

Online Book Ordering System Using Multi-Agent

Khaing May Zin, Khine Moe Nwe
University of Computer Studies, Yangon
khaingmayzin88@gmail.com

Abstract

With the rapid development particularly in online industry, the amount of the online users is also dramatically increased. Online shopping and online ordering system is very popular nowadays. Therefore, the information passing and processing on different areas is much more important for this situation. To be reliable and secure on the users' process, multi-agent systems have become an enormous and widespread methodology for understanding the user's requests and translate them for the other agents[6]. In this paper, the online book ordering system is proposed using multi-agent. In this system, the customers can order the books and also reserve the currently unavailable books of the system. The system can provide the rate of ordered book information to the member publishers related to their publishing house. In addition, XML Serialization is used for Agent Communication in this proposed system.

Keywords

Online Book Ordering System, Multi-agent System, XML Serialization

1. Introduction

The online market is one of the most popular making commerce because of its fast and easy way to do the business. The Internet, Web and other information technologies have brought in and will continue effect on e-commerce. Many types of business and cooperating and interacting between applications and systems become available from the Internet.

With the AI community, there is a growing interest in the development of autonomous systems; i.e., systems which are able to react to unexpected events[3]. Nowadays, agent-based technologies are considered the most promising means to deploy enterprise-wide and worldwide applications that often must operate across corporations and continents and inter-operate with other heterogeneous systems [1]. It is because they offer the high-level software abstractions needed to manage complex applications and because they were

invented to cope with distribution and interoperability.

Multi-agent systems are composed of multiple interacting computing elements, known as agents. For a successful communication in such heterogeneous environments, the agents have to share knowledge with each other[6].

In the proposed online book ordering system, three agents, user agent, publisher agent, and interagent, are cooperating tasks on behalf of its user. The user agent is capable of communicating with interagent for book ordering and reserving books by member customers. Advertising new book to the system and retrieving the rate of ordered book information are provided by communicating the publisher agent with the interagent. The interagent is mainly for exchanging information with user agent and publisher agent and processing the specific functions to the database. In this system, XML Serialization is used for sharing and exchanging information between agents.

2. Background Theory

This paper intended to describe Multi-agent system with Agent Technology. XML Serialization is integrated for communicating between different agents.

2.1. Agent Technology

There is no universally accepted definition of the notion of agent. However, the following four properties are widely accepted to characterize agents: autonomy, social ability, reactivity and proactiveness. Agents are autonomous computational entities (autonomy), which interact with their environment (reactivity) and other agents (social ability) in order to achieve their own goals (proactiveness) [6].

Agents typically represent different users, and there are thus several of them in a given environment. A multi-agent system is considered as a collection of collaborating autonomous agents, each representing an independent locus of control. Agents also provide an appropriate metaphor for conceptualizing certain applications, as the behavior

of agents more closely reflects that of the users whose work they are delegated to perform or support. The following characteristics of a domain are commonly quoted as reasons for adopting agent technology: an inherent distribution of data, control, knowledge, or resources; the system can be naturally regarded as a society of autonomous collaborating entities; and legacy components must be made to interoperate with new applications.

2.2. Multi-Agent System

Agents are sophisticated computer programs that act autonomously on behalf of their users, across open and distributed environments, to solve a growing number of complex problems. Increasingly, however, applications require multiple agents that can work together. A multi-agent system (MAS) is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver[7].

Multiagent systems are systems composed of multiple interacting computing elements, known as 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. Agents are computer systems with two important capabilities. First, they are at least to some extent capable of autonomous action – of deciding for themselves what they need to do in order to satisfy their design objectives. Second, they are capable of interacting with other agents – not simply by exchanging data, but by engaging in analogues of the kind of social activity that we all engage in every day of our lives: cooperation, coordination, negotiation and the like.

The characteristics of MASs are that (1) each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint; (2) there is no system global control; (3) data are decentralized; and (4) computation is asynchronous [2].

2.3. Agent Communication

Agents communicate in order to achieve the better goals of themselves or of the society/system in which they exist. Communication can enable the agents to coordinate their actions and behavior, resulting in systems that are more coherent[5].

