the Detailed Design phases, the focus is on the system specification, regarding the requirements results from phase 1 and phase 2. A finally in Implementation the detailed design specification will be followed [4].

#### MaSE

MaSE is Multi-agent Systems Engineering that provides a methodology covering all aspects of systems’ lifecycle and assists developers in designing and developing multi-agent systems. It is a large-scale methodology being used in the analysis of multi-

agent systems. In MaSE a strong framework for design and development of multi-agent systems is provided by taking advantages of goal-driven development and using the power of multi-agent systems in defining roles, protocols, and tasks. Another advantage of this methodology is that the steps are defined one by one which results in a simple transition between models [8]. In MaSE the processes are described from an initial system specification to system implementation which will provide a roadmap for the system developer. These processes consist of seven steps, which are grouped into two phases [9]: (1) Analysis phase which consists of three steps: Capturing Goals, Applying Use Cases, and Refining Roles. The Design phase has four steps: Creating Agent Classes, Constructing Conversations, Assembling Agent Classes, and System Design [8].

### **III.AGENT BASED SOFTWARE IMPLEMENTATION**

In this research work, an agent-based framework has been used to provide automation in web applications. Automatic agents can be implemented to provide automatic handling of user requests and managing the various operations relevant to the Application. A demo application has been implemented using the intelligent agents. The framework is modeled according to the latest ontology techniques, to handle user request.

Java Agent Development Framework is basically a software environment implemented in Java language which aims at the development of multi-agent systems that complies with FIPA (Foundation for Intelligent Physical Agents) specification. JADE provides many of the classes required for agent-based software development. Some of them are:

- Agent;
- Behaviour;
- ACLMessage;
- Ontology.

JADE simplifies the agents' development process while ensuring standard compliance meet through a comprehensive set of system service and agent. JADE provides the following components for agents.

AMS (Agent Management System), which besides giving white page services, as specified by FIPA, also play the role of authority in the platform.

DF (Directory Facilitator) provide yellow page service to other agents.

ACC (Agent Communication Channel), which provide a Message Transport System (MTS) and is responsible for sending and receiving a message on an agent platform.

The control structure of the agent is composed of two phase. The first phase is dedicated to the control of agent activity which does not depends on another agent, while the second phase is given to negotiation and a reaching agreement. The step for the first phase are:

1. Generate desires, based on beliefs;
2. Generate candidate plans for achieving desired;
3. Generate intention, as the best possible plan;
4. If it has capabilities then execute intention;
5. If it hasn't capabilities then negotiate.

The steps for the second phase are:

1. Receive offers from other agents;
2. Update beliefs and planning knowledge;
3. Update desires and intentions
4. Generate counteroffers and send them to the other agents.

In this paper, a demo application has been modeled using agents. The application is basically a type of e-commerce application. They are basically dedicated multi-agents for shops and clients which are created through a GUI interface that links user's i.e buyers and sellers with their respective agents. However, these agents are in many ways spurious for the operation of the system. More precisely, a personal agent is considered to be a true representative of the user that resides on her machine and represents her interests in all aspects of e-life. Shop agents that will be a part of the e-marketplace; and therefore the GUI and personal agents are omitted from further considerations. The top-level conceptual architecture of the system illustrating proposed types of agents and their interactions in a particular configuration is shown in Figure 1. Let us now describe each agent appearing in that figure and their respective functionalities.

