2000

# Internet-based Multiagent Architecture

Jess H.K. Yuen
*City University of Hong Kong*, isjess@is.cityu.edu.hk

Felix S.K. Leung
*City University of Hong Kong*, isfelix@is.cityu.edu.hk

Huaiqing Wang
*City University of Hong Kong*, iswang@is.cityu.edu.hk

Stephen S.Y. Liao
*City University of Hong Kong*, ishongko@is.cityu.edu.hk

# Internet-based Multiagent Architecture

Jess H. K. Yuen, Felix S. K. Leung, Huaiqing Wang, Stephen S. Y. Liao,
Department of Information Systems, City University of Hong Kong
{isjess, isfelix, iswang, ishongko}@is.cityu.edu.hk

## Abstract

Research in intelligent agents and multiagent systems that run on the Internet has received increased attention and importance in recent years. Since the Internet continues to grow, intelligent agent technology is progressively being introduced to many Internet-based applications for communications between different applications. The aim of this paper is to present an Internet-based architecture for multiagent systems which offers a communication infrastructure and coordination services for agents to achieve their goals.

A structured architecture is proposed to support communication facilities among several agents and coordinate agent activities in distributed environments such as the Internet and Intranets. The architecture consists of 1) Application agents, 2) Communication handler, 3) Knowledge manager, and 4) Repository. (Yuen, et al. 1999; Leung, et al. 1999). These four layers cooperate together and provide common facilities necessary for typical multiagent systems or agent-based applications with various choices.

An Internet-based prototype for auditing and detecting unauthorized transactions within an organization over the Internet or an Intranet is implemented to demonstrate the practicability and feasibility of the proposed Internet-based architecture for multiagent systems.

## Introduction

Intelligent agents are one of the important paradigms for developing traditional software applications (Wooldridge et al. 1995). Due to their special characteristics- relatively small size, proactive autonomous behavior and interactive nature; agent technology is currently regarded as the new revolution in web-based applications. It can run in a highly distributed environment. In fact, many researchers, both in the business or academic area, have shown their interests in agents and related technologies for e-business (Yuen et al. 1999). It is generally believed that agent will play a very important role in development of technology web-based applications.

Unlike traditional business processes, e-commerce can be carried out anytime, anywhere, provided that our computers are inter-connected electronically. For instance, the Internet, which is the largest network in the world, connects people from almost every corner of the world. Not surprisingly, traditional techniques for building software are no longer adequate on the web. Firstly, they are designed for running in the highly distributed environment. Neither data nor information can be exchanged between different platforms. This greatly reduces the efficiency of the application. Secondly, there is an increasing demand for great extendibility of web-based applications which traditional programming techniques cannot guarantee. Thirdly, there is relentless demand for more web services such as cooperative query processing, dynamic data retrieving, data mining, authenticating, content personalizing, and so forth. With all these problems, agent-based systems, or simply referred as agents are introduced to satisfy the needs on the web.

Agents are technologically powerful. Yet one key question remains: How should agent technology be applied in order to deliver business solutions for the Web? How can assure the effectiveness and efficiency of the agent solution be accessed? The answers to these two questions are still remaining in the top agenda.

In our previous research, multiagent architecture was designed for various domains such as decision support systems (IADSS) and artificial intelligent systems (APACS) (Wang et al. 1997; Wang 1997). However, those systems were designed for operating in traditional environments. In this paper, we introduce a newly implemented multiagent architecture which is enhanced for running on Internet. The Internet-based architecture, which supports the build up of the multiagent systems, can be reused, easily extended and can be flexibly adapted to future changes in agent technology or other environments. In the following sections, the contributions and limitations of related work will be presented. Then major features of the services provided by the architecture will be addressed and discussed. Finally, we will evaluate the performance of the architecture by applying it into the application domain: a Java-based Multiagent Auditing System for financial organizations.

## Related Works

Over the years, many researchers and developers have investigated the most efficient ways to develop an agent system or architecture to assist operations. In particular, JAT is a set of programs which consists of several Java templates and a Java-based agent

infrastructure. It provides a common facility system such as the communications infrastructure for agents to build agent. In the 1996 March, modified version of JAT called JATLite was revealed.

Another work on Agent architecture involved the Advanced Plant Analysis and Control System (APACS). APACS is one of the largest real-time knowledge based system projects undertaken in Canada (Wang et al. 1996). Intelligent agents were introduced into a modern nuclear operating environment. We designed APACS not only for its immediate practicability to nuclear plant systems, but also to demonstrate the feasibility and great benefits of using intelligent agent technology into the real-time process industry (Wang et al. 1997). In APACS project, we introduced a multiagent architecture that consists of three main layers: the application layer, knowledge broker layer and repository layer.