Agents can improve the coherence of their problem solving by planning the content, amount, type, and timing of the communication they exchange. Using abstraction and meta-level information (for example, organizational knowledge) is helpful because they help decrease communication overhead. In dynamic and open environments, inhabited by heterogeneous agents, additional issues

need to be faced. The most prominent among them is agent interoperability.

Agents establish communications using agent communication language around a set of message types (preformatted), message content schemas, and message meta-information. Among them, XML serialization is applied in this system in order to easily dispatch information to and from between two agents.

2.3.1. XML Serialization/Deserialization

Object Serialization is a process through which an object's state is transformed into some serial data format, such as XML or binary format, in order to be stored for some later use. The process of transforming the contents of an object into XML format is called serialization, and the reverse process of transforming an XML document into a .NET object is called deserialization [4].

Some good uses for serialization/ deserialization include:

- Storing user preferences in an object.
- Maintaining security information across pages and applications.
- Modification of XML documents without using the DOM.
- Passing an object from one application to another.
- Passing an object from one domain to another.
- Passing an object through a firewall as an XML string.

Here is a simple Book class that is passed between two agents:

```
public class Book
{
    public string BookID;
    public string BookTitle;
    public Book(string bID,string bTitle)
    {
        BookID=bID;
        BookTitle=bTitle;
    }
}
```

Figure 1 shows the sample flow of serialization and deserialization of an instance object of above book class, "Book1". XML serialization applies to the instance object "Book1" as XML format with binary data. As binary information, it is easy to send to remote agent of the applications.

At the remote application, after completing the receiving the XML binary data, the information have to deserialize to the original object instance. And then the application can process on that object.

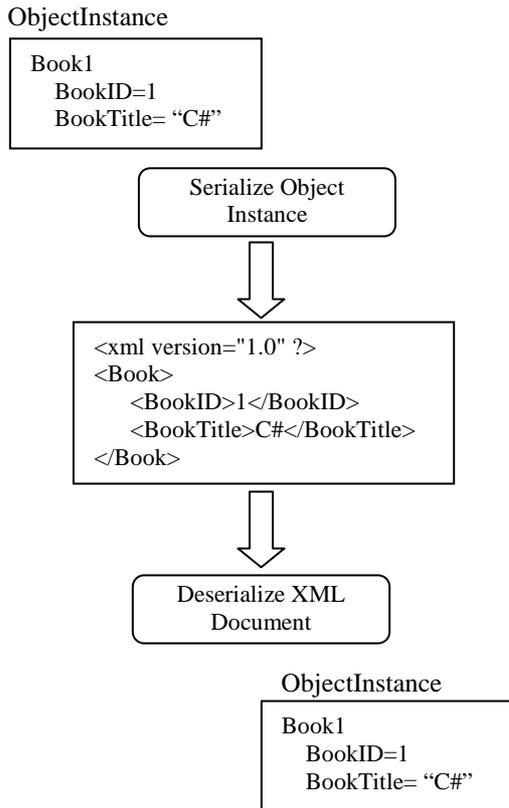


Figure 1. Sample Flow of serialization and deserialization

XML formatted data is easily pass through the firewall without difficulties for sending and receiving the information to any application, any place or any area. After completing the waiting message from the receiving agent, the received XML document is deserialized and the agent continues to process the specific function. The interaction between agents using XML serialization is shown in Figure 2.

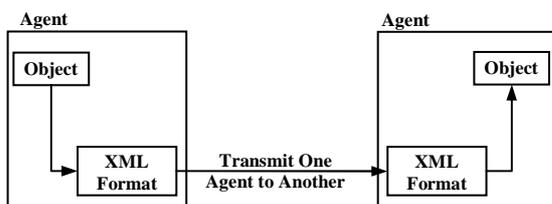


Figure 2. Interaction between Agents using XML Serialization

3. Proposed System Architecture

In this paper, the online book ordering system is proposed by using multi-agent. The overview architecture of the proposed system is shown in Figure 3. In this proposed system, XML serialization is used for communication language between agents.