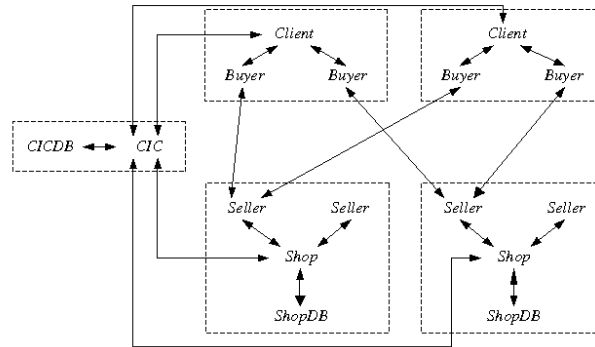


Figure 1: Agent Interaction

A Client agent (CA) is created to act within the marketplace on behalf of a user that attempts at buying something. Similarly, a Shop agent represents a user who plans to sell something within the e-marketplace. After being created both Shop and Client agents register with the interaction agent to be able to operate within the market. Returning agents will receive their existing IDs. In this way, the structure provides support for the future goals of agent behaviour adaptability. Here, agents in the system are able to recognize the status of their counterparts and differentiate their behaviour depending if this is a returning or a new agent that they interact with. There is only one information control agent in the system. It is responsible for storing, managing and providing information about all the agents existing in the system. To be able to participate in the marketplace all Shop and Client agents must register with the information agent, which stores information in the database. The database combines the function of client registry, by storing information about and unique IDs for all users, by storing information about of all shops known in the marketplace. Thus Client agents communicate with the information agent to find out which stores are available in the system at any given time. In this way, we are (i) following the general philosophy of agent system development, where each function is embodied in an agent and (ii) utilizing the publisher-subscriber mechanism based on distributed object-oriented systems. Furthermore, this approach provides us with a simple mechanism of correctly handling the concurrent accesses to a shared repository without having to deal with typical problems of mutual exclusion etc. Actually, all these problems are automatically handled by JADE's agent communication service. A Client agent is created for each customer that is using the system. Each Client agent creates an appropriate number of slave negotiation agents with the buyer role (Buyer agents hereafter).

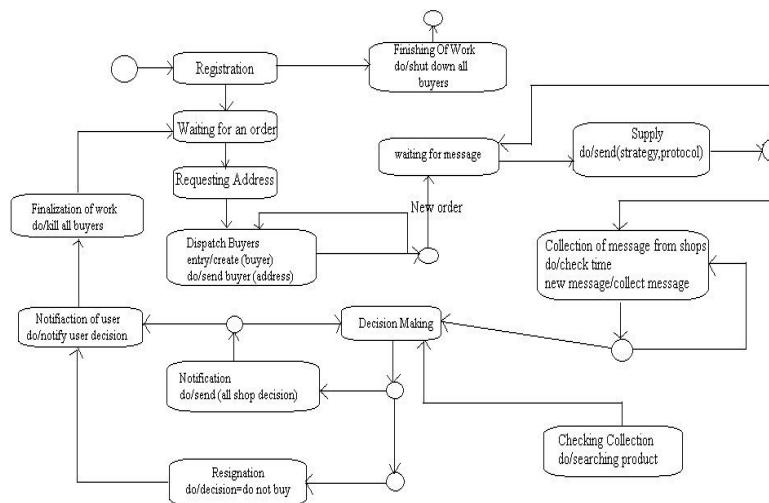


Figure 2: State Chart Diagram

Seller agents await incoming Buyer agents interested in buying their products and upon their arrival engage in negotiations with them. Let us now describe what happens in the marketplace after a customer has made a purchase request until a request is completed.

1. As specified above, a Client agent registers with the information agent. It obtains a new ID if it is a new Client or recovers its original ID if it is a returning Client. The information that an agent with a given ID is active in the marketplace is stored in the database
2. The Client agent queries the information agent to obtain the list of Seller agents selling the product it is expected to purchase. For each Seller agent on this list, it creates a Buyer agent to negotiate conditions of purchase.
3. Buyer agents migrate to Seller agent sites and query Seller agents about the negotiation protocol used in a given e-store and which Seller agent they should negotiate with. Then, Buyer agents dynamically load appropriate negotiation from Client agents and subscribe to the designated Seller agent, waiting for the negotiation process to start.