Certainly, JATLite provides us many great features for building an agent communication systems, yet it has a limitation. The infrastructure of JATLite is built purely for supporting agent communications. It does not facilitate the use of any particular reasoning tools or knowledge systems. As a result, the application of JATLite in the AI area is limited. Apart from the JATLite, APACS is not designed for running in heterogeneous environments, which make it difficult to operate smoothly on the Internet. Thus, we would like to have multiagent architecture for building an agent system on the Internet.

## Major Services of Intelligent Agents

### KQML

The ultimate goal of the proposed architecture is to provide the common facilities necessary for typical multiagent systems or agent-based applications with various choices. It assumes an underlying communications protocol that determines the possible relationships among all agents (Lejter et al. 1996). For instance, as Labrou (1999) mentioned, it is believed that the development of an effective, rich agent communication language (ACL) is one of the keys to the agent paradigm. Therefore, one major objective of building this architecture is to offer a good agent communication language to agents in the distributed environment.

Currently, our architecture supports the usage of Knowledge Query and Manipulation Language (KQML), which is a flexible, reusable and extendible ACL for many agent architectures currently in use (Finin et al. 1994). The contribution of KQML is to provide the means for agents to exchange information and knowledge between agents and the knowledge manager.

### Intelligent Capability

Secondly, the architecture provides and supports a relatively sophisticated event-processing capability (Bigus et al. 1997). In other words, agents provided by the architecture will act and react intelligently. Hence, our agents will need to handle events from other agents, events, and outside applications. Moreover, other reasoning facilities such as Java Expert System Shell (JESS) or other expert systems can be added into the architecture.

### Knowledge Sharing and Storage

Thirdly, our architecture is equipped with an object-oriented database system called ObjectStore for permanent storage and knowledge sharing. As the OO DBMS is employed, knowledge can be shared in a transparent fashion (Wang et al 1996). Besides, the OO DBMS should govern the storage and processing of knowledge and information over the Internet or interconnected computer systems in which both data and processing functions can be distributed among several areas.

## Implementation Overview

Our agent architecture maintains certain services such as agent communication services, reasoning engine, and knowledge sharing via web. Agent system developers can easily adopt such kinds of provided built-in services. Most importantly, they can develop new services for a particular application domain, or enhance the existing services without requiring them to re-write the existing codes. Rather, as the architecture is implemented in a purely object-oriented approach, the developers can extend the original classes by enjoying the inheritance of OO architecture.

We use Java Remote Method Invocation (RMI) to facilitate the communication of remote Java objects communicating. The system is a 3-tier distributed one which is developed with JDK 1.2. The first tier is the web based user interface, which in fact is a Java Applet. The agents, which are implemented as Java applets too, are distributed to the clients and perform their tasks. These Java-developed agents could move around a network to accomplish complicated tasks (Hendler et al. 1999). The knowledge manager and communication handler, which are the central parts of the second tier, run at the server side and activate while the system starts. (Yuen et al, 1999). They process agent's messages and offer agent services to agents. At the back-end side, we employ Object Store, an object-oriented DBMS, as the information repository located in the server for permanent knowledge and data storage.

Clients can activate the agents by entering the IP address or domain name of the server on the web browser. The Java applet is then downloaded from a web server

that is dedicated for handling client's requests from every point on the Internet. Java agents will be created at run time and remotely registered to the knowledge manager. Agents can then freely communicate together via the ACL services provided, and perform their tasks or achieve the goals specified by the knowledge manager.

## Testing

In order to evaluate and verify our architecture, what we need to do is to apply it into an application domain with a prototyping effort. A set of agent services is implemented and facilitated for agents in the prototype system to demonstrate the feasibility of the architecture we proposed.