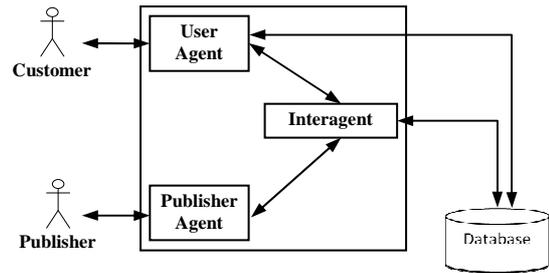


Figure 3. Overview of System Architecture

The customers make the book ordering and book reservation through the user agent and the publishers do the book advertising and review their ordered book rate via the publisher agent. Both the user agent and publisher agent perform the above process communicating with the interagent.

When the books are currently not in the system, the customer makes the book reservation and this reserved information is saved through the interagent into the database. The available reserved book information is inserted by the administrator as latest books. The matched advertised books, which match the customer's reserved information, will be informed to the customers when the publisher advertise as advertised books to the system.

3.1. User Agent

The user agent is an agent that mainly acts as an intermediary between interagent in order to be effectively and efficiently interaction with the user. This agent serves the member customers for login to the system to continue the order and reserve process. The user agent retrieves the book lists such as customer's interested book list, popular book list, available customer's reserved book list and the reserved books list advertising from the publisher to the system.

Customers collect books to the cart before making the ordering process. When the customer orders the books from the cart, the user agent is able to perform the order process by communicating with the interagent and then saving this information as user's profile into the database.

For each customer, the ordered book information records as user's profile, the user agent can retrieve the interest book list according to the user's interested fields and the popular book list according to the entire records of ordered book information can be retrieved by interacting with the interagent.

Currently, the customer can reserve the book which is not available in the system. The interagent sends the message of available customer's reserved book information and new advertised book information. The user agent records this information in user's profile.

3.2. Publisher Agent

This agent is able to perform on the behalf of the member publishers and communicate with the interagent for the advertised and related ordered book information. When the member publisher advertises their published new book, the publisher agent is able to do as an advertisement to the system, and then the publisher agent can also send this new published book information to the interagent. The publisher agents also retrieve the information related with order rate of each publishing house by communicating with the interagent to provide the publisher.

3.3. Interagent

The interagent is the main part of the whole system. The user agent and publisher agent communicate with the interagent in order to process the specific function to the database. The interagent checks the user validation from both the user and publisher agent. The interagent communicates with the user agent to perform order and reserve process for the customer request and interacts with the publisher agent to advertise book into the system.

The administrator does the management process through the interagent. When the administrator inserts the new book information in the system, the interagent matches with the reserved book information and sends back the available reserved book information through the user agent to the customer. When the interagent receives the new advertised book list from the publisher agent, this agent informs the new advertised books to the user agent and match with reserved books according to the customer request.

When the publisher can request the rate of their ordered book information through the publisher agent, this agent can send this information through the interagent to publisher agent in order to retrieve the ordered information. This agent can also keep track as the descending the rate of customers' ordered book information communicating the publisher agent and interagent.

3.4. System Design

This system can be mainly processed for the ordering process. The three sequences diagrams of customer processes, administrator processes and publisher processes are described below.