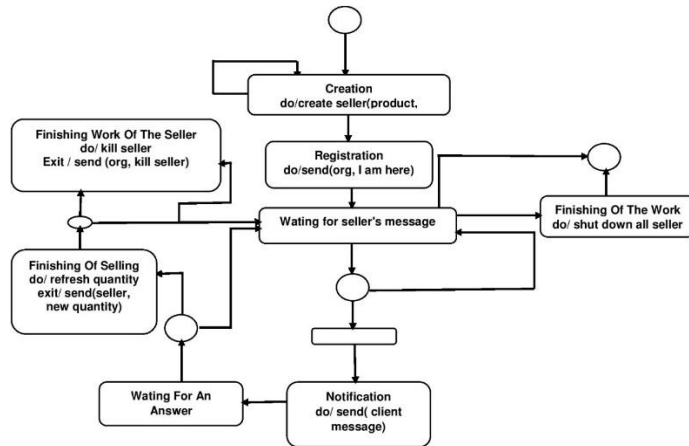


Figure 3: Seller Agents State Chart

Message Agent: Message Agent focus on client agent. Its monitor all the activities of client agent means how many time clients interact with the application, the scale of purchasing, monitor the products which are mostly visited by the client.

If client agent is temporary (client which interact application after the certain interval), then Message Agent only provide the information about those products which consist offer.

If client agent is regular (client which interact with application time to time), then the message agent generates the special offer for a client.)

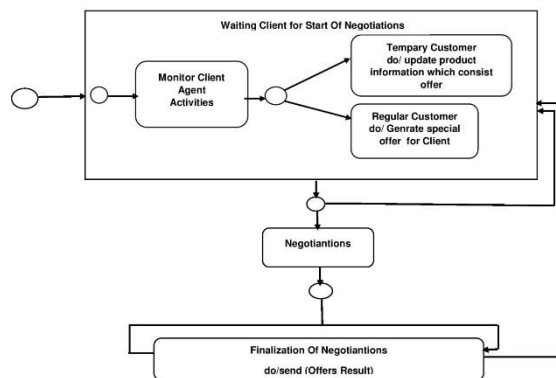


Figure 4: Statechart of Message Agent

## CONCLUSION

The automated negotiation has economic outcomes, because it has lower transaction costs, enabling higher volumes and new types of transactions in the electronic business domain. Through its automation, the negotiation mechanism becomes available to autonomous systems, improving the performance of these systems, when negotiation is used for agent coordination and cooperation, instead of existing interaction mechanisms.

The agents' behaviour can change during negotiation, according to previous interactions with other agents in the system. Changing behaviour may refer to either the use of different negotiation strategies or to concessions made for other agents, with which they have successfully negotiated in the past. To this aim, an agent develops a set of profiles during negotiation: the preference profile, the partner cooperation profile, and the group-of-partners' negotiation profile. The first two profiles characterize individuals, while in a group negotiation profile, several agent profiles are clustered, according to commonly discovered features. The automated negotiation mechanism facilitates the self-interested agents to make decisions, which give them the optimal outcome.

The research work presents an extended study of concepts related to multi agent's properties, implementation, and usage. The implementation framework presented in this research work provides a comprehensive methodology for developing an agent-based web application. Such a kind of framework will not only reduce the execution time of the application, it also makes the application more user-friendly.