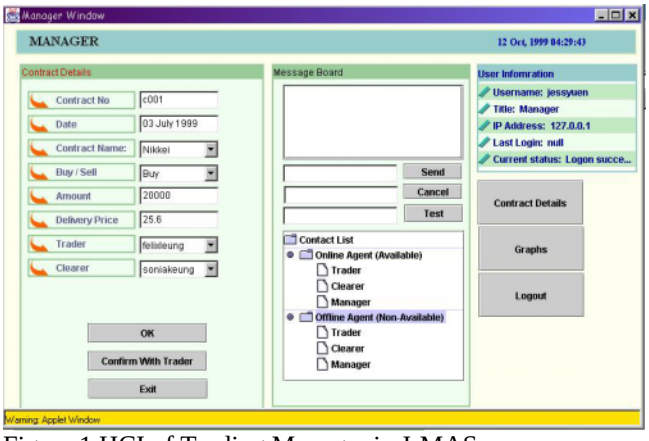

Figure 1 HCI of Trading Manager in J-MAS

We have built a small-scale prototype system in which we applied our newly implemented architecture. Our prototype system is a web-based, client /server system. Java-based MultiAgent Auditing System (J-MAS), which applies a generic multiagent architecture, is implemented to handle security issues inside several financial organizations. In particular, we select the case of the crash of Barings plc. to simulate how the system audits and monitors the transactions. In February 1995, the oldest British merchant banking group, Barings plc. collapsed suddenly. One major reason for the sudden collapse was the uncontrollable and unlimited amount of transactions made by the senior trader, Nick Leeson, (Hunt et al. 1996). Thus, the prototype system is developed to demonstrate the use of intelligent agents and our newly developed architecture for auditing the transactions within the organization, detecting potential risks, and avoiding uncontrollable events via the web similar to Leeson in the Barings' case.

J-MAS consists of several intelligent agents operating Internet /Intranet applications such as Audit agent, Tracking agent, Information agent, HCI agent and DAQ agent. They cooperate and coordinate with to achieve the goal of the systems (Yuen et al. 1999). In fact, a pilot test demonstrates that, while autonomous agents continually collect data over the web and perform real time analysis on the knowledge manager, users are not only become more alert to abnormal behaviors of system users, but can also prevent the potentially hazardous events from happening.

## Conclusions

The architecture proposed is designed for the dynamics of the Internet environment at all levels. Firstly, for supporting communications, agents can inter-exchange their knowledge with others and also transfer messages to knowledge managers in the web. Secondly, the knowledge manager collects information from various types of agents in a timely fashion and reacts immediately. A small pilot test has successfully demonstrated that the communication infrastructure does help agents to inter-operate and communicate in order to achieve their goals. Presently, we are going to construct a larger scale prototype to assess the extendibility of our architecture. In the future, more testing will be conducted to assure its flexibility and practicability.

## References


Bigus J. P., Bigus J., "*Constructing Intelligent Agents with Java - A Programmer's Guide Smart Applications*" John Wiley & Sons, Inc., 1997.

Finin T., Fritzson, R., McKay, D., McEntire, R., "*KQML as an Agent Communication Language*" in Bradshaw, JM (ed) Software Agents, MIT Press, pp. 291-316.

Hendler, J., "*Making Sense out of Agents" IEEE Intelligent Systems*, 1999 Mar/April, pp.32-37.

Hunt, L., Heinrich, K., "*Baring Lost - Nick Leeson and the collapse of Barings Plc*." Butterworth-Heinemann Asia 1996.

Labrou, Y., Finin, T., Peng Y., "*Agent Communication Languages: The Current Landscape*" IEEE Intelligent Agents, 1999 Mar/April, pp.45-52.

Lejter, M., Dean, T., "*A Framework for the Development of multiagent Architectures*" IEEE Expert, December 1996, pp.47-59.

Leung, F., Yuen, J., Liao, S., Wang, H., "*A Conceptual O-O Model for Internet-based Intrusion Detection Agents*" Proceedings of the Fifth Americas Conference on Information Systems, 1999 Aug.

Wang, H., "*Intelligent Agent–Assisted Decision Support Systems: Integration of Knowledge Discovery, Knowledge Analysis, and Group Decision Support*", Expert System with Applications, Vol.12, No.3, 1997, pp.323-335.

Wang, H., Wang, C., "*Intelligent Agents in the Nuclear Industry*", Computer, November 1997, pp.28-34.

Wang, H., Wang, C., "*APACS: a multi-agent system with repository support*", Knowledge Based Systems Vol. 9, 1996, pp.329-337.

Wooldridge, M., Jennings, N., "*Intelligent Agents: Theory and Practice*" The Knowledge Engineering Review, Vol. 10, No. 2, 1995, pp.115-152.

Yuen, J., Leung, F., Wang, H., Liao, S., (1999), "*A Multi-Agent Architecture for Internet Security*" Proceedings of the Fifth Americas Conference on Information Systems, 1999 Aug.

"*Jess, the Java Expert System Shell*", http://herzberg.ca.sandia.gov/jess/index.html

"*JATLite*" http://java.stanford.edu/java_agent/html/

*UMBC KQML Web*" http://www.cs.umbc.edu/kqml