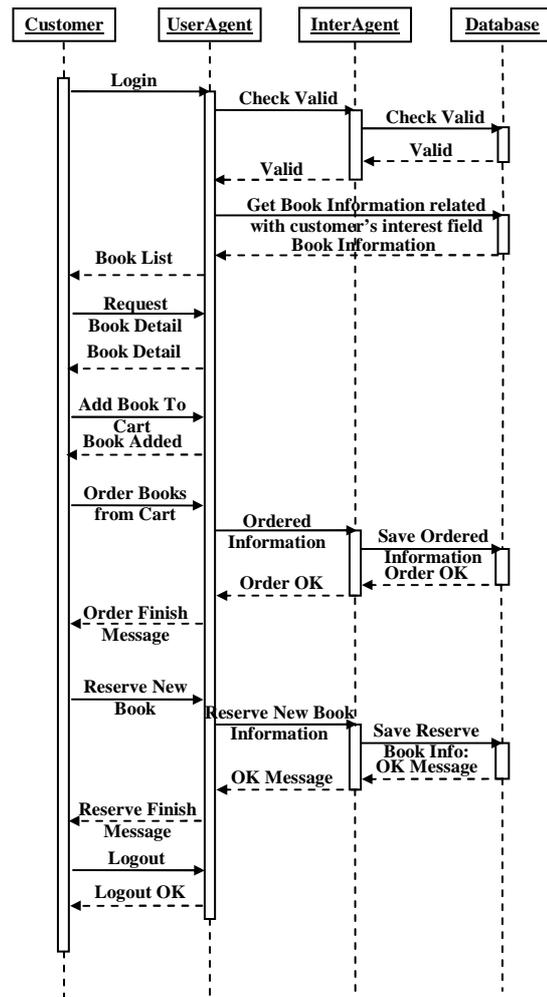


Figure 4. Sequence Diagram of Customer's login, book ordering and reservation processes

The sequence diagram of customer's login, book ordering and reservation processes is shown in Figure 4. Customer can login to the system, can retrieve the book information, can order books and can reserve the book which is not currently available in this system.

As in Figure 5, the sequence diagram of the publisher's login, retrieving rate of ordered book information and advertising processes is shown. Publisher can mainly apply to view the rate of their ordered book information and advertise their new published books.

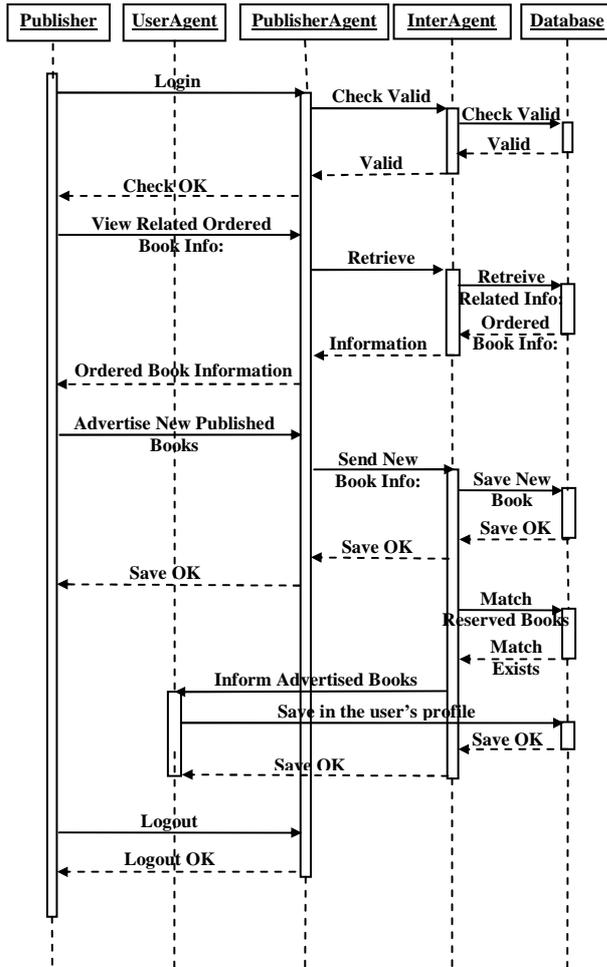


Figure 5. Sequence Diagram of Publisher's login, retrieving rate of ordered book information and advertising processes

The sequence diagram of the administrator's management processes is shown in Figure 6. Administrators can manage book information from the system such as adding, editing and deleting book information, retrieve rate of ordered book information and edit publisher information through the interagent.