## REFERENCES

- [1] Chen Y., Peng Y., Finin T., Labrou Y., Cost S., A Negotiation-based Multi-agent System for Supply Chain Management, Proceedings of Agents 99 Workshop on Agent-Based Decision- Support for Managing the Internet-Enabled Supply-Chain, 1999
- [2] Butz M.V., Learning Classifier Systems, GECCO 2010, Proceedings of the 12-th Annual Conference Companion on Genetic and Evolutionary Computation, 2331-2352, 2010
- [3] M. Winikoff and L. Padgham, "The Prometheus Methodology," in Methodologies and Software Engineering for Agent Systems. vol. 11, F. Bergenti, M.-P. Gleizes, and F. Zambonelli, Eds., ed: Springer US, 2004, pp. 217-234.
- [4] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An Agent-Oriented Software Development Methodology," Autonomous Agents and Multi-Agent Systems, vol. 8, pp. 203-236, 2004/05/01 2004.
- [5] Naroditskiy V., Polukarov M., Jennings N.R., Optimal Payments in Dominant-strategy Mechanisms for Single-parameter Domains, ACM Transactions on Economics and Computation, 1, 4, 1-21, 2013
- [6] Skylogiannis T., Antoniou G., Bassiliades N., Governatori G., Bikakis A., Dr-Negotiate: A System for Automated Agent Negotiation with Defeasible Logic-Based Strategies, Journal Data & Knowledge Engineering, 63, 2, 362-380, 2007
- [7] Kakas A., Moraitis P., Adaptive Agent Negotiation via Argumentation, Proceeding of the AAMAS'06 Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, 384-391, 2006
- [8] S. A. Deloach, M. F. Wood, and C. H. Sparkman, "Multi-agent Systems Engineering," International Journal of Software Engineering and Knowledge Engineering, vol. 11, pp. 231- 258, 2001.
- [9] Skylogiannis T., Antoniou G., Bassiliades N., Governatori G., Bikakis A., Dr-Negotiate: A System for Automated Agent Negotiation with Defeasible Logic-Based Strategies, Journal Data & Knowledge Engineering, 63, 2, 362-380, 2007
- [10] Trascau M., Tartareanu T.A., Benea M.T., Radu S., Establishing Social Order Through Norm Emergence, Proceedings of the 5-th International Conference on Computational Collective Intelligence, Technologies and Applications, Craiova, Romania, 2013
- [11] Wooldridge M., An Introduction to MultiAgent Systems, John Wiley & Sons, Ltd, Publication 2002
- [12] Goradia H.J, Vidal J.M., An Equal Excess Negotiation Algorithm for Coalition Formation, Proceedings of the 6-th International Joint Conference on Autonomous Agents and Multiagents Systems, 1059-1061, 2007
- [13] S. Kumar, Agent-Based Semantic Web Service Composition, Springer Briefs in Electrical and Computer Engineering, DOI: 10.1007/978-1-4614-4663-7\_2, the Author(s) 2012
- [14] Tamma V., Phelps S., Dickinson I., Wooldridge M., Ontologies for Supporting Negotiation in e-Commerce, Engineering Applications of Artificial Intelligence, 18, 223-236, 2005
- [15] Choi H.R., Kim H.S., Hong S.G., Park Y.J., Park Y.S., Kang M.H., Implementation of Framework for Developing Multi-Agent based Automated Negotiation Systems, Proceedings of the 7-th International Conference on Electronic Commerce, ICEC'05, 306-315, and 2005.
- [16] Dong H., Hussain F.K., Chang E., State of the Art in Negotiation Ontologies for Multi-agent Systems, International Journal of Web Services Practices, 3, 3-4, 157-163, 2008.
- [17] Sandholm T., Lesser V., Leveled Commitment Contracts and Strategic Breach, Games and Economic Behavior, 35, 1-2, 212-270, 2001.
- [18] Sierra C., Faratin P., Jennings N.R., A Service-Oriented Negotiation Model between Autonomous Agents, Lecture Notes in Computer Science, 1237, 17-35, 1997
- [19] Wooldridge M., An Introduction to Multi-Agent Systems, Second Edition, John Wiley & Sons, Ltd, Publication 2009.
- [20] Badica C., Ganzda M., Paprzycki M., Implementing Rule-based Automated Price Negotiation in an Agent System, Journal of Universal Computer Science, 13, 2, 244-266, 2007
- [21] Lin R., Kraus S., Can Automated Agents Proficiently Negotiate with Humans, Communications of the ACM, 53, 1, 78-88, 2010