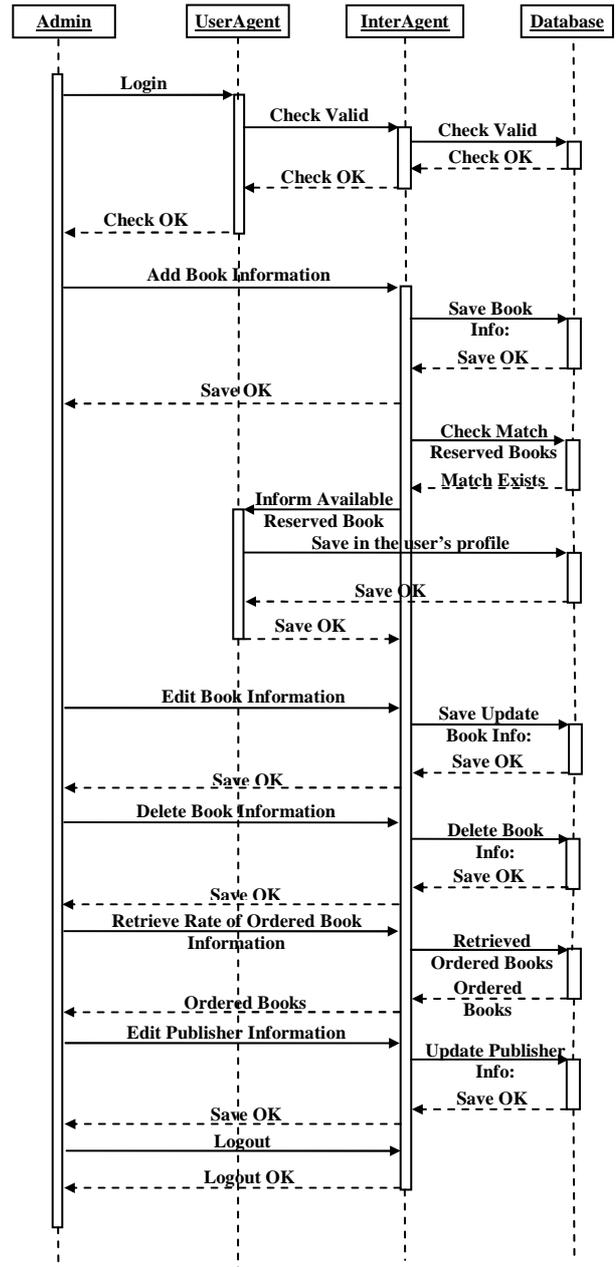


Figure 6. Sequence Diagram of Administrator's management processes

4. Conclusion

Multiagent system, which consists of a number of agents, interacts with each other by exchanging information. Agents can operate independently without other direct intervention. In this system, three agents such as user agent, interagent and publisher agent are implemented in order to cooperate for book ordering, reservation and advertisement processes. This system can provide the member customer by providing the customer's interested fields of the books for ordering process

and book reservation to the system. This system can also support the member publisher by providing the advertisement of their published new books and provides the information related with the rate of ordered books.

5. Limitation and Future Work

In this online book ordering system, the quantity of ordered books does not consider to control the stock of the system when doing the ordering process. The system can be extended by adding the stock control as future work. The system can be extended with e-banking services so that the customers can make online payment as the prepaid system. To be secure on exchanging information, the data encryption can be applied to the information before XML serialization as future work.

6. References

[1]. B. Fabio, P. Agostino and R. Giovanni, "Developing Multi-agent Systems with JADE", Intelligent Systems, Springer, Italy, 2001.

[2]. K.P. Sycara, "Multiagent Systems", AI Magazine, American Association for Artificial Intelligence, 1998.

[3] M. Roberto and T. Pietro, *Team Cooperation for Plan Recovery in Multi-Agent Systems*, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.6169&rep=rep1&type=pdf>

[4] N. Tony and W. Shawn, *Microsoft .Net Framework 2.0 Application Development Foundation*, Microsoft Press, America, 2006.

[5] W. Gerhard, *Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence*, MIT Press, England, 1999.

[6] W. Michael, *An Introduction to MultiAgent Systems*, John Wiley & Sons, England, 2002.

[7] Intelligent Software Agents, <http://www.cs.cmu.edu/~softagents/multi.html>